UNIVERSITY OF CALIFORNIA,
IRVINE

Comparison of multi-atlas based segmentation techniques for human MRI

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Biomedical Engineering

by

Vyshnnavi Parthasarathy

Thesis Committee:
Dr.Frithjof Kruggel, Chair
Dr.Gultekin Gulsen
Dr.Zhongping Chen

2016

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF VARIABLES

Variable Name	**Description**
x	Voxel of interest
i,j	Voxel indices
P(x_i)	Patch centered at voxel x
s	Reference atlas
w	Weight assigned between patches
y(s,j)	Label at index j in atlas s
L(x_i)	Label decision at voxel x
v(x_i)	Label estimate value
h	Width of Gaussian
σ	Standard deviation of intensity difference distribution
N	Number of voxels in a patch
β	Scaling factor
A	Matrix of intensity values
b	Vector of intensity values in a patch
λ	Optimization constraint
W	Probability that a particular voxel has a particular label
Y	Label values in reference atlases
I	Intensity values in reference atlases
T	Hidden true segmentation
Θ	Performance parameters of raters
μ	Mean of intensity values
q	Standard deviation of intensity values

ACKNOWLEDGMENTS

I would like to thank my committee chair, Professor Frithjof Kruggel for the opportunity to work under his expert guidance. His motivation, valuable advice and support have played a significant role in the successful completion of this thesis as well as my graduate studies at UCI. I would also like to thank Professor Gultekin Gulsen and Professor Zhongping Chen for their valuable time and suggestions.

I am thankful to Jerod Rasmussen and Yang Zhang, my colleagues at the Signal and Image processing lab for having greatly enhanced my research experience in the lab. I would also like to thank Julia Gordon for her contribution related to my research in the summer of 2015.

I am greatly indebted to my wonderful cousin Amuthan A Ramabathiran for all the moral support and timely guidance which made my graduate student life very smooth and successful. Finally, I would like to thank my parents Parthasarathy D and Usha P and brother, Veeraraghavan Parthasarathy for believing in my abilities and supporting me in all my endeavors.

ABSTRACT

Comparison of multi-atlas based segmentation techniques for human MRI

By

Vyshnnavi Parthasarathy

Master of Science in Biomedical Engineering

University of California, Irvine, 2016

Dr.Frithjof Kruggel, Chair

Medical image segmentation is the process of segmenting/ sectioning out a particular structure of interest from an entire image, which is obtained from an imaging modality such as MRI or CT. The segmentation procedure used often depends on different factors such as the imaging modality, the properties of the structure of interest and the computational performance required. The problem of image segmentation is a widely explored topic in the domain of medical image processing. It makes the study of complex structures easier, which in turn helps immensely in better diagnosis and treatment planning. In this work, the aim is to study the performance of five different approaches for segmenting five different structures of the human brain in a T1 MR image. These methods make use of information from already segmented reference images to perform segmentation on the input and hence are classified as multi-atlas (multiple references) based techniques. They treat the entire brain volume as a group of patches (made of individual voxels) and perform segmentation by operating at the patch level and hence are called the patch based methods. The Dice coefficient is used as a measure to evaluate segmentation performance by each of these methods. Through this analysis, the objective is to implement, understand and analyze each of these methods and also identify their shortcomings.

Chapter 1

Introduction

1.1 Image segmentation

Segmentation splits an image into distinct regions. A region consists of a spatially connected set of image elements that are considered as homogeneous with respect to a specific predicate (homogeneity criterion). Image segmentation is a crucial step for object recognition. Automated object recognition finds application in a wide variety of fields such as satellite imaging, surveillance and medical imaging. As a result, the domain of image segmentation continues to be an active topic of research.

In medical image processing, segmentation procedures are a key in image analysis and help making better informed diagnostic decisions by enabling a better understanding of the structure of interest be it a tumor, or a structure oblivious to the human eye in a normal MRI or CT scan. Prominent applications of medical image segmentation include:

- performing image analysis, identifying and analyzing structures of different organs in medical images to get useful diagnostic information

- quantifying volumes of the structure either for pathological assessment or for surgery planning
- assessing disease progression such as in case of tumor growth
- treatment planning (contouring during radiotherapy planning in treatment of cancer).

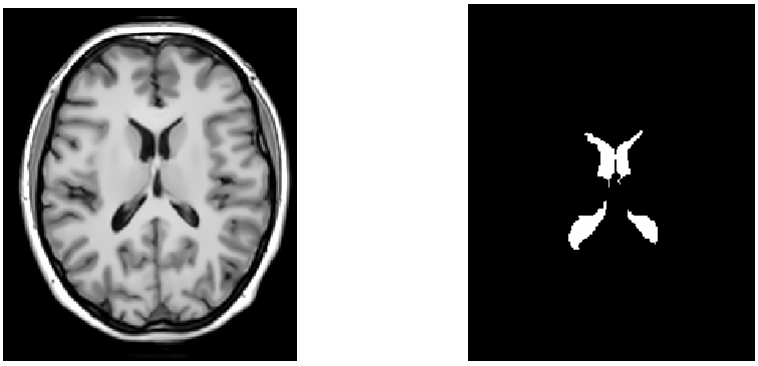


Figure 1.1: T1 weighted MRI of human brain(axial slice) with segmentation of ventricles

1.2 Motivation

The study of human brain is one of the most actively pursued as well as challenging domains of research. The complexity of the brain constantly necessitates the need for newer and better techniques at various levels to understand it. Performing segmentation of the human brain to study structures of interest better takes us closer to the goal of recognizing the functional and structural correlation between the different structures that make up the brain. This is especially useful for the diagnosis and treatment of various brain diseases and disorders. Existing research points to a number of cases where structural variation in the brain has been both a cause and consequence of disorder including psychological disorders [2]. Hence, development of robust segmentation techniques could potentially open more doors towards understanding the origin and progression of disease better. The objective of this work is to study the performance of some of the existing multi atlas based segmentation techniques on a

publicly available MR brain image dataset[8] and to incorporate the most efficient algorithm into a bigger scheme of processes aimed at modeling the human brain.

1.3 Existing techniques

Often, in medical image segmentation procedures, there is a trade off between the accuracy and the efficiency of performance. The performance is also impacted by the presence of noise in the image occurring due to magnetic field inhomogeneities as well as motion artifacts. The kind of segmentation technique that has to be employed depends on the structure of interest to be studied, the imaging modality used as well as the purpose of the study. The type of application using the segmentation procedure also decides between a manual, automated and a semi automated segmentation procedure.Segmentation routines are based on regional properties (e.g. intensity value, the location, texture, shape based features) that serve as a homogeneity criterion.

Sharma et al [17] have reviewed the existing automated medical image segmentation techniques applicable to MR and CT images. They categorize existing techniques into four types:

- **Gray level features based methods:**

  Procedures that make use of gray level features operate by thresholding. This thresholding could be at different levels - thresholding based on histogram for bimodal images, thresholding of edges for identification of object boundaries and region based thresholding to perform region based segmentation (region merging and region growing).[14]

- **Texture feature based methods:**

  Texture based methods make use of the texture features of an object in an image to perform segmentation. These texture features could be extracted statistically, syntac-

tically or by spectral approach; what results in each method is usually a feature vector in a multi dimensional space which can be used to perform segmentation as well as classification. [19, 18]

- **Model based segmentation:**

  Model based methods make use of the repetitive geometry of the structure of interest to develop a probabilistic model for the given structure. The similarity of an image region is used to detect its presence in similar images.

- **Atlas based segmentation:**

  Atlas based techniques involves the use of an expert segmented reference image as the standard to perform segmentation of the input test image. This involves ensuring that both the reference and the input images are aligned to the same coordinate space using an appropriate method (image registration). Once the alignment has been performed, label information is transferred from the atlas to the sample image.

The main focus of this thesis is the atlas based segmentation method. A single atlas based segmentation makes use of information from a single expert segmented reference (atlas) image. An extension of this idea is the multi atlas based segmentation (MAS) technique. Including multiple references enables a wider variation in the properties of the structure to be considered for segmentation, thereby making the method more robust in comparison to SAS. Since these use multiple references, there is an associated disadvantage of increased computational memory and time requirement. Hence, the implementation of these methods has to include an appropriate choice of optimization steps. A detailed discussion about the technique and its application for studying the structures of interest in the human brain is presented in the thesis. The next chapter gives an overview of the different aspects of the MAS technique. In the subsequent chapters, the methods used in this work, the details of the dataset used and the implementation are discussed. The final chapter includes a discussion of the results, the primary conclusions of the thesis and suggestions for future research.

Chapter 2

Basic Concepts

2.1 Multi atlas based segmentation technique

Multi atlas based segmentation (MAS) technique was introduced as an improvement over the single atlas based segmentation technique (SAS).MAS draws from extracting information from similar atlases to make label decisions in an input test image. The basic idea is to assess the majority vote of the reference atlases for a particular voxel. In comparison to SAS, MAS represents information pertaining to a wider variety of anatomical variations since label information for a given structure based on multiple reference images is made available. Iglesias et al [11] reviewed different MAS methods for the procedural steps in performing MAS. A typical MAS pipeline involves the steps shown in the following flowchart.

