Master's Thesis
A method for classification of labeled brain images

Guillermo Robles Fernández [RFD138]
Supervisor: Sune Darkner
University of Copenhagen

August 6, 2016

Abstract

This thesis presents and evaluates four different multi-atlas segmentation approaches for brain MRI images; *Expectation-Maximization with Maximum likelihood(ML) and Maximum a Posteriori(MAP) estimators, and expands it with Markov Random Fields*, *Non-local STAPLE*, *Attribute Similarity and Mutual Saliency Weighting* and *Support Vector Machine*. The experiments have been carried out using human brain templates in two different atlases; Miccai and MGH. The human brain templates provide a common frame of reference through all the brain images in the atlas, they form local clusters of similar anatomy. The segmentation methods improve the labelling by focusing in the local region of the target voxels. The approaches have been implemented by machine learning techniques and classification methods in order to create a segmentation of an unseen brain.

The results show that is possible to achieve a better segmentation by implementing sophisticated label classification methods. Furthermore the experiments show a segmentation improvement in the data-set MGH over the template based labelling.

Keywords. segmentation; label classification; image registration; templates; MRI

Contents

1 Introduction **4**

2 Literature review **6**

3 Image Registration **11**

4 Image Segmentation **14**
4.1 Expectation-maximization (EM) 14
4.1.1 EM implementation 15
4.2 Maximum Likelihood Estimator & Markov Random Fields 17
4.3 Maximum a Posteriori 20
4.4 Non local Spatial STAPLE 21
4.4.1 STAPLE 21
4.4.2 NLS method 22
4.4.3 MapReduce 23
4.5 Attribute Similarity and Mutual-Saliency Weighting 24
4.6 Support Vectors Machine 25

5 Experiments **28**
5.1 Introduction 28
5.1.1 Image Registration 28
5.1.2 Dice score coefficient 30
5.2 Expectation-maximization ML and MAP 31
5.3 Expectation-maximization MRF 37
5.4 Non local STAPLE 43
5.5 Attribute Similarity and Mutual-Saliency Weighting 50
5.6 Support Vectors Machine 54

6 Conclusion **61**

1 Introduction

Label classification of brain images has gained interest in recent years. New methods are necessary to improve labelling accuracy across the images. This is a fundamental task in neuroimaging. For instance, in dementia cases a correlation between regions volumes, like hippocampal atrophy and disease progress has shown to be an indicator for Alzheimer's disease. [6]. The application and importance of this task motivate a search for better approaches to the labelling of brain images.

This master thesis is focused on researching and evaluating different approaches to label classification and comparing them. A challenge in label classification is to register properly the target images in the atlas which is important in order to reduce structural differences and improve correspondence across images. The segmentation process classifies brain images into classes or tissues. Fully-automated methods, rather than manual ones, provide robust and accurate segmentations and save time when working with large data sets.

The goal of this master thesis consists on performing different segmentations methods in order to evaluate the generated brain templates [4], from two data set of MRI brain images: Miccai and MGH. The templates are effective in many ways. They offer a common coordinates system that increases local correspondence between the volumes. The templates create a generalization of a certain population such as the elderly or children, and aids to perform more precise comparisons. Moreover, the templates provide a prior for different template based segmentation methods. On the other hand, the template is not a real brain image and could not be effective for a label classification method because the template is an average of multiple brains and not a single brain.

In general the methods are based on the local intensity, neighbourhood-dependent local weighting, label information and intensities similitude. The straightforward approach is **majority voting**, where the label of a target voxel is estimated by *counting* the most common label in the registered atlas, so the highest frequency label is the one predicted for the target voxel. In despite of the simplicity, majority voting has some disadvantages. It does not provide a unique majority label, it does not produce a label estimation of the target voxel and also it does not provide any relation with the neighbouring voxels. The alternatives combine different strategies to achieve a better classification such as:

Expectation maximization-Based EM is a widely used method for label classification. On a iterative way, EM calculates the maximum likelihood or maximum posterior probability label of a voxel. It can be extended by including a global and stationary Markov random fields [6], which would ensure the consistency of the neighbourhood structures. Moreover it employs a local weighting scheme to define subject-specific probabilistic atlas.

Non local STAPLE (NLS) Several methods are based on a direct correspondence between the intensity and label information, which makes them dependent on a high quality image registration and on a large data set of brain images. NLS, [19], uses non local means to consider some complex image characteristics such as noise structure and, spatially varying correspondence. NLS decomposes the 3-dimensional images in a set of volumetric patches, then determines the underlying image structure by quantifying the similarity and correspondence between all volumetric patches. NLS determines which class should be a target voxel given a perfect correspondence with the target and the atlas. NLS resolves imperfection correspondence from the registration process.

Attribute-based similarity and matching reliability metrics In label classification many methods assign higher weights to the atlases that are more similar to the target(such as EM-MRF). Attribute-based similarity and matching reliability metrics, [21], provide a way to measure the similitude differences between the voxels, it propose that the matching of two voxels is complete if they are similar to each other and not more similar to any other voxel in the neighborhood. By using Gabor attribute vector, which provides more information than voxel intensity, this method reflects the underlying geometric and anatomical characteristics.

Support Vector Machine SVM is a common method used on data classification [26]. It is able to build a model that classifies voxels by using a linear kernel function. Thus it classifies each voxel from an unknown labelled brain image. For training, the model uses different features such as voxel location, voxel intensity and the voxel neighbourhood intensity similarity. Moreover SMV is applied independently over small patches of the brain image, in this way a better accuracy in label prediction is reached.

2 Literature review

Multi-class image segmentation is a extensively used method in the area of MRI brain analysis. Various algorithms have been proposed in order to fit a better estimator into the analysed brain images such as morphological operations, edge detection, fuzzy c-means and probabilistic methods. LoAD([7]) and an article by Legid and colleagues [6] describe a probabilistic method that fits with the expectation maximisation (EM) algorithm, and includes a spatial consistency model based on Markov Random Field(MRF). The described method is a global brain segmentation. The proposes MFR energy matrix improves the topological characteristics of the segmentation and reduces the partial volume layer thickness.

Multi-atlas label propagation and intensity-based refinement with Expectation-maximization algorithm, MALP-EM [12], proposes a subject-specific probability. Firstly, the authors calculate the atlas transformation with a non-rigid registration method based on free-form deformations(FFD) [13] [14], which is followed by a preceding rigid and affine alignment. The probabilistic atlas is created with a locally weighted multi-altas fusion strategy [15], with a Gaussian weighted sum of squared differences on rescaled and intensity-normalized images. By employing the method of van Leemput and colleagues [16], the segmentation of the target MRI images is estimated. They have incorporated the probabilistic model into the Expectation-Maximization algorithm. Finally Ledig and colleagues have applied a regularization method on the segmentation based on Markov Random Fields(MRF)([7]).

Simultaneous Truth And Performance Level Estimation (STAPLE) ([18]) is another example of the EM algorithm. The expectation of the complete data log likelihood with respect the density is calculated and the performance parameters are computed by the maximum likelihood or maximum a posteriori estimation. Non local STAPLE ([19]) reformulates the STAPLE algorithm and provides a non-local means approach which aids to solve improprieties in the registration process and provides a method that does not need large amount of atlas images. NLS demonstrates theoretical and empirical improvements over the STAPLE family algorithms, in non-local-means the images is divided into small patches and the similarity of these patches is analysed in order to learn the correspondence of the image structures. Non-local-means framework has been used for de-noising images [23] and, for image segmentation ([24]). NLS approaches a model that fits a perfect correspondence between the voxels in the atlas and the target. In this way the proposed method is able to improve the segmentation process.

Attribute Similarity and Mutual-Saliency Weighting for Registration and Label Fusion, [21], proposes the registration and segmentation of the atlas images which uses an attribute-based similarity metric and a mutual-saliency-based reliability metric. The method uses Gabor attributes to analyze the correspondence of the voxels by their similarity and the reliable similarity with the mutual-saliency metric. The objective is to determine a higher weight to the atlas voxels that are more reliably similar to the target

voxels.

The results from studies employing these methods are presented below.

Expectation maximization MRF

In [6] *Multi-class brain segmentation using atlas propagation and EM-based refinement* the authors evaluated 30 manually segmented MRI brain scans where, each brain image had 83 different labels. The authors ran the experiment using a leave-one-out strategy. The EM algorithm terminates once the convergence is complete. The authors determine that such convergence occurs when the change of the likelihood is less than 0.5% from the last iteration.

The authors followed pre-processing steps in order to fusion the labels:

1º Since the structures of the brain appear in both parts, and they have the same intensity both structures are fused to the same label which reduces the quantity to 46 classes.

2º White matter(WM) structures and the *corpus callosum* are merged into one label, and all cebrospinal fluid(CSF) structures are then merged with the *ventricles*. At the end, there were 41 classes, 8 subcortical grey matter(GM), 29 cortial GM, and GM fractions of the brain-steam and of the cerebellum.

The table below presents the computed mean dice coefficient score over the EM refinement and local weight fusion(LWF).

	EM	LWF
hippocampus	0.823	0.836
41 classes	0.869	0.841
39 all GM classes	0.825	0.789
29 cort. GM classes	0.799	0.757

Dice score coefficient increases in cortical GM structures when it is compared EM refinement and LWF.

LoAD

In LoAD [7], two experiments have been carried out: Atlas dependency evaluation and segmentation evaluation.

Atlas dependency evaluation A set of 40 images, 20 images from patients diagnosed with AD and another 20 images from healthy subjects with matched age and gender, were selected from the ADNI database. These 40 images were segmented by two different anatomical atlases, one is the `ICBM452` and the other one is `MNI305`. The `ICBM452` was

built by a non-rigid registration and averaging 452 normal MRI scans, `MNI305` was created by a affine-registration of 305 norrmal MRI scans. `ICBM452` and `MNI305` are representative of a normal population, the difference is the way that the registration was created for both atlases. The table below show the dice score coefficients over the two evaluated sub-sets of images; **AD** with patients diagnosed with AD, and **Control** a set of 20 images that match the age and gender of every patient in the AD group. Moreover the method has been evaluated with 4 different `relaxation factors(RF)`.

	AD	Controls
RF:0	0.906	0.924
RF:0.33	0.91	0.935
RF:0.66	0.924	0.935
RF:1	0.924	0.935

The results show when a relaxation factor is applied in the AD group, the median dice score increases. However, in the other group, the median dice score increases from $RF : 0$ to $RF : 0.33$ then it keeps equal.

Segmentation evaluation Twenty images sets were evaluated from the data-base `BrainWeb` [1]. Each set contains a T1-weighted image and its corresponded ground truth probabilistic atlas. The simulated data was accomplished by a spoiled FLASH sequence with $TR = 22ms$, $TE = 9.2ms$, $\alpha = 30$ž and 1-mm isotropic voxel size with simulated 3% noise and 20% INU [8].

The twenty images were segmented using the proposed method in LoAD, and another three methods in order to compare the results between all of them, *SPM8* [10], *FAST* [11] and *FASTASM* [9]. *SPM8* is a segmentation method which, is an instance of EM with a bias correlation. *FAST* is also an EM based method that incorporates an MRF approach in order to add consistency to the spatial neighbourhood.*FASTASM* is fuzzy c-means segmentation method.

The dice score has been calculated in order to measure the overlap of the estimations of the methods and the real segmentation, a $RF = 1$ has been used. The average dice score for the 20 images are shown in the next table.

LoAD	FAST	SPM	FANTASM
0.959	0.941	0.929	0.927

MALP-EM

MALP-EM([12]) was evaluated by using the Open Access Series of Imaging Studies, *OASIS*, database, with a total of 20 MRI brain images. The obtained dice coefficient

[1] http://www.bic.mni.mcgill.ca/brainweb

for cortical regions was 73.28% and for sub-cortical regions was 82.52%, which makes a total average dice coefficient of 75.76%.

The matrix G that defines the connectivity of the tissues has a shape of 139×139, and it is created with the next formula;

$$G(i,j) = \begin{cases} 0 & \text{if i = j} \\ 1.0 & \text{if the tissue i and j share a boundary} \\ 1.5 & \text{if i and j are distant} \end{cases}$$

SPATIAL STAPLE

In *Characterizing Spatially Varying Performance to Improve Multi-Atlas Multi-Label Segmentation* [17] introduces Spatial STAPLE algorithm. Spatial STAPLE modifies STAPLE in order to include spatially varying performance levels.

Spatial STAPLE has been evaluated in contrast with majority vote approach and STAPLE. Fifteen images has been registered to a set of 24 target images using *Vectorized Adaptive Bases Registration Algorithm(VABRA)*. Dice Similarity Coefficient has been used as a similarity measure in order to compare the volumen fuses from the segmentation methods and the real segmentation. The segmentation of the images consists in a total of 41 labels. Spatial STAPLE employs a sliding window with a size of 10 voxels in the three dimensional space($10 \times 10 \times 10$). The results shows that Spatial STAPLE improves over STAPLE, especially in the gray matter and some smaller labels in the mid-brain. Majority vote is slightly better than Spatial STAPLE. However Spatial STAPLE results improve in special cases over majority vote. The table below shows the average dice coefficient over the 24 target images, with 5 and 15 registered images.

	Spatial STAPLE	STAPLE	Majority Vote
5 registered images	0.916	0.9	0.924
15 registered images	0.931	0.914	0.932

Non local STAPLE

In *Non-Local Statistical Label Fusion for Multi-Atlas Segmentation*, [19], the authors carried out experiments in order to test non local STAPLE approach. They ran the experiments using different sizes of patched neighbourhood; $1 \times 1 \times 1$ and $3 \times 3 \times 3$. The dataset consists of 5 Magnetic Resonance image of the brain, from *OASIS*, for each atlas they have considered a collection of 26 labels, from large structures to smaller volumetric parts of the brain. The experiments were carried out by using two different registration processes; a pairwise non-rigid registration and a pairwise affine registration.

The mean dice similarity coefficient over the whole brain segmentation in 2 subjects using a **Pairwise Affine + Non-Rigid Registration** is represented in the table below:

	NLS(1x1x1)	NLS(3x3x3)
subject 1	0.8979	0.8923
subject 2	0.9138	0.9097

The table below shows the experiment result on another two subjects with a **Pairwise Affine Registration**,

	NLS(1x1x1)	NLS(3x3x3)
subject 3	0.8968	0.9070
subject 4	0.8879	0.9019

Asman and collegues point out that the results show the importance of quality registration and how NLS shows improvement in increasing the patch neighbouring when the registration worsens, specially when the expected voxel correspondence is non-local. [19].

Attribute Similarity and Mutual-Saliency Weighting

Attribute Similarity and Mutual-Saliency Weighting for Registration and Label Fusion, [21], shows the results of the proposed method based on the registration process that is describe in DRAMMS [22], and the fusion label or segmentation process. The segmentation process consists on using a similarity and mutual-saliency weighted voting strategy. The authors used a training data set of 15 subjects from *OASIS* data-base, and divided each subject into 140 different labels. The table shows leave-one-out results:

	Weight mean dice	Direct mean dice
ground-truth ss	0.861	0.756
auto ss	0.843	0.727
no ss	0.839	0.726

The results show that depending whether the ground-truth skull-stripping is used in the registration process or not, the segmentation can differ and that the use of ground-truth skull-stripping helps to improve the segmentation.

3 Image Registration

Image registration is the process of aligning two images of the same environment, by corresponding the same points in the image A and B. The process consists on transforming one image, $A : \Omega \to \Gamma$ where $\Omega \subseteq \mathbb{R}^n$ and $\subseteq \mathbb{R}$ with respect to a reference image $B : \Omega \to \Gamma$ such that some functional $\mathcal{F}(A, B)$ is minimized. $\mathcal{F}$ has the form:

$$\mathcal{F}(A, B, \phi) = \mathcal{M}(A \circ \phi, B) + \lambda S(\phi),$$

where the parameters are the transformation that is applied to the original image, $\widetilde{A} = A \circ \phi$ and $\phi : \Omega \times \mathbb{R}^M \to \Omega$.

$\mathcal{M}$ is a similarity measure, S is a regularization term and λ is a free parameter. The goal of ϕ is to increase the similarity of the image A to that of image B. The registration method can be linear or non-linear.

Consistent image registration consists in to finding the transformation f that transforms A into the correspondence with B, and finding the transformation g that transforms B into the correspondence of A.[1]

$$f = g^{-1}$$

A *symmetric* function consists of:

$$\mathcal{F}(A, B, \phi) = \mathcal{F}(A, B, \phi^{-1})$$

Image similarity and Mutual Information

The most common way to measure similarity is with the form:

$$\mathcal{M}_\Omega = \int_\Omega F(x, A(x), B(x))dx$$

Other ways to measure similarity have the form:

$$\mathcal{M}_\Gamma = \int_\Gamma^2 F(x, a, b, h_{A,B}(a, b))dadb,$$

where $h_{A,B} : \Gamma^2 \to \mathbb{R}_+$ is the joint histogram of image A and B with intensity a and b.

The mutual information of two images, A and B can be defined by:

$$MI(A, B) = H(A) + H(B) - H(A, B)$$

By decreasing the entropy of the joint A and B, the mutual information is maximized. The MI considers the individual entropies of both images.

Normalized mutual information is defined by,

$$NMI(A,B) = \frac{H(A) + H(B)}{H(A,B)}$$

Locally Orderless Registration

LOR is a framework for performing N-dimensional image registration [2]. It provides a unifying way to calculate wide seemingly different similarity measures like Correlation Ratio, mutual information, normalized mutual information, Huber Norm and others. The framework uses the methodology of Locally Orderless Images (LOI) which uses three scale parameters:

- measurement scale: "the effective resolution of the initial image"
- intensity scale: "the effective number of bins in the histogram"
- integration scale: "the effective local spatial extent of local histograms"

According to LOI, a local histogram is obtained as follows: "first a possibly deformed image I is smoothed with the kernel K, a soft isophote i is extracted using kernel P, and finally the isophote mass is calculated in a neighborhood of a point x with kernel W". [2]. Formally:

$$h_I(i, x, \Phi, \alpha, \beta, \varphi) = P(I(x, \Phi, \varphi) - i, \beta) * W(x, \alpha),$$
$$I(x, \Phi, \varphi) = I(x, \Phi) * K(x, \varphi),$$

Where $P : \mathbb{R} \times \mathbb{R}_+ \to [0, 1]$ is an intensity measurement of scale β, it is referred as the Parzen Window. φ is the spatial scale, and α is the integration scale.

$K : \mathbb{R}^N \times \mathbb{R}_+ \to \mathbb{R}_+$ is a spatial measurement kernel of scale, $\varphi, W : \mathbb{R}^N \times \mathbb{R}_+ \to \mathbb{R}_+$ is an integration window of integration scale, α.

Discrete Diffeomorphic Deformations

Discrete Diffeomorphic Deformations, D3 framework, produces diffeomorphic registrations to machine precision. Diffeomorphic transformation preserves the topology of the image by keeping connected the subregions and maintaining the neighbourhood relationships between the structures [1]. By preserving the topology, it allows to combine the information from different sources and maintain a recognized anatomy. D3 framework, uses Normalized Mutual Information, Locally Orderless Registration and Stationary Velocity Fields, it is fully symmetric and consistent. If there is no previous knowledge about the images to register it is recommended to use a symmetric registration model.

D3 framework consists of two symmetric registrations, one `linear affine registration`, and then a `non-rigid registration`. The `linear affine registration` is globally applied to each point in the grid map,

$$x_{lin} = \phi_{lin}(x_{orig}) = Tx_{orig}$$

Where T is a matrix that defines four properties of the global image, translation, rotation, shearing and scaling.

The `non-rigid registration` is determined by a Stationary Velocity Field.

Stationary Velocity Field

Stationary Velocity Field(SVF), v, is a time independent vector field inducing a diffeomorphic mapping ϕ. As Darkner and colleges define in D3[3].

Each mapped point has a homeomorphism function, that is a continuous function between the topological spaces that has a continuous inverse function. The inverse function mapping is needed for it to be a symmetry method. The SVF transformation for a point x, in a time t. It is defined by:

$$x_t = \phi_{nlin}(x_0) = x_0 + \int_0^t v(x_s)ds$$

By negating the velocity of the mapped points gets the inverse function,

$$x_0 = \phi_{nlin}^{-1}(x_t) = x_t + \int_0^t -v(x_s)ds$$

The transformation of an image is defined by:

$$\phi_{final} = \phi_{nlin} \circ \phi_{lin}$$

4 Image Segmentation

Image segmentation is a process in which an image is divided in subregions, each subregion is defined by a specific label. It allows the extraction of meaningful features of the image.

In the case of clinical and medical image analysis, the process of segmentation large data sets can be very expensive and time consuming. In order to solve this problem there are several techniques that automate the process such as Shape-Based Segmentation, Interactive Segmentation, Image-Based segmentation and Atlas-Based segmentation. Atlas-Based segmentation methods will be presented below.

4.1 Expectation-maximization (EM)

EM algorithm consists of two main steps: **expectation step** (E-step) and **maximization step** (M-step). [25]. E-step calculates the probability of the unknown underlying target voxels for all labels, based on the observed training data set and the current estimation. The M-step maximizes the expectation in order to define a label with the highest probability for a voxel target. It then repeats the process until the algorithm converges.

$$\theta = \{\theta_k, k \in K\}$$

The mean and the standard deviation for each Gaussian class define the parameter:

$$\theta_k = (\mu_k, \sigma_k)$$

EM approaches a solution to estimate the underlying parameters. Firstly, it estimates the parameters for the unknown labelled images and then it forms the complete data set and calculates the new θ. Formally EM can be described as an initial $\theta^{(0)}$. Expectation-step(E-step) calculates the conditional expectation:

$$Q(\theta \mid \theta^{(t)}) = \varepsilon[logf(y, z \mid \theta) \mid z, \theta^{(t)}]$$

The M-step maximizes $Q(\theta \mid \theta^{(t)})$ and calculates the new model parameters that is used in the next iteration:

$$\theta^{(t+1)} = argmax_\theta Q(\theta \mid \theta^{(t)})$$

It then repeats the E-step, until the algorithm converges in a local maximum.

In every iteration, the EM algorithm increases the likelihood until the local maximum likelihood is reached. In this iteration the likelihood can not increase or decrease further.

For the method evaluation in this Master's thesis, the algorithm finishes when the change of the likelihood of the next iterations is less than 0.5%.

4.1.1 EM implementation

Expectation-maximization has been implemented as a generic framework in order to expand with the precise probabilistic function that defines the class likelihood of each voxel.

The algorithm 1 is the pseudo-code of the implemented EM.

Algorithm 1 Expectation-Maximization framework

Require: $Target, Atlas$
  $\theta \Leftarrow initialize(Atlas)$
  repeat
    $Q \Leftarrow expectation(\theta, probabilisticFunction)$
    $targetLabeled \Leftarrow maximization(Q)$
    $\theta \Leftarrow calculateParameters(Target, Atlas, Q)$
  until converge
  return $targetLabeled$

Where θ is a matrix with the class Gaussian distributions. Q defines a matrix with all class estimations for the target voxels. The algorithm requires a *target* that is the incomplete labelled MRI brain image and that will estimate the segmentation. It also requires an *atlas* with a complete labelled train data-set composed of an image set image and the expert labelling of them.

The function *maximization*() is straightforward. It defines a label by seeking for the highest class probability in the matrix Q for all voxels in the target image. Once the change of likelihood of the estimations is less than 0.5% in comparison to the last iteration, the algorithm converges and returns the label prediction for the target image. The function *initialize*() initialize the parameters, θ, with the class distributions from the train data.