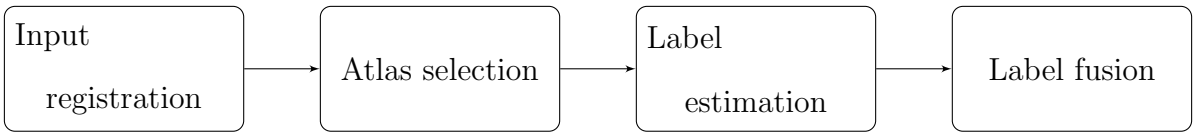


Figure 2.1: Steps in MAS, adapted from [11]

The scheme can be broken into three steps - identification of the most similar reference

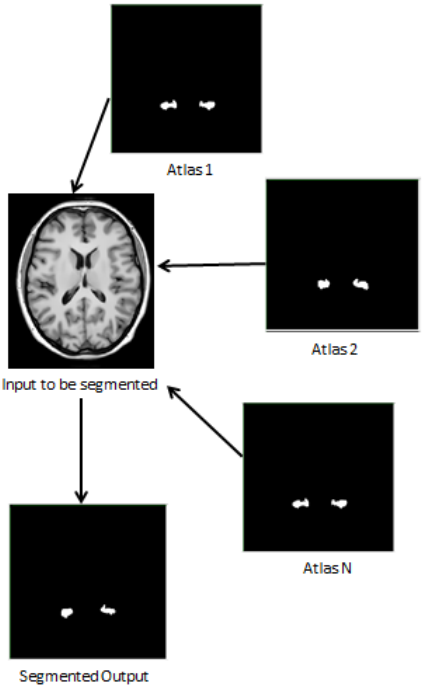


Figure 2.2: Representation of MAS technique for hippocampus segmentation

atlases, comparison of the structure of interest across the references and (label estimation) then label decision based on the majority. This step of combining label information from multiple atlases is often termed label fusion. There are multiple ways to achieve the objective in each of the steps mentioned and thus there are a variety of ways to implement MAS. In order to evaluate the segmentation performance, the Dice coefficient [7] is often used.

$$D.C = \frac{2TP}{2TP + FN + FP} \tag{2.1}$$

where TP is the true positive rate FP is the false positive rate and FN is the false negative rate. It measures the overlap between the segmented result and the ground truth. The Dice coefficient usually takes a value between 0 and 1; a perfect segmentation gives a value of 1.

2.2 Patch based methods

An effective discussion about the different aspects of the MAS technique would be incomplete without giving an overview of the idea of patch based methods. Inspired by the efficiency of these methods in image denoising [3, 4], patch based methods have begun to be used in different aspects of image processing.The concept of non local means, which is the basis of the patch based method, was introduced in [3, 4] to develop a non local means filter. This filter works by replacing the intensity of a voxel as a weighted average of the intensities of the neighboring voxels. The most similar neighbor voxel gets the highest weight and vice versa. This way of denoising shows good performance. A voxel considered along with its neighbors (patch) provides more useful information about itself with reference to the region of interest than a voxel that is considered individually. Especially in MAS, when a comparison is made between a reference image and an input to identify a structure, it is important to ensure that similar voxels are considered for comparison. This is important, especially to overcome errors during registration, in one to one voxel mapping. In patch based methods, a patch is defined as a window of voxels centered at a particular voxel - which is being studied. So the entire image volume can be considered to be made of patches. Processing operations are then performed at the patch level instead of the voxel level, with each patch being described by its central voxel.

2.3 Label estimation and fusion techniques

The preliminary step in MAS is atlas selection. To choose a given number of most similar atlases, the intensity difference with the input is computed and the most similar atlases with the lower intensity difference values are used. In our implementation, we use the sum of

squared intensity difference across all voxels in the input and reference, as suggested in [5], to choose a particular number of atlases. Once the atlases have been chosen, there are multiple ways to get the segmented output. This chapter gives a brief review of such techniques from the literature.

The essential idea in label estimation is to extract representative information from the atlases and make use of this representation to perform the labeling in the input. Hence, the different techniques basically differ in what they extract as representative information and how they extract the relevant information. Techniques also differ on how the label information from each atlas is integrated into the label decision on the voxel under study. It may be absolute as in the case of majority voting or might take into consideration other factors in which case there are weights associated with the votes to reflect the contributing factors, and hence called weighted majority voting.

In [10] active appearance (statistical description) of an object of interest in the given image (the shape and texture) is made use of to estimate labels and the implementation of the proposed technique utilizes T1, T2 and proton density (PD) weighted MR images of the subjects. The idea is to reconstruct the input using extracted information from the atlases and perform segmentation based on minimizing the error between the reconstructed image and the actual input. A signed distance function is used to represent the shape of the manually segmented object.The statistical texture variability (gray intensity variability) in the training dataset is computed as a function of the mean gray level values of the normalized T1, T2 and PD images and eigen vectors representing intensity variation in each category. A new image can be synthesized based on the given information of shape and texture and the difference between the input image and the corresponding synthesized image is minimized using a suitable optimization function. There is an increase in the performance of the technique with increase in the training data size although this dependency is reduced with

better alignment. Though effective, the application of this technique cannot be extended to atrophied target brain images unless the training data has a good representation of the same.

In [22], random local binary patterns are used. Label fusion is achieved on the basis of random local binary pattern based features extracted from a patch surrounding the target voxels and performing linear regression. The local binary pattern is a feature representing the variation in the image intensity values in the adjacent voxels in an image. A random local binary pattern (RLBP) feature vector is constructed by transforming the local binary pattern vector with a random transformation function. Once the RLBP feature vector is obtained, a linear regression model is constructed which is used to classify the input voxel into a particular label type. The five main parameters used in the regression model include search radius, patch radius (both based on the anatomic structure of interest), number of training samples, number of features and balance parameter. The segmentation accuracy was measured using a set of 9 metrics which show that the performance is comparable with other patch based techniques such as majority voting, non local patch based fusion and local label learning method. The advantages of this method in particular are that the inclusion of the randomness makes it robust to noise, illumination and contrast.

Another method that carries out label fusion based on feature extraction is atlas encoding using randomized forests [23]. An atlas forest encodes a single atlas by training one randomized classification forest exclusively on the data from the atlas. Every point is described by the appearance (given by the intensity) and a label prior map (PL) obtained from a probabilistic atlas used to get the location information. The features at a certain location x are a combination of deterministic local features (local intensity) and randomly instantiated non local features (obtained using a patch). Training is stopped at a certain tree depth (d=40). Each tree learns a class predictor based on the intensity image and label priors. The decision

for the input image is based on what a majority of these individual class predictors (trees) suggest. While this method involves phases of training and testing, the advantage lies in it requiring only a single registration. Like in any other application of random forest classifier, there is an inherent trade off between performance improvement and computation time with an increase in the training dataset size - both increase with the dataset size.

A limitation in many multiatlas segmentation methods is that the atlases used for label estimation need to be similar in terms of having at least the same number of labels. Iglesias et al [12] proposed a solution to the limitation to make use of information from atlases that may be generated from different protocols. The evaluation is based on the assumption that the registered atlases and the test image are generated by a statistical atlas of labels and intensities. A coarse label L (corresponding to manual delineation) is obtained from the fine label y in each of the atlases.

$$L_{ij} = f(y_{ij})$$

where f is a deterministic protocol specific function, i and j are voxel indices. The finer labels are estimated from the coarse labels using probability calculations involving the expectation maximization algorithm. The proposed multi-protocol label fusion was tested on 74 images from four different datasets and the performance measured by the Dice score shows that the technique achieves 3% to 6% improvement over the existing general label fusion methods (STAPLE[20], majority voting and local weighted voting[5],[16]). However, this algorithm is not very successful in labeling cortical structures with a great accuracy.

There are multiple novel methods available currently that are able to achieve good to very good segmentation results. Apart from the methods discussed above, there is a class of patch

based methods which we have analyzed in our study. These include the Gaussian weighted majority voting ([5],[16]), regression based weighted majority voting [21], non local STAPLE [1] which uses expectation maximization to perform segmentation, and random forest based classification to perform segmentation. There are a few factors that determine the outcome of the segmentation performance such as the registration technique used (linear / non- linear), the number of training samples and choice of parameters specific to the algorithm being used (e.g. patch size). Also, some of these methods are computationally expensive and could benefit from the utilization of parallel processing. In our study, we try to understand the method and the nature of parameter tuning required for each of the five techniques. The detailed description of these methods and the details of the analyses follow in the next chapters.

Chapter 3

Methodology

In this thesis, we compare the performance of five different techniques used to perform multi atlas based segmentation (MAS). The objective of this study is to understand, implement and compare certain patch based methods. While it makes sense to look at what the majority of references say about the label of a particular patch, a factor to be considered is how close/similar each of these references is to the input. Label decisions could be erroneous if a majority of the references are drastically dissimilar to the input. To overcome this issue, a weighted majority voting scheme is adopted instead of a direct majority voting where the weights indicate how close a reference is to the input. Three methods [5],[16],[21] included here differ in how these weights are defined. The random forest classification based segmentation method treats the segmentation as a classification problem at each voxel. It relies on the reference atlases to build the classification forest and uses this to classify the label for each patch. Non local STAPLE[1] uses expectation maximization to make a label decision. The details of each of the methods is discussed in the sections that follow.

3.1 Non local patch based segmentation - Label fusion using weighted majority voting

3.1.1 Gaussian weighted majority voting

Coupe et al [5] and Rousseau et al [16] independently proposed an elegant and a simple way of performing mutli-atlas based segmentation using weighted majority voting for label fusion. For each patch to be labeled in the input, a corresponding weight reflecting the similarity between the input patch and the patch under consideration in a given reference is computed. These weights are used in the estimation of the final patch label. If the patch centered at voxel x_i whose label has to be determined is represented as $P(x_i)$ and the reference patch being considered is $P(x_{s,j})$ with j as the central voxel in reference atlas s and the corresponding weight, representing the similarity between the two patches is $w(x_i, x_{s,j})$, the weighted label $v(x_i)$ for the voxel x_i is calculated as:

$$v(x_i) = \frac{\sum_{s=1}^{N} \sum_{j \epsilon V_i} w(x_i, x_{s,j}) y_{s,j}}{\sum_{s=1}^{N} \sum_{j \epsilon V_i} w(x_i, x_{s,j})} \tag{3.1}$$

$$L(x_i) = \begin{cases} 1 & \text{if } v(x_i) > 0.5 \\ 0 & \text{if } v(x_i) < 0.5 \end{cases} \tag{3.2}$$

where $y_{s,j}$ is the label given by the expert to the voxel $x_{s,j}$ at location j in reference atlas s and $L(x_i)$ is the final label.

The value of the weight is calculated as a function of the intensity difference. The basic idea is to give a higher weight to a sample that is closer to the input patch than the rest as done

in non local means filtering and explained earlier. The estimation is done using a Gaussian weighting function with a nonlocal means approach as defined in [3]. The expression to calculate the weight is then given by,

$$w(x_i, x_{s,j}) = \exp\left[\frac{-\|P(x_i) - P(x_{s,j})\|_2^2}{h}\right] \tag{3.3}$$

where $||.||_2$ is the normalized L2 norm computed between the intensity values of the voxels in patch P(x_i) and P($x_{s,j}$). The parameter h reflects the bandwidth of the Gaussian. The smaller the value, it becomes more selective, a fewer yet very similar patches are considered and are attributed higher weights. A larger value makes the selection more inclusive and tends towards the classical average.

The major difference between approaches in [5] and [16] is how this value of h is determined. In [5], h is estimated as a function of the intensity difference :

$$h_i = min(P_{x(i)} - P_{x(s,j)}) \tag{3.4}$$

where as in [16],h is a function of the standard deviation of the intensity difference distribution,σ.

$$h_i = 2N\sigma_i\beta_i \tag{3.5}$$

In this equation, N represents the number of voxels making up the patches used for comparison and β represents a scaling factor. The effects of how the choice of estimation for h impacts the overall performance of the algorithm has been studied and is discussed in later chapters.

3.1.2 Regression based weighted majority voting

Zhang [21] et al proposed a method of multiatlas based segmentation that utilizes the concept of regression to estimate weights and thus perform segmentation. In order to estimate the weights that reflect the similarity of patches, input patch is reconstructed from the set of reference patches. LASSO (Least angle shrinkage and selection operator) regression [15] is used for the same. LASSO is a particular kind of regression which is used when the number of variables or covariates are more than the number of observations. In this case, the covariates are the different patches coming from different atlases and the observations are the intensity values making up each patch. In addition to calculating regression coefficients, LASSO also selects the most relevant covariates by placing a constraint on the calculated coefficients. The regression coefficients become the weights that are needed to perform weighted label fusion. Weights represent the most relevant contribution from the reference atlases to estimate the label. The corresponding equation for the LASSO regression to address the segmentation problem is given by,

$$\min_{w_x} \frac{1}{2}\|Aw_x - b_x\|_2^2 + \lambda\|w_x\|_1 \tag{3.6}$$

where A represents a matrix made of intensity values from the reference patches from the reference atlases, b_x represents a vector of intensity values from the input patch centered at voxel x whose reconstruction is sought and w_x represents the regression coefficients, which are the weights we are interested in. Thus, the set of weights to be used for label estimation for each patch is determined from equation 3.6. λ places a constraint on the values the regression coefficients can take. If the values do not meet the constraint, the coefficients become zero and thus the corresponding covariates are not chosen. Once the values of the weights are estimated, the corresponding labels are calculated using the equations 3.1 and 3.2.

3.2 Non local patch based segmentation - Label fusion using STAPLE

The STAPLE (Simultaneous Truth And Performance Level Estimation) method uses a set of segmented reference atlases to provide a probabilistic estimate of the true hidden segmentation using an expectation maximization scheme. Asman et al [1] proposed to include STAPLE into the non local patch based segmentation method in order to combine the information from the weights - which essentially represent the probability of non local correspondence between the input voxel and a reference voxel - with the manual segmentation accuracy of the raters for the reference atlases. Thus,the decision is not based on the similarity to a reference patch but also on the accuracy of the reference label.

Mathematically, if W_{li} represented the probability that the voxel at location i has label l,

$$W_{li} = f(T_i = l|Y, I, T, \Theta^{(k)}) \tag{3.7}$$

where $f(T_i{=}l)$ is the apriori distribution of the underlying segmentation represented by T, Y represents the label decisions in the reference atlases, I represents the intensities in the different atlases and Θ quantifies the rater performance. The expression on the right in turn depends on the probability of non local correspondence between voxels i in the image and i' in the atlas s and is given by $\mathrm{v}(x_i)$ from equation 3.1. In the expectation step the probability W_{li} is calculated for given values of performance parameters (described by sensitivity and specificity, Θ) and probability of non local correspondence(α). In the corresponding equation listed below (E step), n represents the different labels possible. In this study there are two - label 0 when the structure is absent and 1 when it is present. In the maximization step, the values of the performance parameters are updated so as to maximize the likelihood of observing the estimated probability values. The two steps are repeated till the point where

no more update occurs.

E step:

$$W_{li}{}^{(k)} = \frac{f(T_i = l) \prod_j \Sigma_{i'P_s(i)} \Theta^k{}_{sl'l} \alpha_{si'i}}{\Sigma_n f(T_i = n) \prod_j \Sigma_{i'P_s(i)} \Theta^k{}_{sl'n} \alpha_{si'i}} \tag{3.8}$$

M step:

$$\Theta^{(k+1)}_{s,l'l} = \frac{\Sigma_i (\Sigma_{i' \in Pi: Y_{i'j} = l'} \alpha_{si'i}) W^{(k)}_{li}}{\Sigma_i W^{(k)}_{li}} \tag{3.9}$$

The final value of W_{li} is then used to make the label decision,using:

$$L(x_i) = \begin{cases} 1 & \text{if } W(_{li}) > 0.5 \\ 0 & \text{if } W(_{li}) < 0.5 \end{cases} \tag{3.10}$$

3.3 Random forest based segmentation

Here, the label decision is treated as a classification problem. A set of features sampled at a patch centered at a voxel is obtained. The random forest classifier is trained using features from patches making up the region of interest to be segmented. A random forest classifier contains a forest of decision trees. Each tree assesses a specific number of features randomly sampled, in order to determine a class label. This decision is based on a threshold that maximally differentiates between the two classes. Each individual tree is grown up to a particular depth and a final label or class for the given input set of features is decided based on decisions from all the trees in the forest. By including more trees and by randomizing the feature sampling at each decision tree, the accuracy of the classifier is improved. The number of trees and tree depth are parameters that usually impact the performance of the

random forest classifier. The features used in our method describe the location as well as texture of the patch. The texture is described using a set of features such as mean, median, entropy, Euclidean distance, standard deviation of intensity. The number of trees and the tree depth need to be tuned in order to get the most optimal values. For the each structure, the random forest is trained using corresponding reference atlas images and then assessed on a set of input images.

Chapter 4

Test Environment

Multi-atlas based segmentation techniques have been widely used to segment structures of interest in the brain. These are especially commonly used in studies such as the Alzheimer's disease neuroimaging initiative (`http://adni.loni.usc.edu/`) to gauge structural changes in the hippocampus. In order to assess the performance of the methods discussed in the previous chapter, we selected five structures in T1 MR brain images, with their reference segmentation from the Hammers dataset [9]. The structures of interest include the hippocampus, ventricles, caudate nucleus, palladium and thalamus. These five structures represent a considerable variation in terms of size, shape, intensity contrast with neighbors and thus constitute for a good way for assessing the robustness of each of the algorithms.

4.1 Dataset overview

The Hammers dataset [8],[9] consists of 30 T1 high resolution MR images with an isotropic voxel size of 0.94mm, obtained from a group of 15 young men and 15 women. Each image has a corresponding manually segmented atlas with up to 83 labels that represent specific

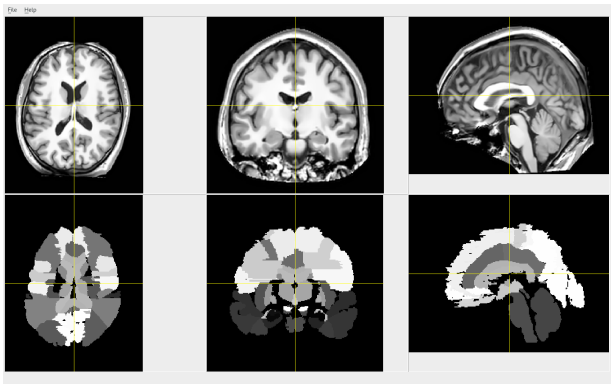


Figure 4.1: Processed Image - Greyscale image and segmented atlas

structures of the brain. Here, we use the hippocampus with labels (1,2) ventricles with labels (45,46), caudate nucleus with labels (34,35), palladium with labels (42,43) and thalamus with labels (40,41).