The function *calculateParameters*() recalculates θ, with the current iteration's expectation, it is used in the next iteration. The pseudo-code 2 defines how to calculate the parameters, θ

Algorithm 2 Calculate Parameters, θ

Require: $Target, Atlas, Q$
  $\theta \Leftarrow [[]]$
  for all Q_i in Q **do**
    $distribution \Leftarrow []$
    $estimation \Leftarrow []$
    $x \Leftarrow 0$
    for all Q_j in Q_i **do**
      $distribution[x] \Leftarrow Target[i]$
      $estimation[x] \Leftarrow Q_{i,j}$
      $x \Leftarrow x + 1$
    end for
    $\mu = mean(distribution, estimation)$
    $\sigma^2 = var(distribution, estimation, \mu)$
    $\theta_{i,j} = (\mu, \sigma^2)$
  end for
  return θ

Where Q is the expectation matrix with the probabilities for each voxel to be in each class, the sub index i denotes a voxel and the sub index j refers to one class. The mean, μ, and the variance, σ^2, define a Gaussian distribution for one specific class. μ is calculated as it is described in 3, the algorithm 4 shows how to calculate σ^2.

Algorithm 3 Mean, μ

Require: $distribution, estimation$
  $\mu \Leftarrow 0$
  for all $intensity, probability$ in $(distribution, estimation)$ **do**
    $\mu \Leftarrow \mu + (intentisy * probability)$
  end for
  $\mu \Leftarrow \mu \div sum(estimation)$
  return μ

Algorithm 4 Variance, σ^2

Require: $distribution, estimation, \mu$
  $\sigma^2 \Leftarrow 0$
  for all $intensity, probability$ in $(distribution, estimation)$ **do**
    $\sigma^2 \Leftarrow \sigma^2 + (probability * (intensity - \mu)^2)$
  end for
  $\sigma^2 \Leftarrow \sigma^2 \div sum(estimation)$
  return σ^2

K-means clustering is a similar algorithm to EM, it is used in image segmentation for clustering the image in different segments or clusters. As EM, it iterates until converge in a optimal solution. The main difference, it is that a voxel can be either in one cluster(class) or not, in EM each voxel has a probability to belong to one possible class. To conclude in EM each voxel has partial belonging to each class, Q matrix defines the probability for each class in the algorithm 1. In K-means, a voxel belongs only to one class.

4.2 Maximum Likelihood Estimator & Markov Random Fields

This section describes the maximum likelihood estimator that it uses as part of the Expectation-Maximization algorithm in order to calculate the probability of one voxel to be in one tissue. Moreover this approach can be extended by using Markov Random Fields as Ledig and collegues proposed in their research [6]. An elementary theory of MRF is presented below to adapt the EM algorithm.

Let an image with n voxels, to be indexed by $i = 1, ..., n$, the image is defined by an intensity $y_i \in \mathbb{R}^m$ for all voxels, $y = y_1, y_2, ..., y_n$. The segmentation of the image is given by $z = z_1, z_2, ..., z_n$, where $z_i = e_k$ with $1 \leq k \leq K$, which voxel z belongs to class k, K is the number of tissue classes. e_k denotes a unit vector having the k^{th} component equals to 1 while all others are equal to 0. The goal is to estimate z by the observed intensities, y, of the image. It is assumed that the intensities of one class k are normally distributed and the voxels are statistically independent.

The probability of observing an intensity y_i at voxel i has the form of:

$$f(y_i \mid \Phi) = \sum_k f(y_i \mid z_i = e_k, \Phi) f(z_i = e_k)$$

Where Φ is the normal distribution of the intensities in one class with mean, μ_k and standard deviation σ_k.

$$\Phi = (\mu_1, \sigma_1), (\mu_2, \sigma_2), ..., (\mu_k, \sigma_k)$$

$f(z_i = e_k)$ is the probability that the voxel i, has to be a class k. $f(y_i \mid z_i = e_k, \Phi)$ is the maximum likelihood estimator(MLE). The global probability of one image with known parameter Φ has the form of:

$$f(y \mid \Phi) = \prod_i f(y_i \mid \Phi)$$

The probability density function is used to calculate the maximum likelihood estimator, it gives the probability of a intensity voxel to be in a class of the registered atlas. It has the form:

$$f(x, \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{1}{2}\Big(\frac{x-\mu}{\sigma}\Big)^2$$

This model can be modified by using **Markov Random Fields**, MRF analyse the similar intensities of the regions in each voxel. The Markov property point out that the probability of one voxel, i to be in one class, k, given the intensities of all other classes is the same as the probability of the neighbour voxels of i to be in the class k.

It is useful for analysing spatial and contextual connection in a physical system, S, such as image labeling. A voxel has relation of neighbourhood with its own closer voxels in the image. Formally is defined by:

$$N = \{N_i \mid \forall i \in S\}$$

Where N_i is the voxels neighbouring i, it has two properties:

- A voxel is not neighbour to it self: $i \notin N_i$
- The relationship in the neighbourhood is mutual: $i \in N_{i'} \iff i' \in N_i$

Let $F = F_1, ..., F_m$ be a set of random voxels on the image, S. F is a Markov random field on S for a neighbourhood set, N iff the two next properties are asserted:

- *Positivity* $P(f) > 0, \forall f \in \mathbb{F}$
- *Markovianity* $P(f_i \mid f_{S-\{i\}}) = P(f_i \mid f_{N_i})$

Where $f_{S-\{i\}}$ is the set of labels in the voxel set $S - \{i\}$ and f_{N_i} is the set of labels of all voxels neighbourhood:

$$f_{N_i} = \{f_{i'} \mid i' \in N_i\}$$

Hammersley-Clifford(1971) theorem has established that MRF is characterized by a Gibss distribution with the form:

$$P(x) = Z^{-1} \exp(-U(x))$$

Where Z is a partition function and $U(x)$ is the energy function that has the form:

$$U(x) = \sum_{c \in C} V_c(x)$$

C is all cycles that define the voxel pair in the neighbourhood. The Gibss distribution is used to define a probability in state with a concrete energy. The energy function measures the energy at a given state which, is built for a specific scenario. In this way it is possible to take into account the influence of the neighbourhood and the proportional

influence of them over the label estimation for the target voxel. The EM approach can be modified by expanding $f(z_i = e_k)$ to:

$$f(z_i = e_k \mid p_{N_i}^{(m)} \Phi_z^{(m)}) = \frac{\pi_{i,k} e^{-U_{M,R,F}(e_k | p_{N_i}^{(m)}, \Phi_z^{(m)})}}{\sum_{j=1}^{K} \pi_{i,k} e^{-U_{M,R,F}(e_j | p_{N_i}^{(m)}, \Phi_z^{(m)})}}$$

Where N_i is the set of first-order neighbours of one voxel.

U_{MRF} is the MRF energy function:

$$U_{MRF}(e_k \mid p_{N_i}^{(m)}, \Phi_z^{(m)}) = \sum_{j=1}^{K} G_{k,j} (\sum_{l \in N_i^x} s_x p_{lj} + \sum_{l \in N_i^y} s_y p_{lj} + \sum_{l \in N_i^z} s_z p_{lj})$$

Where G is a $K \times K$ matrix, that defines the connectivity between class k and j.

$$g(i,j) = \begin{cases} 0 & \text{if } i = j \\ 0.15 & \text{if structures i and j share a boundary} \\ 0.75 & \text{if structures i and j are distant} \end{cases}$$

p_{lj} is the density probability of voxel l to be in the class k. $s = \{\frac{1}{d_x}, \frac{1}{d_y}, \frac{1}{d_z}\}$ defines the Euclidean distance between the target voxel and the neighbourhood.

For K classes and N atlases, the anatomical probabilistic atlas for a voxel i for class k is computed as:

$$\pi_{i,k} = \frac{\sum_{n=1}^{N} \omega_{i,n} \gamma_{i,k}^n}{\sum_{k=1}^{K} \sum_{n=1}^{N} \omega_{i,n} \gamma_{i,k}^n}$$

Where $\omega_{i,n}$ is the inverse of the sum squares differences of one intensity voxel and its first-order neighbours.

$$\omega_{i,n} = (\sum_{j \in N_{i,r}} (y_j^u - A_j^n)^2)^{-1}$$

$$\gamma_{i,k}^n = \begin{cases} 1 & \text{if } z_i = e_k \text{ in } A^n \\ 0 & \text{if else} \end{cases}$$

4.3 Maximum a Posteriori

By employing Maximum a Posteriori (MAP), EM algorithm is modified in order to include a posterior probabilistic, the posterior is calculated with a prior and likelihood probability. This probabilistic prior improves the segmentation by defining a probability based on the intensity or label information of the target voxel and the neighbouring.

Two probabilistic priors have been designed in this project: a local prior and a neighbour prior(semi-local).

A local prior takes into account the most common label for a specific voxel in the atlas, such as the prior which is higher for a label that has more presence in the atlases. For a given set of labelled train images, S, the local prior probability is calculated for each voxel as described below.

$$prior(e_k) = \frac{\sum_{i=0}^{S} t(s_i, e_k)}{|S|}$$

$$t(s_i, e_k) = \begin{cases} 1 & \text{if } s_i = e_k \text{ in } e \\ 0 & \text{if else} \end{cases}$$

The neighbour prior probability is calculated by including k-order neighbours, where $k = 1$ first order neighbours, for the evaluation of this prior. This prior expands the label information to the neighbouring in order to count the most common labels for the nearest voxels to the target. This prior could improve the segmentation in such cases that there is a bad quality registration and the voxels of the atlas are not well aligned.

There are many ways to improve the prior probabilistic to analyse the similarity intensity of the neighbourhood voxels and its respective labels. Thus the prior probability for a tissue is higher when the similarity of the target voxel and it neighbourhood is similar and they are in the same the tissue. This matter is explored in more detail in Non local STAPLE and in Attribute-Similarity-and-Mutual-Saliency-Weight sections.

Bayes' theorem is generally used to amend the probability of one event with extra information, more formally a posterior probability is defined by Bayes' theorem, that is determined by the prior probability and the likelihood estimator.

$$p(e_k \mid y) = \frac{p(y \mid e_k)p(e_k)}{p(y)}$$

Where $p(y \mid e_k)$ is the likelihood probability and $p(e_k)$ is the prior probability. The prior probability can be either global or local. $p(y)$ is considered to be the marginal likelihood.

In fact, the denominator, $p(y)$ can be removed since it is a constant for all classes k, the posterior can be rewritten to:

$$Posterior = Likelihood \times Prior$$

The goal consists on maximizing the posterior probability to suit the best class for a given voxel intensity y_i:

$$f(y_i) = arg_{e_k} max\ p(e_k \mid y_i)$$

Expectation Maximization is then modified to include the MAP estimator and updates the parameters for each iteration with the posterior probabilistic.

MAP can be interpreted as a regularization process that introduces prior knowledge for preventing problems, as over-fitting. In ML, the prediction model can over-react as the voxels have very different intensities. This can happen if the atlas has a poor registration and there are not enough registered atlases. Regularization can prevent over-fitting.

4.4 Non local Spatial STAPLE

Non-Local Statistical Label Fusion for Multi-Atlas Segmentation is described in this section. Firstly a theory framework of *Simultaneous Truth and Performance Level Estimation (STAPLE)*[18] family algorithms is presented, then NLS approach is illustrated.

4.4.1 STAPLE

Let considerer N be a set of voxels from one image, R a collection of raters(registered atlases), and the set, L represents the possible values(or labels) that a rater can assign to one specific n voxel. D is an $N \times R$ matrix that represents the label decisions. T, is a vector that represents the hidden true segmentation of the image, with the same size as N. Where $T \in 0, 1, ..., L-1$. The objective of the STAPLE method is to estimate the true segmentation by using the collection of raters,R, and the current estimate of the rate performance level parameters.

By maximizing the complete data log likelihood function, the estimated performance level parameters are selected.

$$\hat{\Theta} = \arg_{\Theta} \max \ln f(D, T \mid \Theta)$$

It is assumed that the label decision is independent from one image segmentation to other one. For each, Θ_{jm}, in Θ denotes a confusion matrix LXL, it is referred for a specific rate j and in a region B_m. Θ_{jm} is defined over the voxels in the region B_m.

W is a matrix, $L \times N$. For each element in the matrix, $W_{s,i}$, indicates the probability that correct segmentation for the voxel i, is the label s. W in a iteration k has the form of:

$$W_{s,i}^{(k)} \equiv f(T_i = s \mid D_i, \Theta_{j,m}^{(k-1)})$$

$$= \frac{f(T_i = s) \prod_j \Theta_{j,m,n,s}^{k-1}}{\sum_{s'} f(T_i = s') \prod_j \Theta_{j,m,n,s'}^{k-1}}$$

m is choosen according $i \in B_m$ and $\Theta_{jmns}^{(k-1)}$ denotes the probability of the rater j to belong to the label n for a given true label, s, in the region B_m. The prior probability, $f(T_i = s)$ is a global prior. It estimates the amount of voxels that belong to the label s in the true segmentation.

4.4.2 NLS method

As EM is used in other approaches to estimate the true segmentation. As explained by Asman and collegues [19], NLS is also used to calculate the distance variance and intensity variance for each voxel in the patch neighbourhood. This approach provides a technique to determine a correspondence model between the target voxel and the atlas voxel. There are two primary components:

- **Intensity similarity** between a given atlas voxel and the target of interest by using a Gaussian intensity difference in a normalize intensity set.
- **Spatial compatibility** between two voxel locations in the common target image coordinate system. The search patch is limited in order to prevent disorientation between classifications with similar intensities.

The probability of correspondence between an atlas voxel and the given target voxel is the product of two Gaussian distributions: Gaussian intensity difference and Gaussian window-based.

$$f(A_{i'j} \mid I_i) \equiv \alpha_{ji'i} = \frac{1}{Z_\alpha} \exp\left(- \frac{(\varphi(A_{i'j}) - \varphi(I_i))^2}{2\sigma_i^2} \right) \exp\left(- \frac{\varepsilon_{ii'}^2}{2\sigma_d^2} \right)$$

φ is a set of intensities in the patch neighbourhood, for the target voxel and the atlas. σ_i is the standard deviation of such distribution. In the spatial compatibility, $\varepsilon_{ii'}$ is the Euclidean distance between the voxel i and i', and the σ_d^2 is the standard deviation.

Z_α is partition function that enforces:

$$\sum_{i' \in N(i)} \alpha_{ji'i} = 1$$

With this constraint, $\alpha_{ji'i}$ can be interpreted as a probability that the voxel i' in the atlas j is correspondence with the voxel i.

The underlying performance level parameters are defined by:

$$f(D_{i^*j} = s', A_j \mid T_i = s, I_i, \Theta_{js's}) = f(D_{i^*j} = s' \mid T_i = s, I_i, \Theta_{js's}) \equiv \Theta_{js's}$$

By using the non local correspondence model is possible to estimate the corresponding voxel i^*,

$$f(D_{i^*j} = s', A_j \mid T_i = s, I_i, \Theta_{js's}) = \sum_{i' \in N(i)} \Theta_{js's}\alpha_{ji'i}$$

Where $N(i)$ is the patch neighbourhood and $\alpha_{ji'i}$ is the previous defined non local correspondence model.

4.4.3 MapReduce

The complexity of NLS is high due to the amount of calculations to perform in order to estimate a tissue for each voxel. The time complexity for each iteration in EM algorithm with NLS as probabilistic estimator is:

$$O(n * a * k * p)$$

Where n is the total amount of voxels in the image, k is the number of classes, p is the size of the neighbourhood patch and a is the number of atlas used as train data.

To address this issue, the computation of NLS has been parallelized with an implementation of the programming model MapReduce [20]. MapReduce is a technique which is commonly used for computing large dataset in parallel, it is composed for two procedures, *map* and *reduce*.

Firstly the *map* procedure divides the total amount of voxels in equal parts, all parts are uniformly distributed over a set of workers. The worker sequentially performs the NLS algorithm over the voxels which have been assigned to it. Once the worker has finished the computation, it sends the label estimation of the voxels to the *reduce* procedure. The *reduce* procedure retrieves the results from all workers and merges all label estimations in order to build the segmentation of the complete brain. The *map* procedure creates a set of inputs `key/values` pairs for each voxel estimation, where the key that identifies the voxel and the values are the needed parameters to compute NSL for the specific voxel. It includes the voxel intentity and patch information across all train atlas. The output of *reduce* procedure is a `key/value`, the key identifies the voxel and the value the estimated label for the voxel. Each worker creates a system process and the process

is spread over the available cores in the system. All this, is handled by the operative system.

This proposed solution optimizes the running time of the algorithm in multi-core architectures by spreading the computation over the available cores.

4.5 Attribute Similarity and Mutual-Saliency Weighting

Attribute Similarity and Mutual-Saliency Weighting is presented by Ou and colleagues [21]. This section will describe the concept of similarity of two voxels, mutually-salient, and it establishes a probability for label segmentation of a target voxels.

The similarity of two voxels, i and j is defined by the attribute similarity. It is said that two voxels are similar if the difference of their attibuttes, $A(\cdot)$, is small.

$$sim(i,j) = \frac{1}{1 + \frac{1}{d}\|A(i) - A(j)\|^2}$$

Two voxel i and j are mutually-salient if both are similar and they are less similar to the rest of voxels in the patch neighbourhood.

"Mutually-salient is calculated by dividing the mean attribute similarity of the voxel i over all core neighbour voxels of j, and the mean attribute similarity of the voxel over all peripheral voxels." As Ou and colleagues define in [21].

$$ms(i,j) = \frac{\frac{1}{|CN(j)|}\sum_{w \in CN(j)} sim(i,w)}{\frac{1}{|PN(j)|}\sum_{w \in PN(j)} sim(i,w)}$$

Where $A(\cdot)$ is the multi orientation Gabor attributes [22], CN is the core neighbourhood and PN is the peripheral neighbourhood.

The probability that one voxel, u has the label, l is defined by:

$$Pr(label(u) = l) = \frac{\sum_n sim(T_n^{-1}(u), u) \cdot ms(T_n^{-1}(u), u) \cdot 1(label(T_n^{-1}(u)) = l)}{\sum_n sim(T_n^{-1}(u), u) \cdot ms(T_n^{-1}(u), u)}$$

Where a target voxel, u will have n atlas label registered images denoted by $label(T_n^{-1}(u))_{n=1}^N$

The goal is to maximize the previous probability over all labels L such the label for a target voxel,u is,

$$l^* = argmax_l Pr(label(u) = l)$$

Each voxel is characterized by a Gabor attribute vector, $A(u)$, which is calculated by resolving the 3D image in two groups of 2D orthogonal planes: x-y and y-z. These are the two Gabor filters:

- Plane x-y

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\Big[-\frac{1}{2}\Big(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\Big)+j2\pi f_x\Big]$$

- Plane y-z

$$h(y,z) = \frac{1}{2\pi\sigma_y\sigma_z}\exp\Big[-\frac{1}{2}\Big(\frac{y^2}{\sigma_y^2}+\frac{z^2}{\sigma_z^2}\Big)+j2\pi f_y\Big]$$

Where $\sigma_x, \sigma_y, \sigma_z$ are the standard deviations of the Gaussian envelope in the spatial domain. $f_x and f_y$ are modulating factors in the frequency domain.

$A(u)$ is defined by:

$$[|Real[(I_i{*}g_{m,n})(x,y)]|,|Imaginary[(I_i{*}g_{m,n})(x,y)]|,$$
$$|Real[(I_i{*}h_{m,n})(y,z)]|,|Imaginary[(I_i{*}h_{m,n})(y,z)]|]_{m=1,2,..M;n=1,2,..N}$$

4.6 Support Vectors Machine

Support Vectors Machine(SVM) is a supervised method for data classification. By using a trained data set it builds a model that is able to classify unknown data into two labels, SVM is a binary classification of negative and positive class.

SVM builds a hyperplane in an infinite-dimensional space that classifies the train data into two different sets. It is not expected for the trained data to be linearly separated $(\vec{x_1}, y_1), (\vec{x_2}, y_2), ..., (\vec{x_n}, y_n))$, that it is why is transformed in a sufficient high-dimensional space such,

$$(\Phi\vec{x_1}, y_1), (\Phi\vec{x_2}, y_2), ..., (\Phi\vec{x_n}, y_n))$$

In the case where it is possible to separate it by a hyperplane, it is also possible to have multiples hyperplanes, as presented in Figure 1.

In the image all of the hyperplanes separate equally well, but a maximization process should be performed in order to find out which hyperplane will approach better with unknown classified data.

The goal is to find a hyperplane that maximizes the margin with the closest negative and positive class. These margins are called the support vector machines. If the margin is large the separation of the training examples is robust to support small changes in the hyperplane. Consequently the classier error is lower. For a given function kernel that classifies the training data in two parts, it is possible to measure the distance from the training data to the perimeter of the classifier. So, d_+ defines the smallest distance for the positive class(+1) and d_- is the distance to the closest negative class(-1) data point. The margin of the hyperplane is defined by the sum of both distances: $d_+ + d_-$.

The equation of the hyperplane has the form of:

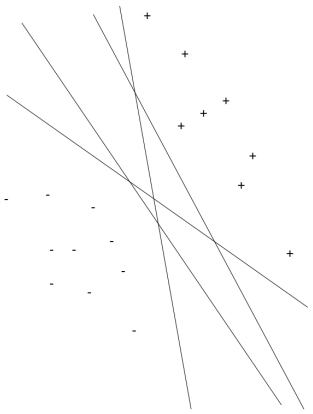

Figure 1: Multiple hyperplanes that fit well the division of the negative and positive class in the given space.

$$\overline{w} \cdot \overline{z} + b = 0$$

Where $\overline{w}$ is a vector and the scalar b, represents the hyperplane inside the transformed space. For any point $\overline{z}$ that satisfies the equation then is in the hyperplane. In the case that for a given point $\overline{z}$ that satisfies $\overline{w} \cdot \overline{z} + b > 0$, that point lies in one part of the hyperplane. And in the case that $\overline{w} \cdot \overline{z} + b < 0$, the point lies in the other part of the hyperplane.

In order to classify new data the function that defines whether the data points belong to the positive or negative class has the form:

$$f(\overline{x}) = \overline{w} \cdot \Phi(\overline{x}) + b$$

Where Φ is the conversion to the transformed space.

The maximization of the margin hyperplane is given by calculating $\overline{w}$ and b,

$$\overline{w} = \sum_{i=1}^{n} \alpha_i \gamma_i \Phi(\overline{x_i})$$

SVM is a two-class classifier and, there are several methods to combine multiple two-class SVMs into a multi-class classifier. One proposed solution by Vapnik(1998) is to

build a K different SVMs, where in each k^th the model is trained using the class C_{k^th} as the positive and the rest classes as the negative class. This method is called, "one-versus-the-rest" approach. However this approach can point to wrong classifications. For example, one data point can be assigned to multiple classes. Another approach proposed by, Weston and Watkins(1999), is a method that trains simultaneously all SVMs and maximizes the margin for each class to the rest classes. The complexity of this method is high, $O(K^2N^2)$, where K is the number of classes and N is the number of data points in the trained data set.

One-against-one is proposed by Knerr et al.(1990), it performs all binary combinations in k classes with a total, $k(k-1)/2$ classifiers. Each classifier is applied to the test data. After giving one vote to the *winning* class, the data is classified with the major voted class. This means that the class that is the most classified with all binary combinations, is the one that classifies the point. In the supposed case that there is a tie between two or more classes, a tie-break strategy is applied. The strategy consists on randomly selecting a class from the tied class set. The main disadvantage of this method is the large number of classifiers that are created when the number of classes is large. This reduces the algorithm's performance.

5 Experiments

This section describes the performed experiments of the implemented methods. A short introduction outlining the system environment, the used data sets and the methodology are presented.

5.1 Introduction

All the experiments outlined below have been executed in the same environment, `Ubuntu 14.04(x86-64)` with a CPU `Intel Core I7-4710HQ`, `4 cores at 2.50 GHZ`. The implementation of the methods and the experiments have been done in `Python 2.7.6`. Two common libraries have been included such as *Numpy* which has facilitated operations with multi-dimensional arrays, and *Matplotlib pyplot* which was employed for visualizing data in plots and charts.