4.2 Dataset processing

The algorithms make use of information from the references in order to segment a structure of interest in the input. Hence, it is important to ensure that the references as well as the input are uniform in all respects to enable proper comparison. A set of pre-processing steps are performed to put all the images in a comparable standard. These steps mainly include filtering, registration and intensity normalization. Another major step in usual MR image pre-processing includes image inhomogeneity correction before intensity normalization. This has to be performed to remove noise, if any, due to bias field effect and make the image uniform. Bias field effect occurs potentially during image acquisition in MR imaging due to RF coil inhomogeneity and impacts the intensity values of the acquired image. Thus becomes important to remove such noises to ensure that the performance of the algorithm is not impacted. It is inspected for the presence of such noise using suitable visual tools (BRIAN Viewer) before application of correction technique. Since bias field correction has already been performed in images in the database, we skip this step. The pre-processing

portion can broadly be represented by the following flowchart:

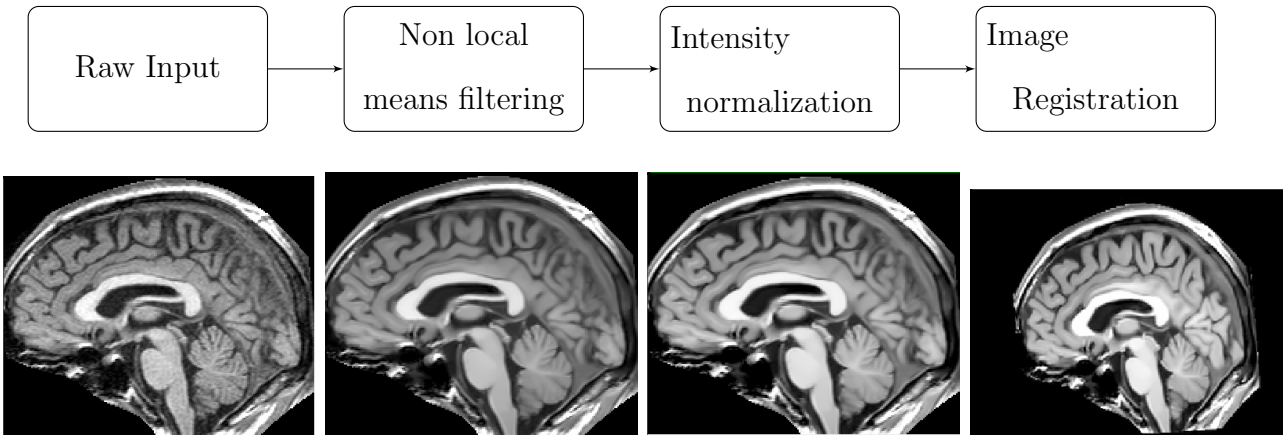


Figure 4.2: Pre-processing steps with representative images.

4.2.1 Non local means filtering

A basic noise removal filter is applied to all the images as a preliminary step. The filter used is a non local means filter [6], which is an extension of [3] for the 3D case. A patch based approach is used in this case, just like in the actual segmentation procedure. The filter functions by replacing the intensity value at a given voxel with a weighted combination of intensity values from its neighbors. If I represented the intensity values

$$I(x_i) = \Sigma_{x_j} w(x_i, x_j) I(x_j) \tag{4.1}$$

where the calculation of the weights is done in a fashion similar to that in Chapter 3. The weights reflect the similarity between the voxel of interest and the neighboring voxel under consideration.

4.2.2 Image registration

Image registration refers to the process of aligning an image to a standard coordinate space. One such standard for brain images is the MNI atlas. All 30 images are registered to the MNI atlas (`http://nist.mni.mcgill.ca`) using a six parameter transform [2]. Once the transformation has been performed, the registered images are checked for accuracy of registration by visually inspecting that anterior and posterior commissures, which are structures in the brain,are connected by a straight line. The registration step is important to ensure that the references and the input are aligned to enable comparison.

4.2.3 Intensity normalization

Intensity normalization is performed to ensure that the intensity scale is fixed for all images and the values are in a comparable range. To perform intensity normalization, we adopt the method described in [13]. There are two intensity scales - one that needs to be transformed or the original scale and the other - the standard scale. Equivalent 'landmarks' are identified on either scale. To identify the landmarks in the original scale , the intensity values at a fixed number of locations are sampled. When this number is large enough, these intensity values give a good representation of the underlying intensity composition. The sampled values are divided into three classes using a Gaussian mixture model and the mean for each individual class is estimated. For the new scale, equivalent class means are defined as required. Hence there are five landmarks used- the minimum and maximum intensity values and the three class means. A look up table is created to transform all sampled intensity values to the new scale. To do so, based on the intensity value to be transformed, the relationship between the corresponding pair of landmarks on the original scale and the standard scale is used. Once the look up table has been created, all intensity values in the input image can be transformed to the new scale using a simple transformation function.

4.3 Implementation details

Each of these methods is implemented using C++ and makes use of an existing library of functions (Brian 2.7.0 -Signal and Image processing tools). Once the images have been pre-processed, the segmentation techniques can be applied directly to the dataset. A pre-processed image registered to the MNI atlas usually has 256 voxels in each dimension. The structures used in this study occupy only a fraction of these voxels. Processing can be significantly reduced by introducing a mask to delimit the search range for the object of interest. By defining a region of interest, other regions in the brain which get imaged with similar intensity values are not considered and hence false detections are avoided. Similar to the choice of defining the region of interest in [5], we have defined the region of interest for a particular structure in our set up to be the union of all reference segmentations of the corresponding structure. As an example, the region of interest used for the ventricles is shown in the following figure.

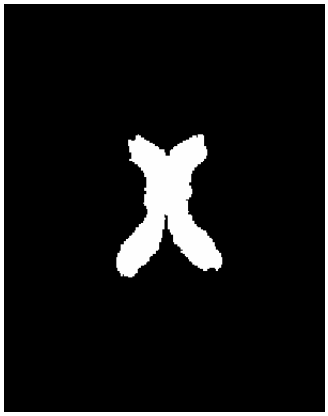


Figure 4.3: Region of interest used for ventricles.

In addition to defining a region of interest, a neighborhood size is defined that specifies a region in which patches are considered to perform segmentation. In order to avoid computation using distinctly dissimilar patches within the search neighborhood, a structural similarity measure [5] is calculated and used to threshold.

$$SSM = (\frac{2\mu_i\mu_{s,j}t}{\mu_i^2 + \mu_{s,j}^2})(\frac{2q_iq_{s,j}}{q_i^2 + q_{s,j}^2}) \tag{4.2}$$

Here, μ and q represent the mean and standard deviation of the intensity values, i represents the voxel to be labeled in the input, j is the corresponding voxel in the reference atlas s. These steps drastically decrease the computation time required by the procedure and at the same time enhance the accuracy of segmentation.

Method	Input Parameters	Additional Information
Gaussian weighted Majority Voting 1	Patch size,Number of atlases, Search neighborhood size	-
Gaussian weighted Majority Voting 2	Patch size,Number of atlases, Search neighborhood size, Beta	Intensity range for choice of beta
Regression based weighted Majority Voting	Patch size,Number of atlases, Search neighborhood size	Parameters for optimization routine as needed
Non local STAPLE	Patch size,Number of atlases, Search neighborhood size	Performance parameters of raters
Random Forest based Segmentation	Tree depth, Patch size, Number of trees	Random forest implementation specific parameters

Table 4.1: Testing environment

For the performance evaluation, the dataset is divided into two portions - one made of reference atlases and the other made of test images. 70% of the dataset is kept for the purpose of reference - all these 21 atlases were chosen randomly and the remaining 9 images were used for testing. The corresponding Dice coefficients obtained are recorded. Certain parameters have to be specified for each of the methods. These include the patch size, the neighborhood size, the number of reference atlases to be considered and the tree depth and number of trees for the random forest method. There is also some amount of prior knowledge about the dataset and the implementation details required in order to achieve good segmentation performance. A description of the testing environment is given above.A parametric study is performed to get a thorough understanding about the parameter tuning. In the following chapter, the results of the parametric study as well as the segmentation are described.

Chapter 5

Results and Discussion

Multiple factors impact the performance of an algorithm such as the dataset being used to study the algorithm, properties inherent to the structure chosen to perform the study as well as parameters that are chosen during the testing of the algorithm. We perform an algorithmic study keeping in mind all of these factors and hence including appropriate analyses in the study to capture information about all aspects of its performance.

All five algorithms are analyzed for their performance in segmenting a given structure in the MR image of a human brain while studying the influence of different parameters used in a particular algorithm choice. We use the Dice coefficient as a performance measure.

5.1 Non local patch based segmentation - label fusion using weighted majority voting

5.1.1 Weight calculation using Gaussian function

The different parameters that influence the working of the two methods that estimate weights using the Gaussian function are the neighborhood search size, the patch size and the number of atlases used as references to perform the segmentation. In addition in [16], β - the scaling parameter has to be set appropriately. The graphs on pages 28 and 29 illustrate how the Dice coefficient varies with each of these parameters in each of the methods.

As seen from the graphs, certain trends are common in both the methods. The Dice coefficient improves with an increase in search neighborhood. Increasing the search size widens the search area for identical patches from the references. This in turn is directly proportional to having more information to make a decision about a particular input patch. With an increase in the number of atlases, some improvement is seen in the performance in the method proposed by method 1 while a significant improvement is seen in method 2. A closer look at how the weights are calculated gives a good idea as to why this happens. In the first method, it is important to add atlases that are most relevant to the input. By merely adding more atlases which are not necessarily relevant to the input, no new information is extracted. Adding more dissimilar atlases will not change the width of the chosen Gaussian function as it it set to the minimum intensity difference, which in turn is a function of the most similar reference atlas and hence the performance efficiency remains mostly the same. However, in case of the second method, the width of the Gaussian is chosen based on the variance of the intensity difference distribution. This value definitely changes with increase in the number of atlases, thus making the performance efficiency sensitive to the number of atlases. This also explains how the patch size parameter influences the performance in

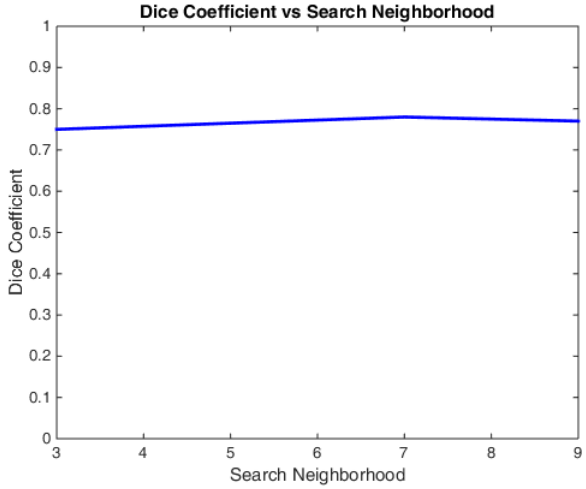
the two methods. In the first method, the width of the Gaussian remains unaffected with change in patch size. In the second method, by increasing the patch size, a larger variation intensity difference for a given atlas is included. As a result, the variance increases and the width of the Gaussian increases accordingly. This results in an averaging of weights throughout the similar and dissimilar patches end up getting similar weight values and hence the performance level decreases.