All the methods were evaluated by following a *test one leave out* in two data sets:

- *Miccai* with 15 MRI brain images, which has been registered with a non-linear method. The images are labelled into 134 regions. The MRI brain images are originated from the OASIS project [2].
- *MGH*, a data set which consists of 10 images with a resolution $182 \times 218 \times 182$, and manually labelled into 106 regions. All brain images have been registered by a non-linear registration approach.

Test one leave out consists on a total n (amount of brain images) permutations, in which an image is used as test image and the rest as training data set. The goal is to estimate the one that is used as a test, then to calculate the similitude score with the real labelling.

5.1.1 Image Registration

In this section, the process for generating a registration template is described for both datasets; *Miccai* and *MGH* . In the report `Benchmarking of template strategies` [4], the authors describe two non-linear registration methods called *All to All* and *mean*. Based on the experiments that are done in the report, the method *All to All* shows a better coefficient overlap. This non-linear registration method was used on both datasets for this Master's thesis.

The *All to All* method is an iterative method in which each image is firstly linearly transformed. There are two sets of images; the linear transformed images, I, and the linear and non linear image transformations, J. At the beginning, J and I are the same group of images. The steps are outlined below:

[2] http://www.oasis-brains.org/

1º Register each image from the group, I to all images in the group J (not including the transformed image).

2º Update J with the current transformation of each image in I.

3º Repeat and for the next iteration, use the obtained transformations as initial value in the optimization.

Firstly the linearly aligned original images are registered to the other original images and the new target images are found. In the second step, each original image is registered to the new targets for the other images and the new target transformation is found. Like the second step, in the last step, each original image is registered to new found target of the other images. This process is repeated with decreasing warp scales.

Overlap is calculated in order to evaluate the generated templates, it is calculated between the labelling of two images;

$$overlap(I, J, i) = \frac{\sum_{\Omega}(label(I, i) \wedge label(J, i))}{\sum_{\Omega}(label(I, i) \vee label(J, i)}$$

The mean overlap is the average of all label overlaps in the image. The mode mean overlap is calculated between an image and the template brain in the template space. The brain template labels are calculated by a majority voting strategy where the most counted label is the one selected for each voxel/point in the image.

The experiment has evaluated the overlap when template is generated by using the mean method and *All to All* method in order to label a unknown label image. The configuration parameters of the experiment are; *All to All* method runs for five iterations at warp scale 40 and 20 mm and 9 iterations at 10mm, λ_2 was set to 0.001. The mean method runs over 5 iterations at warp scape 40 mm, and one iteration at 20 and 10 mm, λ_2 was set to 0.005. This experiment uses the whole dataset in order to build the template. The overlap is calculated by using nine images over ten images and removing one. This process is done ten times and there are in total ten permutations which means that for each permutation a different image is removed.

Figure 2 shows the mode overlap between nine images and the remaining one, for the ten permutations. All brain images from the dataset have been used to build the template.

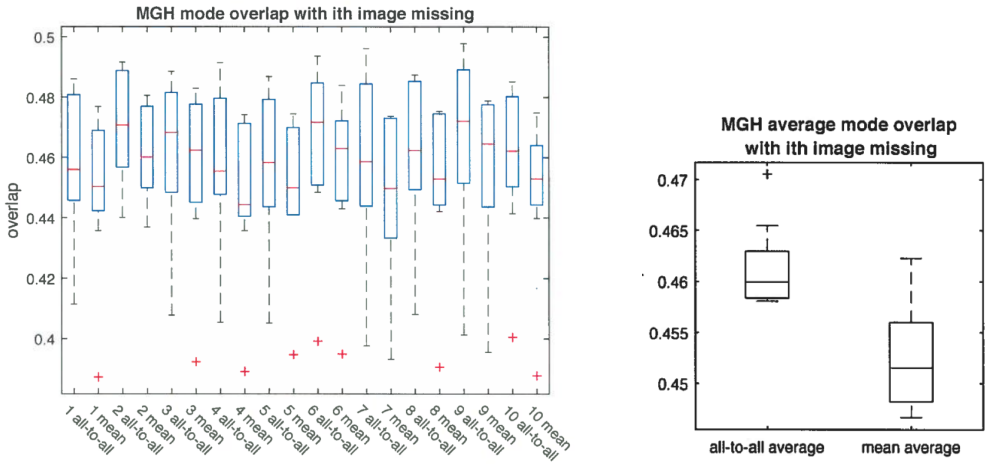

Figure 2: The mode mean overlap shows the bow plot for each method in all possible permutations, the average mode mean overlap shows the the average mode mean overlap of all permutation for the two evaluated methods.

5.1.2 Dice score coefficient

The dice coefficient was used as a similarity measure to evaluate the methods. It compares the label overlap of two images, the real label image and the predicted image generated by the selected methods. The dice coefficient, QS is calculated for each label in the images X and Y.

$$QS_k = \frac{2|X \cap Y|}{|X| + |Y|}$$

Where $|X|$ is amount of voxels in the image X that has a label k and $|Y|$ is the amount of voxels in the image Y that has a label k. The union, $|X \cap Y|$, calculates the amount of voxels that have the same label, k, in both images.

In all the experiments, the dice score coefficient is calculated, **direct mean** and the **weighted mean** are presented. **Direct mean** shows the mean over all label dice score coefficients and the **weighted mean** shows weighted mean based on the presence of the label in the segmentation. So for a label that appears more frequently, it has more weight when the mean is calculated. Both measures are calculated as in some cases the presence of the labels is low and the prediction is poor. Then, the direct mean dice coefficient decreases drastically. Having both measures gives a better view of the evaluated methods for the readers.

5.2 Expectation-maximization ML and MAP

Maximum likelihood estimator and maximum a posteriori with a local prior probability and semi-local prior by using the expectation maximization framework are evaluated in this section using Miccai and MGH data.

Miccai

The table 1 compares the three evaluated methods. Both maximum a posteriori approaches perform better than maximum likelihood. This is expected as the prior probabilistic improves the estimation by including more information. Nevertheless, both evaluated priors have similar estimates, since they have similar results.

	Weighted mean dice coefficient	Direct mean dice coefficient
ML	**0.768**	**0.622**
MAP	**0.801**	**0.680**
MAP-1n	**0.801**	**0.681**

Table 1: Dice coefficient of ML, MAP with a local priori and MAP with 1n priori. Miccai Data.

Figure 3 presents the weighted mean dice coefficient for each evaluated image. The plot shows that maximum likelihood approach has an inferior estimation than maximum a posterior approaches, as the table 1 indicates. Moreover, there is an under estimation in the permutations 2, 5 and 12 for the three evaluated approaches.

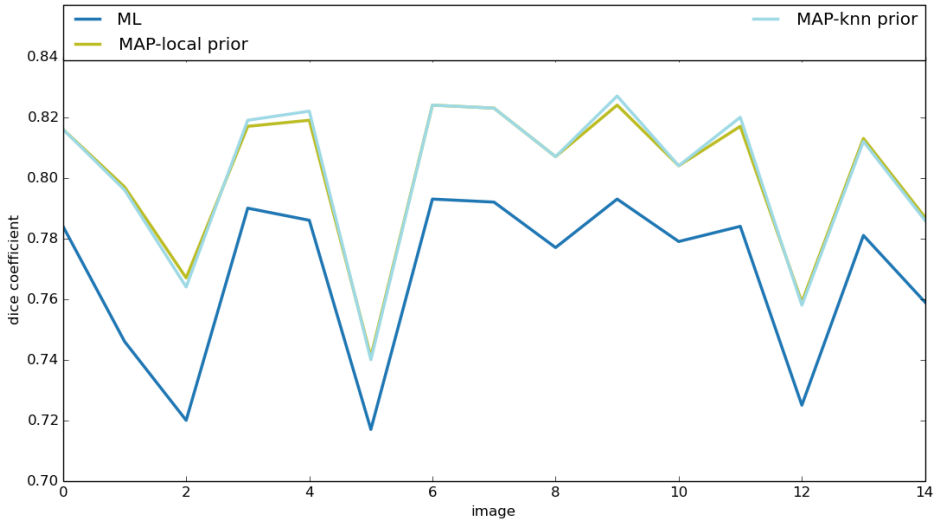

Figure 3: Dice score for each evaluated image in each permutation. The plot shows the weighted mean dice for all labels in the image. Miccai data.

Figure 4 shows the diagram plots of the three different approaches, the `x axis` represents the labels order from max frequency to min appearances frequency, and the `y axis` represent the direct dice coefficient of each label. The dice score seems to be high when there is a high presence of the label, while the presence of the label is lower, then the dice score decreases. In the 6 last labels, which have the lowest presence in the segmented image, the dice score decreases drastically.

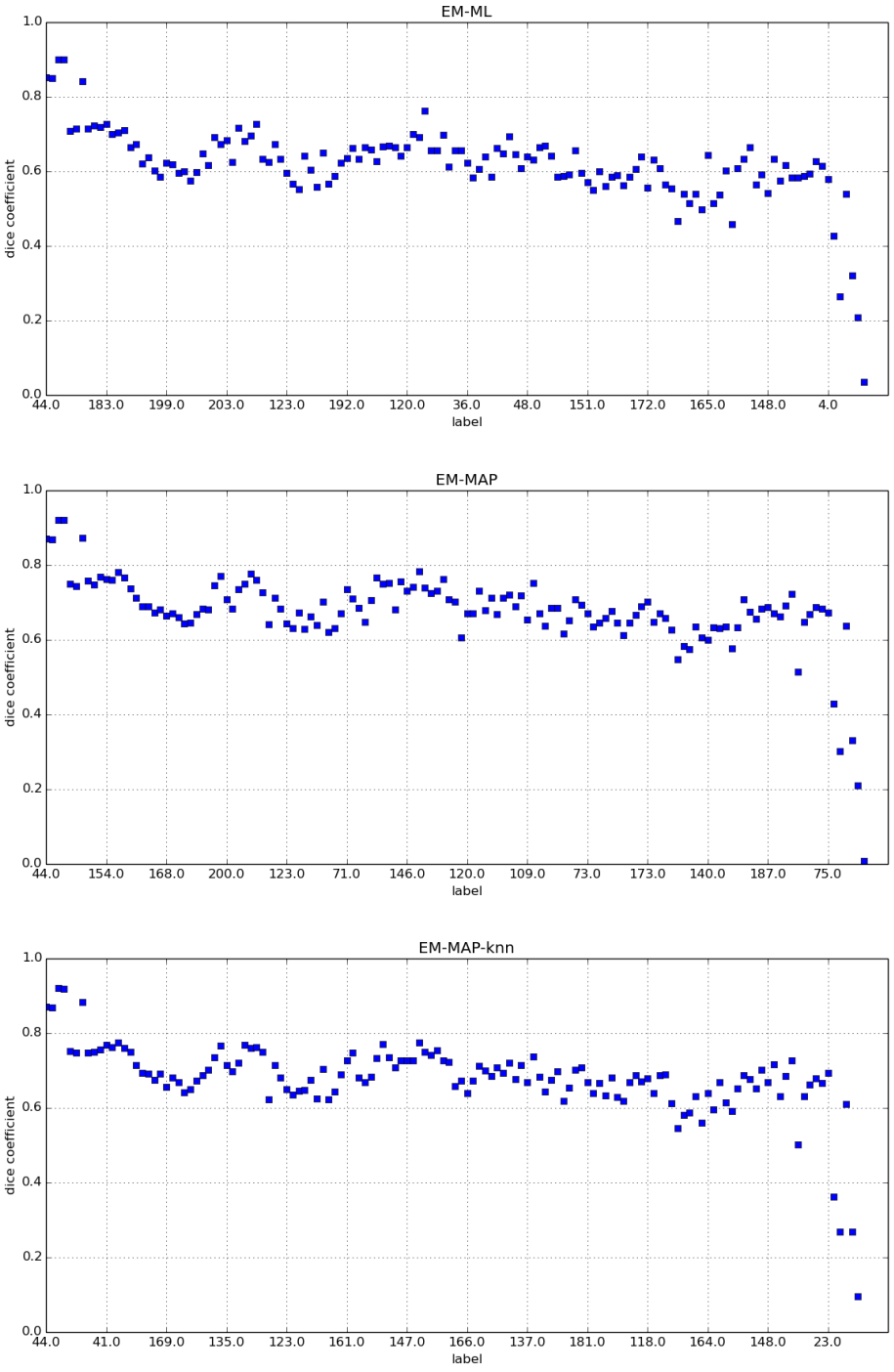

Figure 4: Expectation maximization ML and MAP approaches with Miccai Data. The labels are order from highest to lowest presence

Figure 5 represents the quality accuracy of the three methods in contrast with the real labelling, each image shows the same slice from the front part of the same brain. ML approach seems to make a noisy segmentation, in the image there is discontinuity in the regions. On the other hand, MAP approaches make a more smooth segmentation in the whole image.