In addition to these parameters, tuning β in the second method is important as it represents how the variance is scaled. Using a synthetic dataset, the dependence of this β on the intensity range is obtained initially. It is seen that the β is a function of the intensity range and needs to be set appropriately to make the variance value effective in the calculation of the weights. For the brain dataset, the value of β is chosen over a relevant range and the most suitable choice is made from this range. The results obtained using different β values is listed in the following table.

Data	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}
04	0.72	0.75	0.79	**0.80**	0.79	0.73
05	0.61	0.70	0.74	**0.78**	0.76	0.61
10	0.71	0.78	**0.80**	0.77	0.72	0.27
12	0.68	0.72	0.78	**0.79**	0.78	0.73
Median	0.69	0.73	**0.78**	**0.78**	0.77	0.67

Table 5.1: Results with variation of β

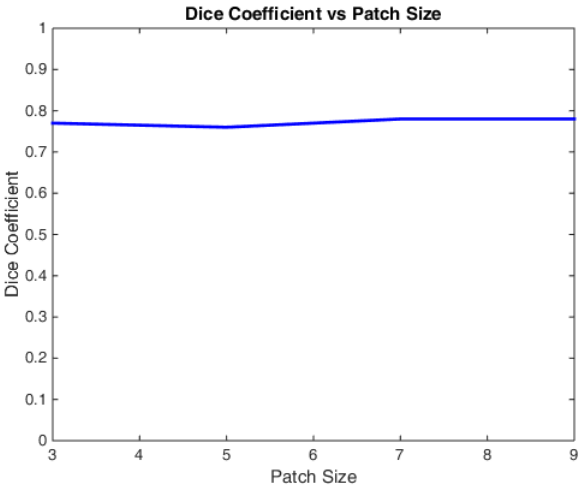
Figure 5.1: Parameter study for Gaussian weighted majority voting - Method 1



Variation of Search neighborhood

Search Neighborhood	Dice Coefficient
3	0.75
7	0.78
9	0.77

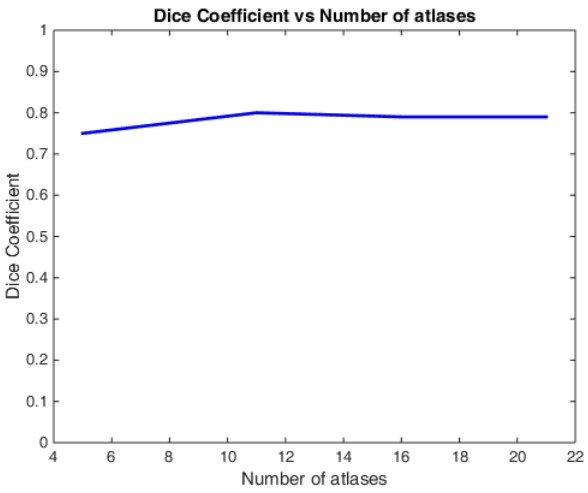
Variation of Search neighborhood



Variation of patch size

Patch Size	Dice Coefficient
3	0.77
5	0.76
7	0.78
9	0.78

Variation of patch size

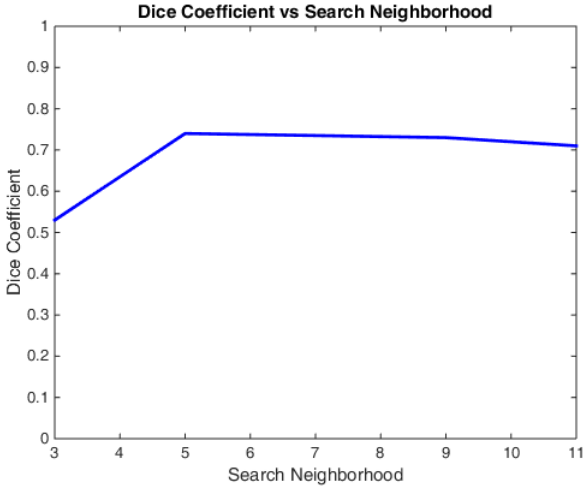


Variation of number of atlases

Number of atlases	Dice Coefficient
5	0.75
11	0.80
16	0.79
21	0.79

Variation of number of atlases

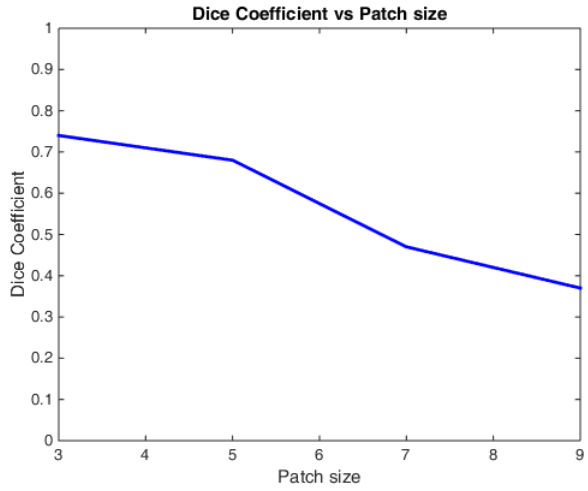
Figure 5.2: Parameter study for Gaussian weighted majority voting - Method 2



Variation of search neighborhood

Search Neighborhood	Dice Coefficient
3	0.53
7	0.74
9	0.73
11	0.71

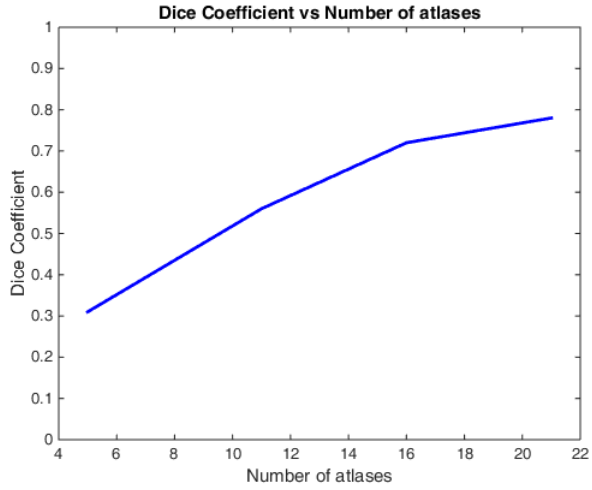
Variation of search neighborhood



Variation of patch size

Patch Size	Dice Coefficient
3	0.74
5	0.68
7	0.47
9	0.37

Variation of patch size



Variation of number of atlases

Number of atlases	Dice Coefficient
5	0.31
11	0.56
16	0.72
21	0.78

Variation of number of atlases

5.1.2 Weight calculation using regression

Sparse patch based reconstruction [21] makes use of regression, specifically LASSO (Least absolute shrinkage and selection operator)[15] regression to estimate weights which are then used to perform label fusion. Similar to the previous weighted majority voting technique, this method's performance is influenced by parameters such as the neighborhood search size, the patch size and the number of atlases. A key to interpreting the graphs in this case is to understand what exactly changes when the specified parameters are used.

When the search neighborhood or the number of atlases is modified, the number of patches is changed. In terms of the regression equation [3.6], this is equivalent to increasing the number of observations taken for calculating the regression coefficients(weights). In both cases, there is an initial increase in the performance efficiency with both parameters. However, with further increase in parameter values, there is no significant change in the performance efficiency. While the initial increase in the parameters could be contributing to making the samples more representative of the dataset, further increase in the number of samples does not qualitatively add to more information and hence does not yield a greater performance. In LASSO regression, the constraint λ is used exactly for this purpose. When the variables become highly correlated, the corresponding regression coefficients become highly variable and can take inaccurately large values. To avoid this, the constraint is placed such that coefficient values exceeding the constraint are made zero. Increasing the patch size is equivalent to increasing the number of values in vector b_x in equation [3.6]. Here, the entity/object to be reconstructed using regression is varied. When the patch size increases, the object being reconstructed changes. At a certain point, the object being reconstructed is very closely representative of the structure of interest. Further increase results in the reconstruction of a portion that includes the object of interest, but is not limited only to it. While this does not directly explain the trend in performance level, a look into the label fusion technique does. In

the reference segmented images, the label values for the structure of interest is set to 1 while all the other portions are given a 0. This portion labeled 0, around the structure of interest, which gets reconstructed during regression could or could not be representing regions similar in all images and there is no way to determine this from the available information. When label fusion is performed, the estimates calculated to make the decisions do not portray the scenario completely, often leading to lower performance efficiency.

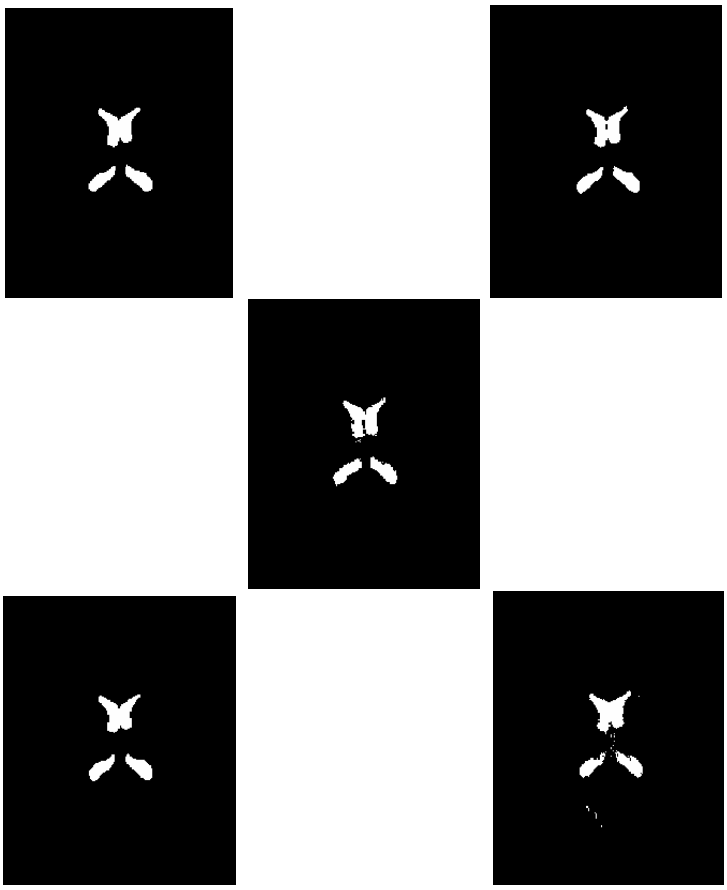
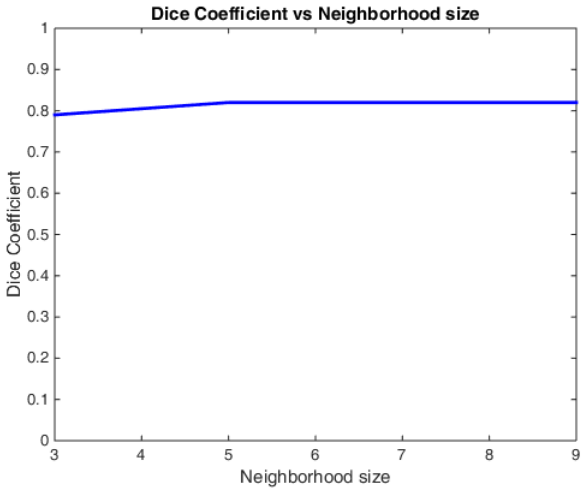


Figure 5.3: Ventricle segmentation output: A - Gaussian weighted majority voting 1, B -Gaussian weighted majority voting 2, C - Regression based weighted majority voting, D - Non Local STAPLE, E - Random Forest based segmentation

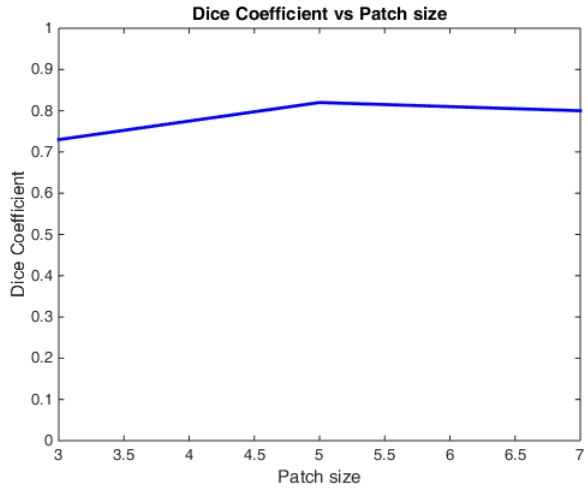
Figure 5.4: Parameter study for Regression weighted majority voting



Variation of search neighborhood

Search Neighborhood	Dice Coefficient
3	0.79
5	0.82
7	0.82
9	0.80

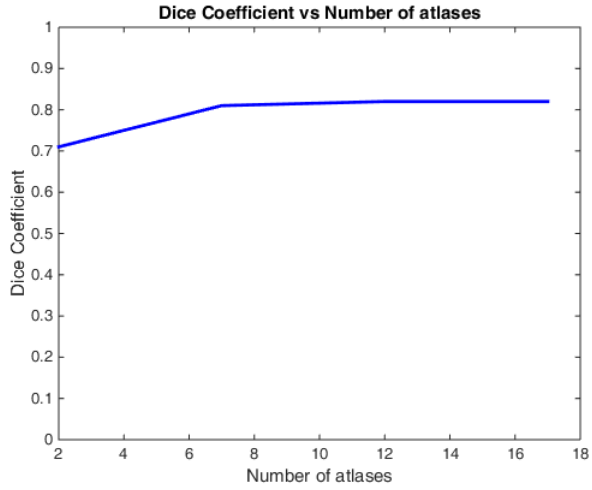
Variation of search neighborhood



Variation of patch size

Patch Size	Dice Coefficient
3	0.73
5	0.82
7	0.80

Variation of patch size



Variation of number of atlases

Number of atlases	Dice Coefficient
2	0.71
7	0.81
12	0.82
17	0.82

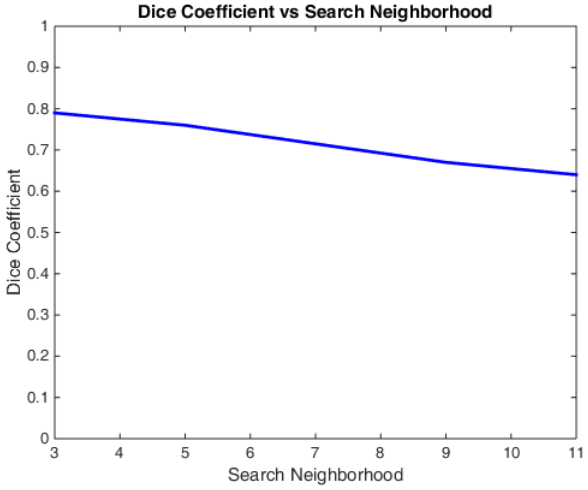
Variation of number of atlases

5.2 Non local patch based segmentation - label fusion using non local STAPLE

Here, the label decision is based on an expectation maximization scheme. The expectation step involves the calculation of probability that a particular voxel belongs to the structure of interest and in the maximization step, the likelihood of this happening, in terms of the experts' performance parameters is calculated. The variation of the performance efficiency with the search neighborhood size, the patch size and the number of atlases is shown in the following figure.

The performance efficiency decreases slightly with increase in the search neighborhood size. As the neighborhood size is increased, the patches being considered for comparison begin to have lesser and lesser correspondence to the input patch being considered. As the probability of correspondence is used in the expectation step to calculate the probability that a voxel belongs to structure of interest, including non relevant voxels does not contribute to information useful for the optimization routine. With increase in patch size as well as number of atlases, there is an initial increase in the performance efficiency which stabilizes later. Thus, merely adding information does not greatly improve performance. When the added information becomes less and less specific about the structure of interest under consideration, the performance does not improve.

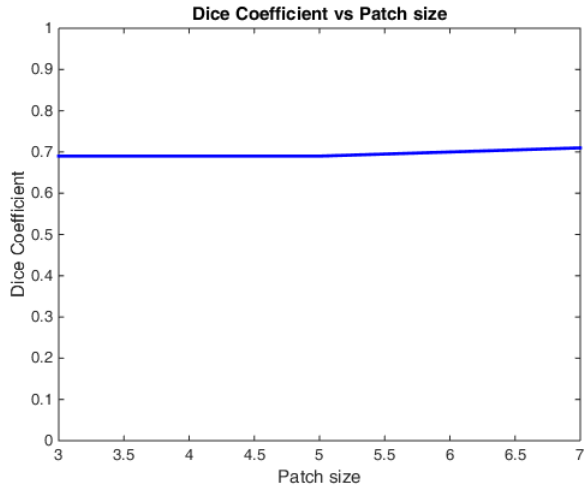
Figure 5.5: Parameter study for Non local STAPLE



Variation of search neighborhood

Search Neighborhood	Dice Coefficient
3	0.79
7	0.76
9	0.67

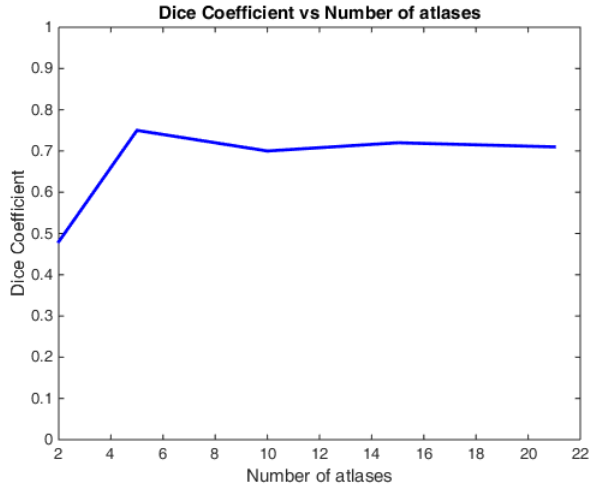
Variation of search neighborhood



Variation of patch size

Patch Size	Dice Coefficient
3	0.69
5	0.71
7	0.71

Variation of patch size



Variation of number of atlases

Number of atlases	Dice Coefficient
2	0.48
5	0.75
10	0.70
15	0.72
21	0.71

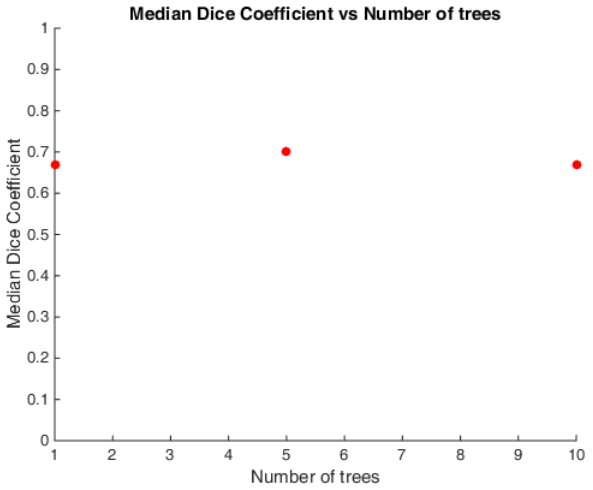
Variation of number of atlases

5.3 Random forest classification

The random forest classifier has two steps in its implementation - the training step and the testing step. The main parameters that are associated with the random forest classifier are the features, the tree depth, the number of trees and the training data size. The features that are made use of for the purpose of segmentation, as mentioned earlier, describe the location and texture of the input. Since the estimation of some of these features includes using the concept of patch, patch size is another parameter to be considered. To understand the dependence of the classifier performance on these factors, multiple iterations of classifier training with variation in the two parameters is done. The following graph represents the dependence of patch size, number of trees and tree depth on the classifier performance.