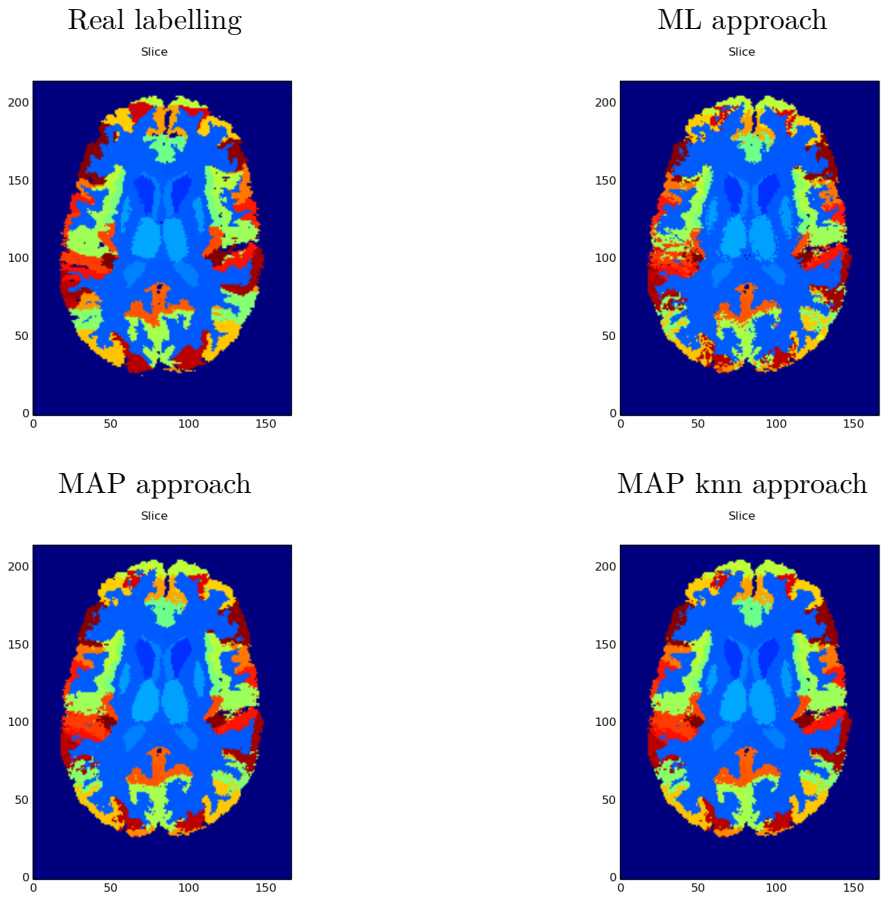

Figure 5: Four slices that represents the quality labelling of the three approaches with Miccai Data

MGH

Table 2 compares the three methods with the weighted and direct mean over all 10 images from the MGH dataset. As expected, MAP approaches have better segmentation than ML approach due to the prior probability that MAP includes.

	Weighted mean dice coefficient	Direct mean dice coefficient
ML	**0.718**	**0.560**
MAP	**0.744**	**0.596**
MAP-knn	**0.748**	**0.602**

Table 2: Dice coefficient of ML, MAP with a local priori and MAP with knn priori. MGH Data.

Figure 6 shows the weighted mean dice coefficient for each evaluated image with MGH data. In the permutation 1, 4 and 7, the dice coefficient is below to the presented weighted mean dice coefficient 2. This circumstance is repeated for ML and both MAPs.

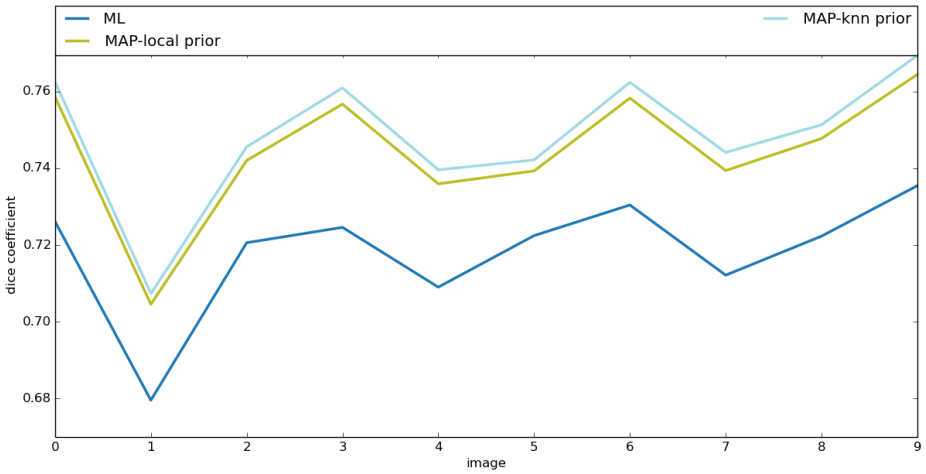

Figure 6: Dice score for each evaluated image in each permutation. The plot shows weighted mean dice of the labels in the evaluated image. MGH data.

Figure 7 shows the dice coefficient over all estimated labels in the three approaches. The three plots seems to be similar. The labels which have highest presence have a better score, while the ones that have less presence in the segmentation have a lower score. The three labels which have lowest presence have a score under 0.4 in the three methods.

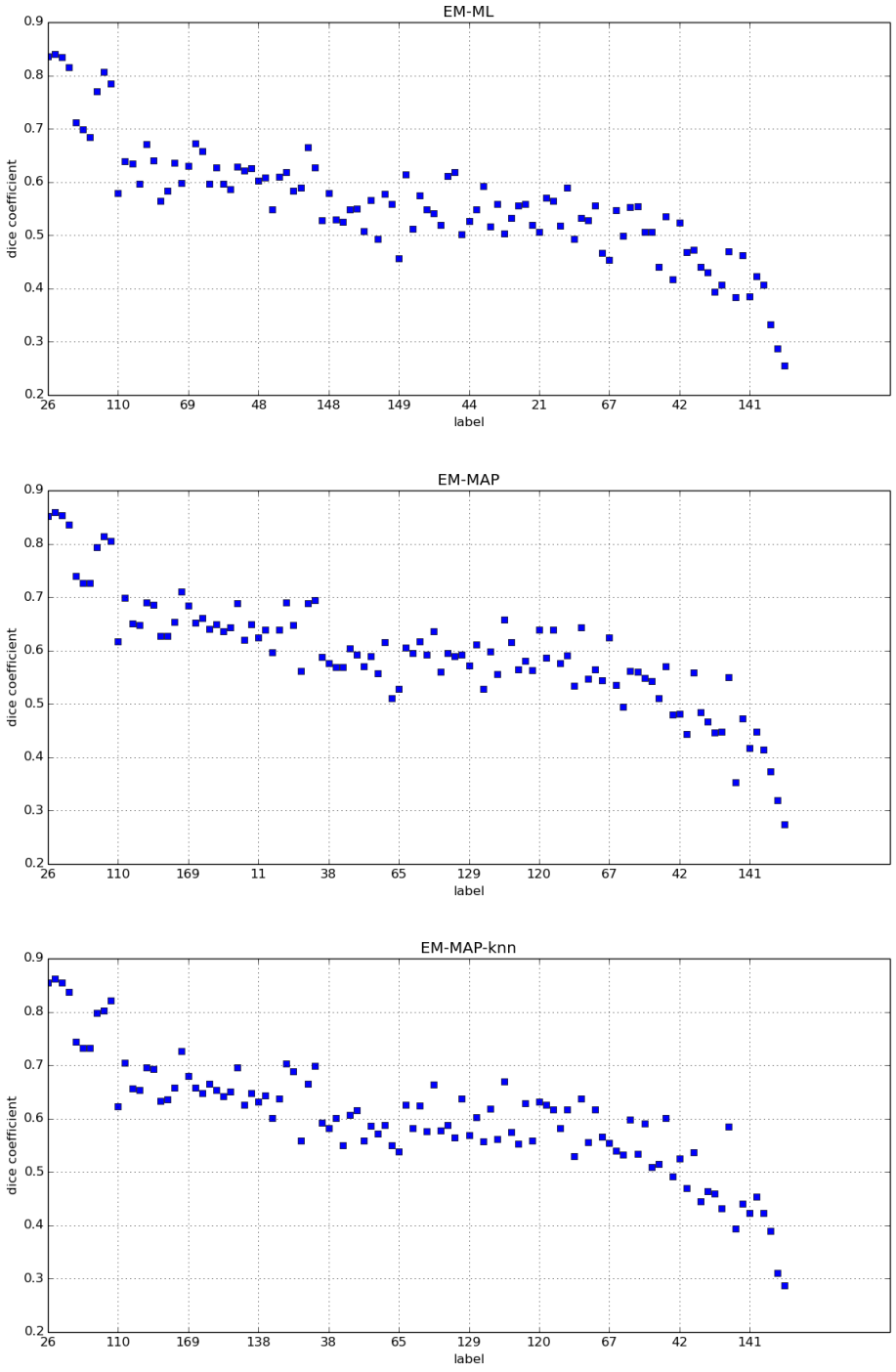

Figure 7: Expectation maximization ML and MAP approaches with MGH data. The labels are order from highest to lowest presence

Figure 8, the images represent the quality label estimation of one slice from the front part of the same brain. As the images show, MAPs approaches have a more smooth segmentation than ML.

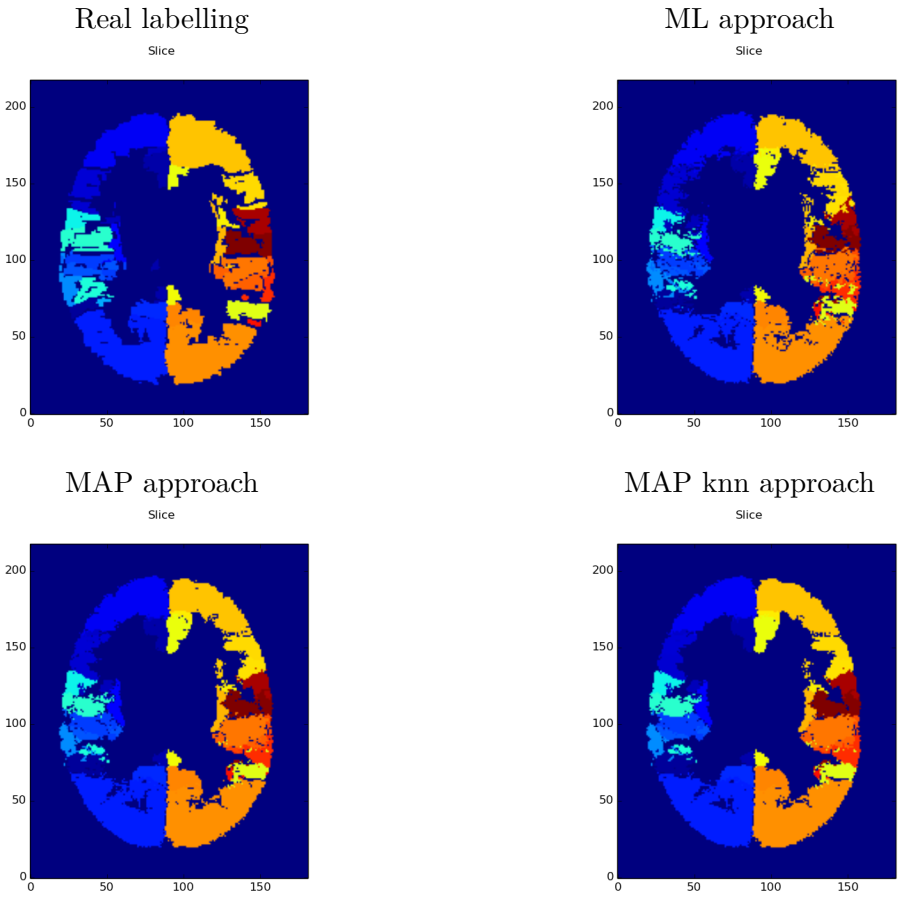

Figure 8: Four slices that represents the quality labelling of the three approaches with the MGH data

5.3 Expectation-maximization MRF

Markov Random Field (MRF) has been evaluated using the EM framework. It has been tested with Miccai and MGH dataset by using a *test one leave out* approach.