As can be seen, the performance improves slightly with increase in patch size and then decreases. This could be attributed to the fact that increasing the patch size beyond a value makes it less descriptive of the particular voxel, which in turn translates to features not very specific about voxels making up the structure of interest. Similarly increasing the number of trees and tree depths, the performance initially increases but then drops later, most likely, due to overfitting. It is important to ensure that the tree depth and number of trees parameters are set to optimal values.

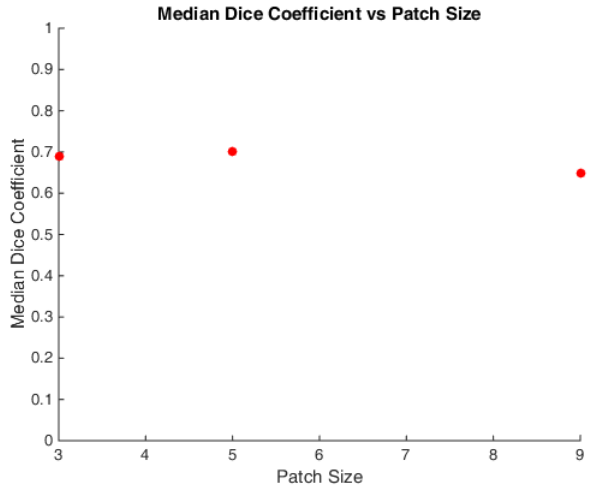
Figure 5.6: Parameter study for Random forest based segmentation



Variation of number of trees

Search Neighborhood	Dice Coefficient
1	0.67
5	0.7
10	0.67

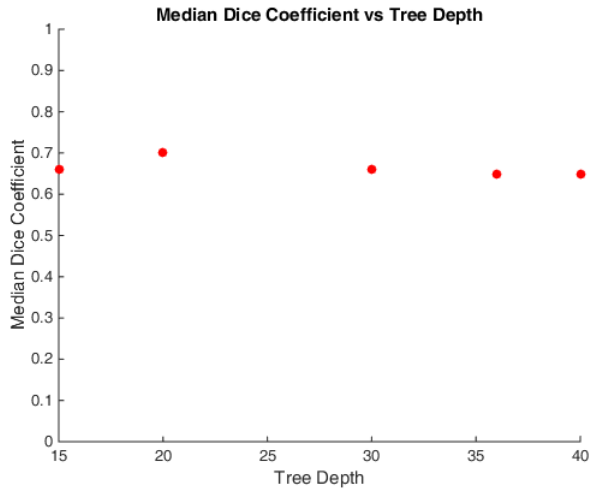
Variation of number of trees



Variation of patch size

Patch Size	Dice Coefficient
3	0.69
5	0.7
9	0.65

Variation of patch size



Variation of tree depth

Number of atlases	Dice Coefficient
15	0.60
20	0.70
30	0.66
36	0.65
40	0.65

Variation of tree depth

5.4 Performance on the brain dataset

Each of the five methods is used to segment five structures of interest in the brain. The patch size and the neighborhood size are chosen appropriately taking into consideration the structure of interest as well as the method applied. The Gaussian weighted voting methods as well as the random forest method make use of 21 reference atlases. The regression based weighted voting method makes use of 10 atlases and the label fusion using non local STAPLE method makes use of 12 atlases. For segmenting the ventricles using the non local STAPLE algorithm, the number of atlases is set to 7. The following graphs show the performance of the five methods in their ability to segment the structures. The corresponding values are tabulated in A.1 - A.5.

Structure/ Method	Hippocampus	Palladium	Thalamus	Caudate Nucleus	Ventricles
GW M1	**0.80**	**0.80**	**0.88**	**0.87**	**0.87**
GW M2	**0.80**	0.78	**0.88**	**0.87**	0.86
Reg W	0.79	**0.80**	**0.88**	0.83	0.83
NLS	0.72	0.75	0.84	0.85	0.79
RF	0.70	0.70	0.84	0.81	0.83

Table 5.2: Median Dice Coefficient Values

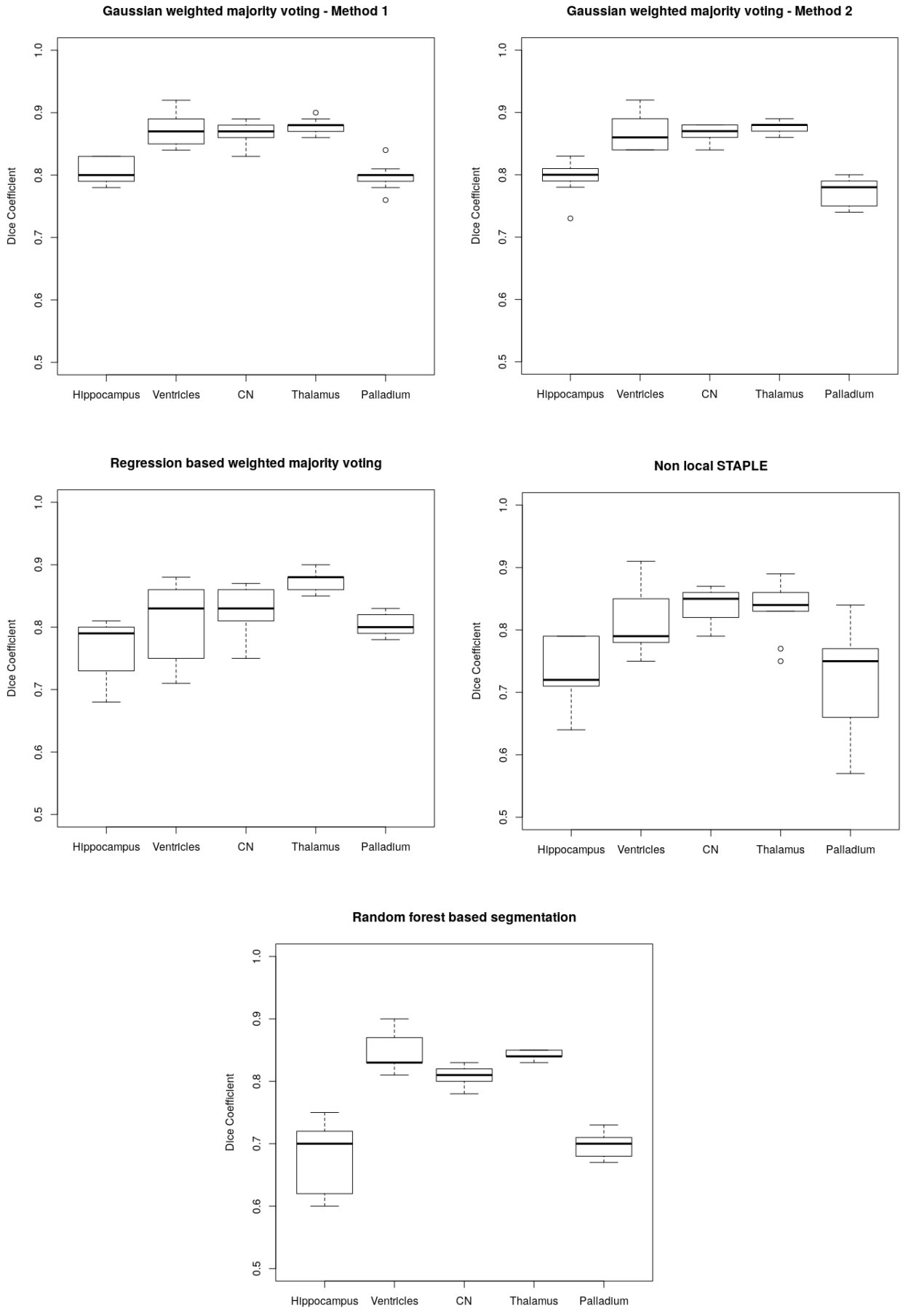


Figure 5.7: Results of segmentation of the brain dataset.

The most striking result from the graphs shown above is that the performance of each of the methods in segmenting the ventricles, caudate nucleus and thalamus is better than in segmenting the hippocampus and palladium. The average median dice coefficient obtained for hippocampus and palladium is approximately 0.76 while that for the rest is approximately around 0.85, which indicates a 12% increase. This is not surprising since the ventricles, caudate nucleus and thalamus are easier to segment even manually due to higher contrast with its neighbors as against hippocampus and palladium which are relatively less discernible from the neighbors.

All three methods that make use of label fusion using majority weighted voting have higher median Dice coefficient values than label fusion using non local staple as well as random forest. For hippocampus and palladium, all three have a median dice coefficient value of approximately 0.80 while the other two have a value around 0.70. Similarly in case of the thalamus, the three methods have a median dice coefficient value of 0.88 while the the other two have a value around 0.84. In case of the caudate nuclues, the methods using Gaussian weighting perform the best followed by the ones using regression based weighting, label fusion using non local STAPLE and random forest classification. In the case of ventricles, the Gaussian weighted voting again performed the best followed by random forest classification, regression based weighting and non local STAPLE. In case of the last three structures, the median Dice coefficient values vary little(less than 10% difference) between the different methods as against the first two structures. In terms of over all consistent performance, both the Gaussian weighted voting methods perform well for all the different structures of interest.