Moreover the anatomical probabilistic atlas has also been evaluated, it consists on expanding $f(z_i = e_k)$ 4.2 to:

$$f(z_i = e_k) = \pi_{i,k}$$

It analyses the relation of the target voxel intensity and the intensities of the first order

neighbourhood over all atlas with the sum of squares difference. It gives higher probability to the voxels that have the same label in the atlas and expected target voxel label. Anatomical probabilistic atlas has been introduced in the EM algorithm in order to maximizes the label expectation of each voxel.

Miccai

Table 3 shows the average weighted and direct mean dice coefficient of the two evaluated approaches in the 15 permutations. Anatomical probabilistic atlas turns out to be a better estimator than MRF for the Miccai data set.

	Weighted mean dice coefficient	Direct mean dice coefficient
MRF	**0.766**	**0.618**
Anatomical prob. atlas	**0.796**	**0.672**

Table 3: Weighted and direct mean dice coefficient of MRF approach and anatomical probabilistic atlas. Miccai Data.

Figure 9 shows the weighted dice mean coefficient for each permutation. For the evaluated methods, MRF and anatomical probabilistic atlas, there are five permutations that have a score below the average, these permutations are 1, 2, 5, 12 and 14.

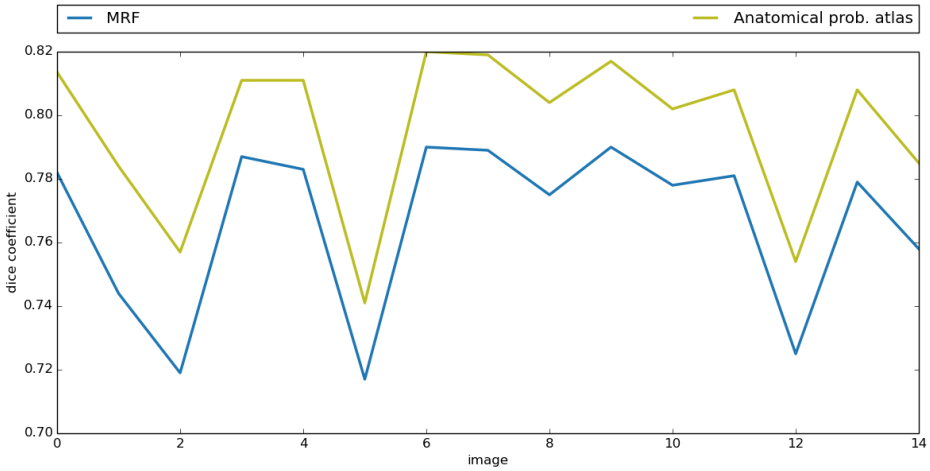

Figure 9: Weighted dice score for each evaluated image in each permutation. Miccai data.

Figure 10 shows three label slices of the same brain, one is the real labelling and the rest are the label estimation of MRF and anatomical probabilistic atlas. MRF is not as

polished as the anatomical probabilistic atlas which is comparable to the ground truth in the real labelling slice.

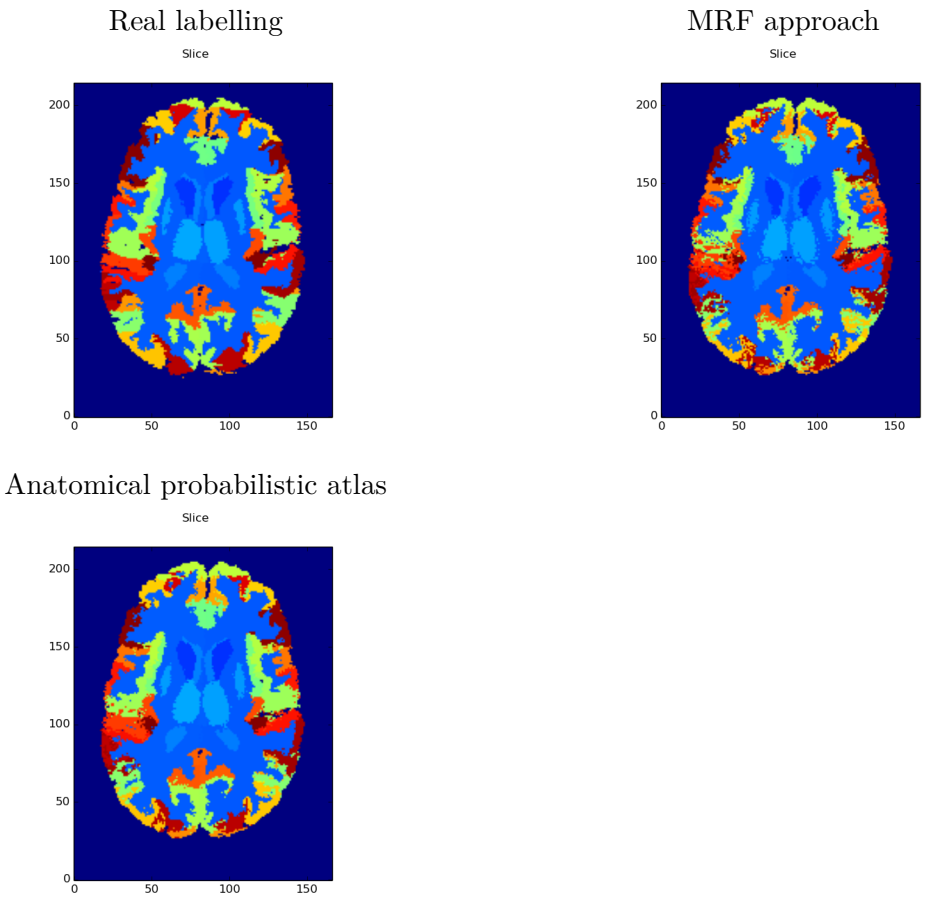

Figure 10: Three slices of the same brain that represents the quality labelling of the evaluated approaches. Miccai data

Figure 11 shows the average dice coefficient of each label for all permutations in both evaluated methods. Both plots are similar, for the 40 labels which have higher presence, the dice score is superior to the rest. Nevertheless there are many outliers, that define a worst segmentation on those labels.

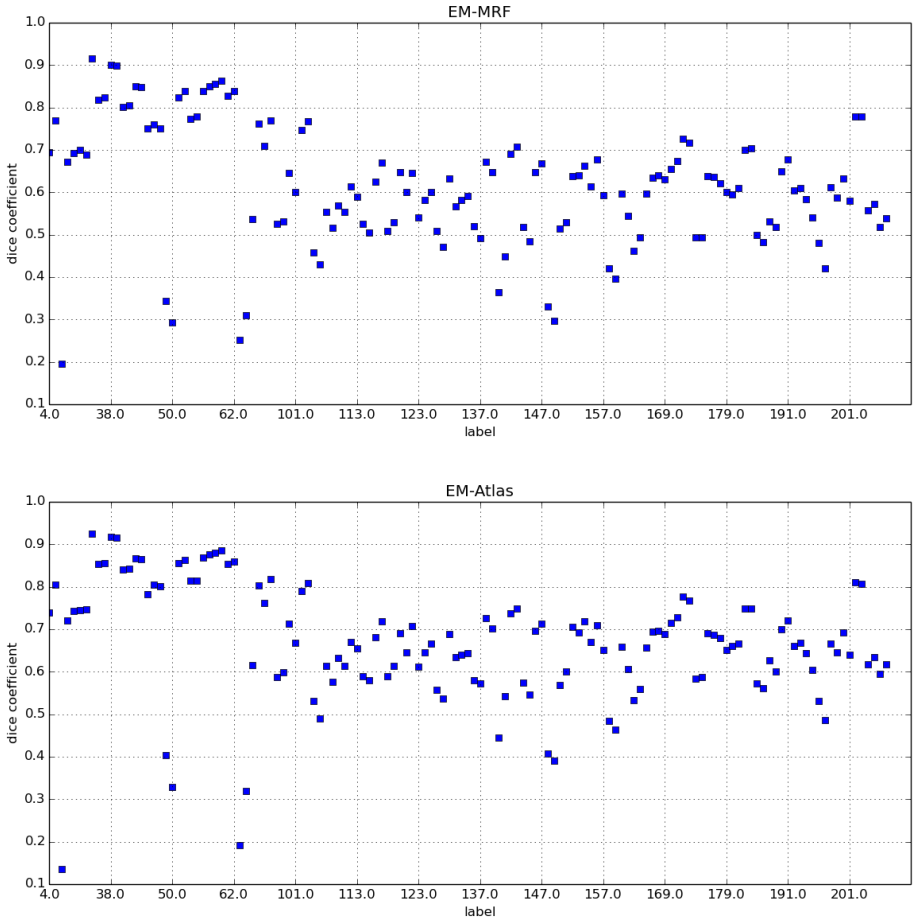

Figure 11: Dice score of MRF classification and anatomical probabilistic atlas classification for all labels. The labels are organized from the highest to lowest presence. Miccai data

MGH

Table 4 shows the average weighted and direct mean dice coefficient of the two evaluated approaches in the 10 permutations. Anatomical probabilistic atlas seems to be a better probabilistic than MRF.

	Weighted mean dice coefficient	Direct mean dice coefficient
MRF	**0.715**	**0.556**
Anatomical prob. atlas	**0.739**	**0.589**

Table 4: Weighted and direct mean dice coefficient of MRF approach and anatomical probabilistic atlas. MGH Data.

Figure 12 shows the weighted mean dice score for each image in each permutation. Anatomical probabilistic atlas has a better score than MRF for all the permutations. Nevertheless in the permutation 1, 4 and 7 that correspond to the images 2, 5 and 8, the score is less than the average for all calculated permutations for MRF and Anatomical probabilistic atlas approach.

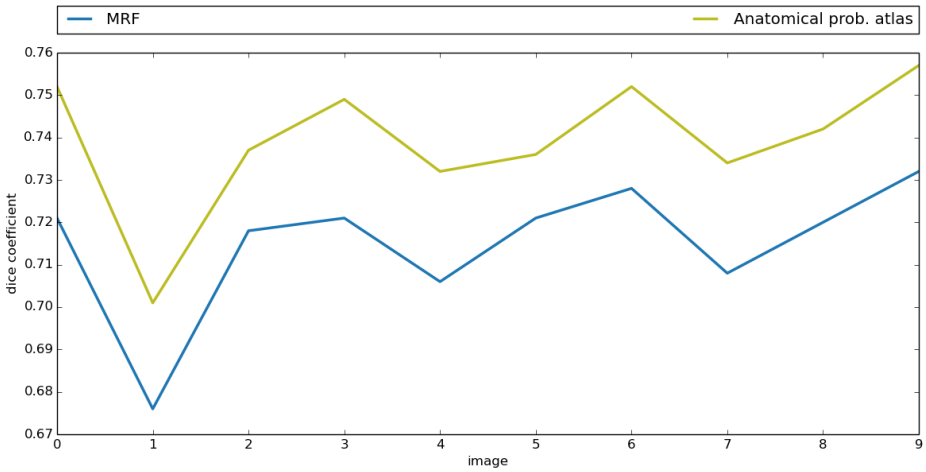

Figure 12: Weighted dice score for each evaluated image in each permutation. MGH data.