5.5 Summary and Conclusion

This work compares and contrasts the performance of five image segmentation routines in segmenting structures in human brain MRI. The median Dice coefficient values obtained above suggest that the Gaussian weighting based label fusion method generally tends to perform better than the rest. However, there are several factors which need to be taken into account before a specific conclusion can be made. One such important factor is the computational time involved. All methods excepting random forest classification take a few hours to finish. In case of random forest classification, a significant amount of time is spent on training the classifier which is certainly a disadvantage when rapid results are required. Even with the given results, one cannot exactly claim a best method. For instance, two of the five methods make use of lesser number of atlases and still give comparable performance in terms of the median dice coefficient values. While there is a good scope for improvement in their performance using more number of atlases, a disadvantage in doing so would be the significantly increased computational cost involved. Similarly, the random forest classification method gives a better performance in case of structures which can be differentiated better than the neighbors. If the feature space is exploited completely and features more descriptive of the hippocampus and palladium with respect to their neighbors are used, the performance efficiency could is improved. Even within the Gaussian weighting based methods, the performance of method 1 does not increase significantly with an increase in the number of atlases while in case of method 2, it does. Hence, if there were more number of reference atlases to choose from, setting the Gaussian width using method 2 would represent a better way of using all information available to get better results.

All of the five methods are suitable to perform image segmentation. Their performance can be enhanced greatly by tuning the parameters appropriately. It is advisable to choose an algorithm based on the purpose of the segmentation task. On the clinical application side, relatively faster processing would be required and the best method to use would be the

Gaussian weighting based majority voting. On the other hand, applications on the clinical research side which focus more on gathering specific details could gain by exploiting methods such as the random forest classifier. Computational time and space as well as the nature of the dataset - how similar or dissimilar the atlases are also play a crucial role in identifying the algorithm to be used. The main focus of further study in this direction should be to analyze some of these methods, specifically the random forest classifier as well as the regression based methods in greater detail for variations in their implementation as well as tackle the challenges related to computational performance of each of the algorithms.

Bibliography

[1] A. J. Asman and B. A. Landman. Non-local STAPLE: an intensity-driven multi-atlas rater model. *Med Image Comput Comput Assist Interv*, 15(Pt 3):426–434, 2012.

[2] B. B. Avants, C. L. Epstein, M. Grossman, and J. Gee. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Medical image analysis*, 12(1):26–41.

[3] A. Buades, B. Coll, and J. Morel. A non-local algorithm for image denoising. 2:60–65, June 2005.

[4] A. Buades, B. Coll, and J. M. Morel. Image denoising methods. a new nonlocal principle. *SIAM Review*, 52(1):113–147, 2010.

[5] P. Coupe, J. V. Manjn, V. Fonov, J. Pruessner, Robles, M., and D. L. Collins. Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation. *NeuroImage*, 54(2):940 – 954, 2011.

[6] P. Coupe, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot. An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images. *IEEE transactions on medical imaging*, 27(4):425–441, 2007.

[7] L. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[8] A. Hammers, R. Allom, M. J. Koepp, S. L. Free, R. Myers, L. Lemieux, T. N. Mitchell, D. J. Brooks, and J. S. Duncan. Three-dimensional maximum probability atlas of the human brain, with particular reference to the temporal lobe. *Human brain mapping*, 19(4):224–47.

[9] A. Hammers, C. Chen, L. Lemieux, R. Allom, S. Vossos, S. L. Free, R. Myers, D. J. Brooks, J. S. Duncan, and M. J. Koepp. Statistical neuroanatomy of the human inferior frontal gyrus and probabilistic atlas in a standard stereotaxic space. *Human brain mapping*, 28(1):34–48.

[10] S. Hu, P. Coupe, J. C. Pruessner, and D. L. Collins. Appearance-based modeling for segmentation of hippocampus and amygdala using multi-contrast MR imaging. *Neuroimage*, 58(2):549–559, Sep 2011.

[11] J. E. Iglesias and M. R. Sabuncu. Multi-atlas segmentation of biomedical images: A survey. *Medical image analysis*, 24(1):205–219, 2015.

[12] J. E. Iglesias, M. R. Sabuncu, I. Aganj, P. Bhatt, C. Casillas, D. Salat, A. Boxer, B. Fischl, and K. Van Leemput. An algorithm for optimal fusion of atlases with different labeling protocols. *Neuroimage*, 106:451–463, Feb 2015.

[13] L. G. Nyul, J. K. Udupa, and X. Zhang. New variants of a method of mri scale standardization. 19(2):143–50.

[14] N. Ramesh, J. H. Yoo, and I. K. Sethi. Thresholding based on histogram approximation. *IEE Proceedings - Vision, Image and Signal Processing*, 142(5):271–279, Oct 1995.

[15] T. Robert. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[16] F. Rousseau, P. A. Habas, and C. Studholme. A supervised patch-based approach for human brain labeling. *IEEE Trans Med Imaging*, 30(10):1852–1862, Oct 2011.

[17] N. Sharma and L. M. Aggarwal. Automated medical image segmentation techniques. *Journal of Medical Physics / Association of Medical Physicists of India*, 35(1):3–14, 2010.

[18] N. Sharma, R. A. K., S. Sharma, K. K. Shukla, S. Pradhan, and L. M. Aggarwal. Segmentation and classification of medical images using texture-primitive features: Application of bam-type artificial neural network. *Journal of medical physics / Association of Medical Physicists of India*, 33(3):119–126, 2008.

[19] Z. Wang, A. Guerriero, and M. De Sario. Comparison of several approaches for the segmentation of texture images. *Pattern Recognition Letters*, 17(5):509 – 521, 1996.

[20] S. K. Warfield, K. H. Zou, and W. M. Wells. Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation. *IEEE transactions on medical imaging*.

[21] D. Zhang, Q. Guo, G. Wu, and D. Shen. Sparse patch-based label fusion for multi-atlas segmentation. 7509:94–102, 2012.

[22] H. Zhu, H. Cheng, and Y. Fan. Random local binary pattern based label learning for multi-atlas segmentation. *Proc. SPIE*, 9413:94131B–94131B–8, 2015.

[23] D. Zikic, B. Glocker, and A. Criminisi. Atlas encoding by randomized forests for efficient label propagation. *Med Image Comput Comput Assist Interv*, 16(Pt 3):66–73, 2013.

Appendix A

Appendix Title

Data	Hippocampus	Ventricles	Caudate Nucleus	Thalamus	Palladium
04	0.80	0.89	0.88	0.87	0.80
05	0.79	0.85	0.89	0.89	0.79
10	0.79	0.85	0.87	0.88	0.80
12	0.83	0.92	0.86	0.88	0.79
20	0.78	0.90	0.87	0.87	0.84
21	0.83	0.84	0.88	0.86	0.80
23	0.83	0.89	0.83	0.88	0.78
25	0.81	0.85	0.86	0.88	0.81
30	0.80	0.87	0.89	0.90	0.76

Table A.1: Dice Coefficients - Gaussian weighted majority voting - Method 1

Data	Hippocampus	Ventricles	Caudate Nucleus	Thalamus	Palladium
04	0.79	0.89	0.88	0.87	0.79
05	0.80	0.85	0.88	0.88	0.74
10	0.81	0.84	0.86	0.88	0.80
12	0.83	0.92	0.86	0.88	0.78
20	0.81	0.89	0.87	0.87	0.80
21	0.81	0.84	0.88	0.88	0.77
23	0.73	0.86	0.84	0.86	0.75
25	0.78	0.84	0.84	0.89	0.78
30	0.80	0.87	0.88	0.89	0.75

Table A.2: Dice Coefficients - Gaussian weighted majority voting - Method 2

Data	Hippocampus	Ventricles	Caudate Nucleus	Thalamus	Palladium
04	0.80	0.86	0.86	0.85	0.78
05	0.73	0.75	0.85	0.88	0.79
10	0.68	0.75	0.83	0.87	0.80
12	0.79	0.88	0.81	0.88	0.82
20	0.75	0.71	0.75	0.86	0.83
21	0.79	0.83	0.87	0.90	0.80
23	0.69	0.83	0.83	0.86	0.78
25	0.80	0.79	0.79	0.88	0.81
30	0.81	0.86	0.86	0.88	0.83

Table A.3: Dice Coefficients - Regression based weighted majority voting

Data	Hippocampus	Ventricles	Caudate Nucleus	Thalamus	Palladium
04	0.72	0.87	0.87	0.84	0.75
05	0.71	0.79	0.86	0.87	0.73
10	0.68	0.78	0.81	0.75	0.80
12	0.79	0.91	0.85	0.85	0.63
20	0.71	0.76	0.85	0.83	0.84
21	0.79	0.79	0.86	0.83	0.77
23	0.64	0.84	0.82	0.86	0.66
25	0.78	0.75	0.79	0.79	0.77
30	0.79	0.85	0.86	0.89	0.57

Table A.4: Dice Coefficients - Non local STAPLE

Data	Hippocampus	Ventricles	Caudate Nucleus	Thalamus	Palladium
04	0.60	0.87	0.80	0.85	0.73
05	0.62	0.83	0.83	0.84	0.68
10	0.62	0.83	0.83	0.84	0.68
12	0.75	0.90	0.80	0.84	0.71
20	0.70	0.85	0.81	0.85	0.73
21	0.75	0.81	0.81	0.85	0.71
23	0.71	0.89	0.78	0.83	0.67
25	0.68	0.83	0.80	0.84	0.70
30	0.72	0.83	0.82	0.85	0.67

Table A.5: Dice Coefficients - Random forest based segmentation

Method	Hippocampus	Ventricles	Thalamus	Caudate Nucleus	Palladium
GW M1	11	7	7	7	11
GW M2	11	7	7	7	11
Reg W	5	5	5	5	5
NLS	7	5	7	7	7

Table A.6: Configuration - Search Neighborhood

Method	Hippocampus	Ventricles	Thalamus	Caudate Nucleus	Palladium
GW M1	7	7	7	7	7
GW M2	7	7	7	7	7
Reg W	5	5	5	5	5
NLS	5	5	5	5	5
RF	5	5	5	5	5

Table A.7: Configuration - Patch Size