Figure 13 shows three label slices of the same brain, one shows the real segmentation which is the ground truth. The other two show the label estimation of the evaluated methods; MRF and anatomical probabilistic atlas. The method that uses the anatomical probabilistic atlas have segmentation with more quality than MRF approach. MRF have discontinuity over the label regions, it is not consistent over the image slice.

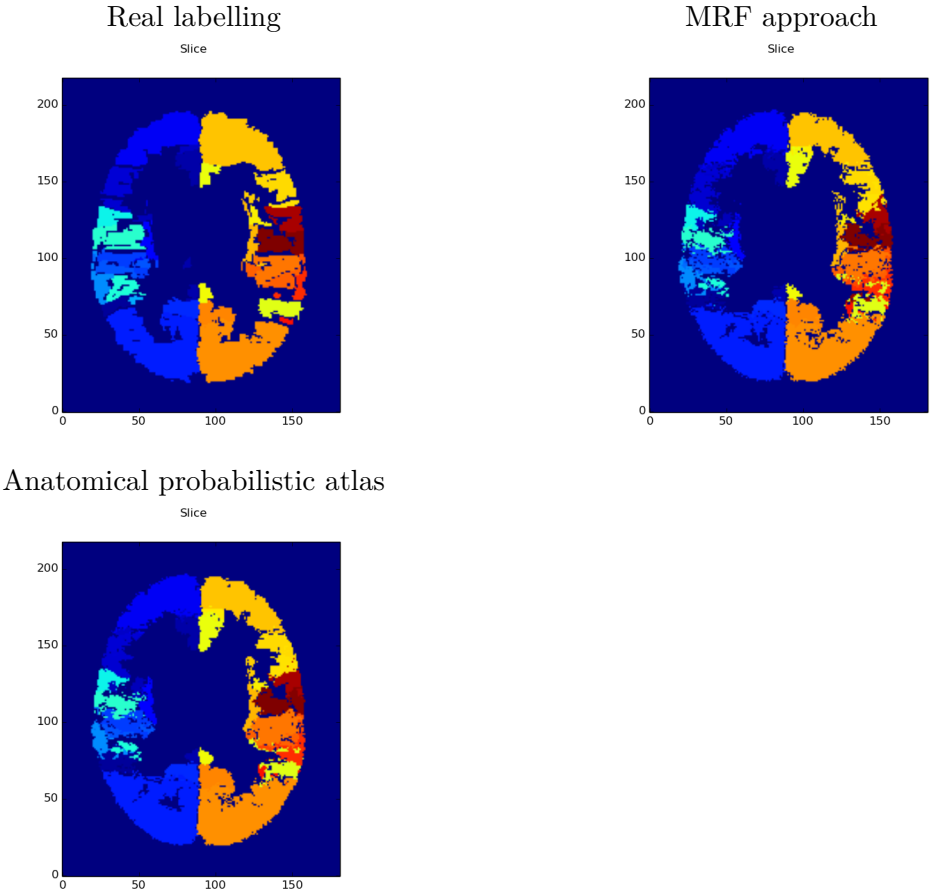

Figure 13: Three slices of the same brain that represents the quality labelling of the evaluated approaches. MGH Data

Figure 14 shows the average dice coefficient of each label for all permutations in both evaluated methods. Both plots are scattered, there is a dispersion of the scores for all labels even if they have highest or lowest presence in the segmentation. The highest or lowest presence of the label does not influence in the segmentation.

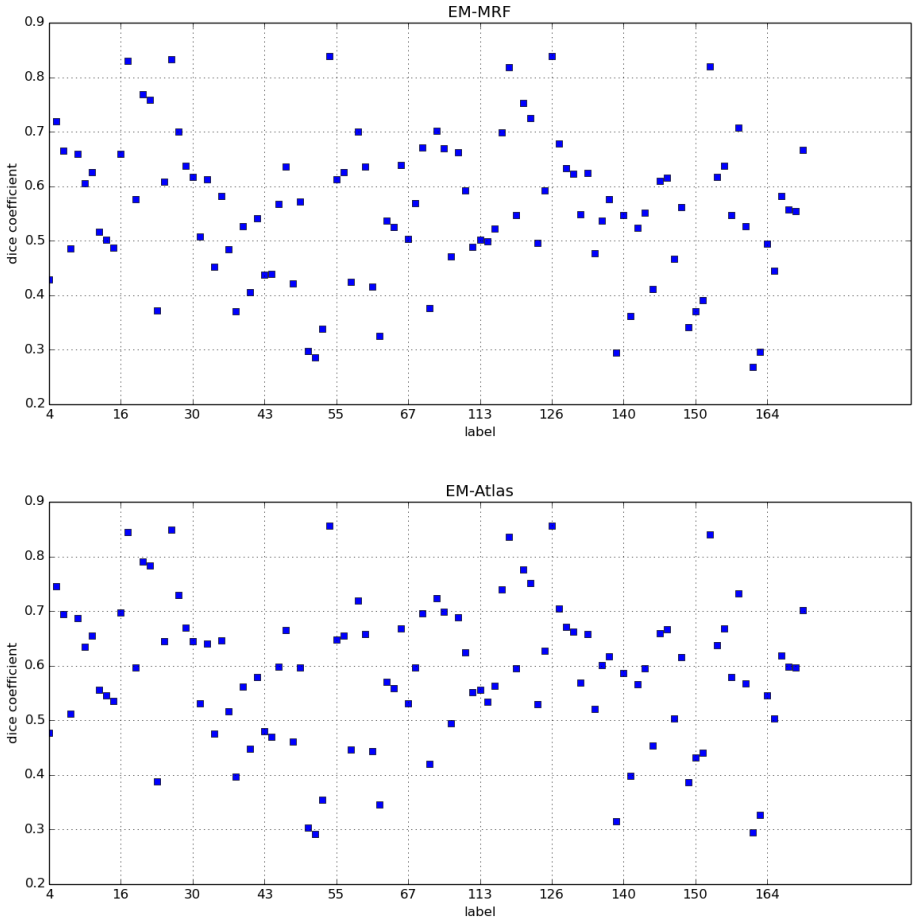

Figure 14: Dice score of MRF classification and anatomical probabilistic atlas classification for all labels. The labels are organized from the highest to lowest presence. MGH data

5.4 Non local STAPLE

Non local STAPLE(NLS) has been evaluated as an instance of the expectation maximization algorithm. NLS analyses the correspondence of the target voxel and the atlas by using the intensity similarity and the spatial compatibility.

It has used two different spatial patch size. One patch includes the first order neighbourhoods, $1 \times 1 \times 1$, and another one that has a size $3 \times 3 \times 3$.

Shape reduced The shape of the images in both datasets have been reduced in order to execute the experiments in the available time frame. This has been done by taking the slices from 50 to 90 in the three dimension space, the resulting shape is $40 \times 40 \times 40$.

Miccai

Table 5 shows the average weighted and direct mean dice coefficient of the two evaluated approaches in the 15 permutations. The smallest size patch, $1 \times 1 \times 1$, has a better segmentation score, even with the smallest patch the algorithm has less information to establish a correspondence in the spatial patch.

	Weighted mean dice coefficient	Direct mean dice coefficient
NLS $1 \times 1 \times 1$	**0.763**	**0.591**
NLS $3 \times 3 \times 3$	**0.695**	**0.496**

Table 5: Weighted and direct mean dice coefficient of NLS with patch sizes: $1 \times 1 \times 1$ and $3 \times 3 \times 3$. Miccai Data.

Figure 15 shows three label slices of the same brain, one of the slices is segmentation provided by the experts, the other two are the label estimation of NLS approaches using two different patches.

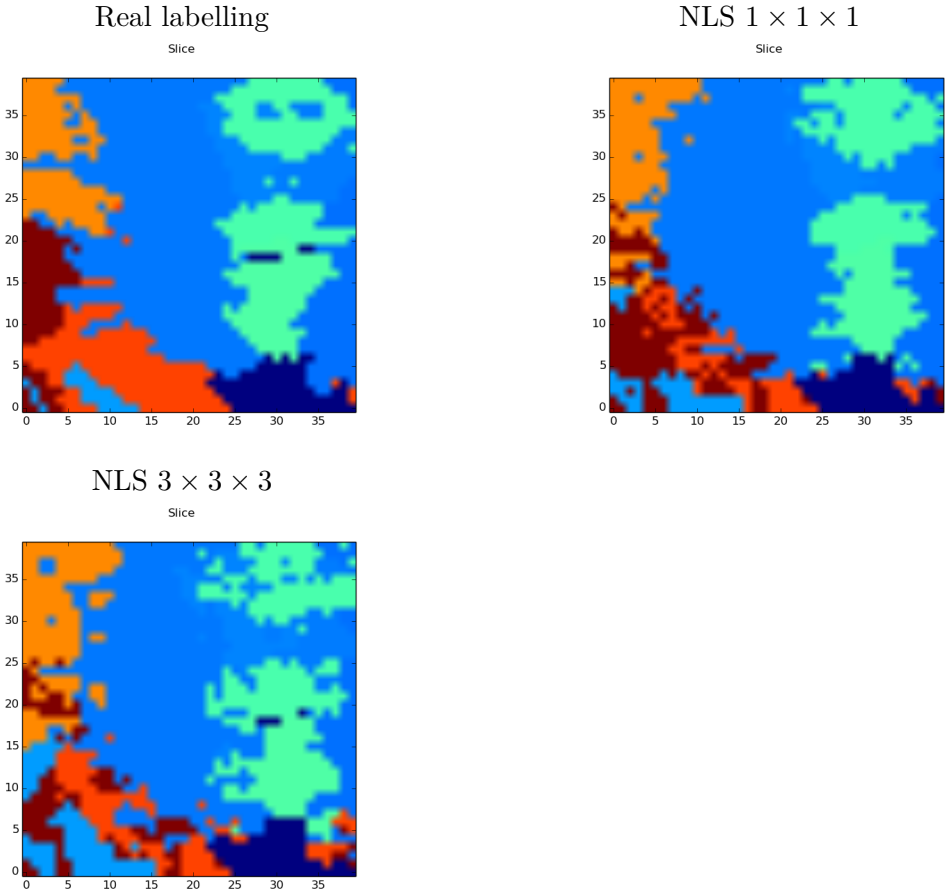

Figure 15: Three slices of the same brain that represents the quality labelling of NLS. Miccai Data

Figures 16 and 17 shows the average dice coefficient of each label for all permutations. Both plots are similar, there is not a clear correlation in the presence and the dice score of the labels.

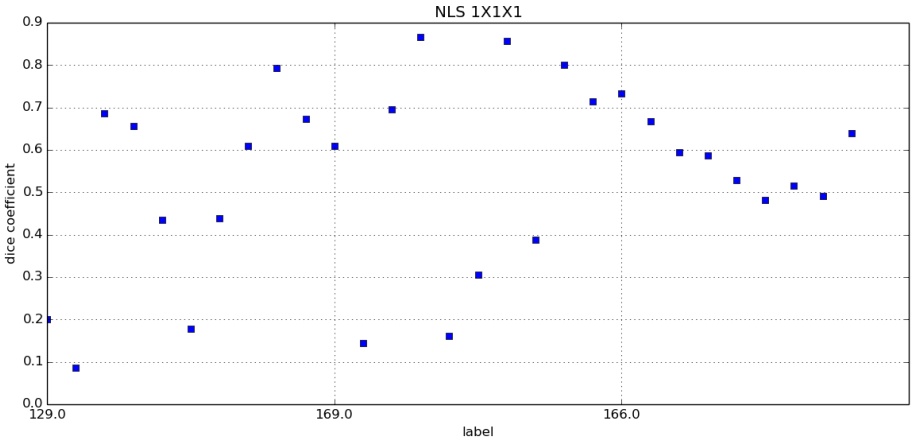

Figure 16: Dice score of NLS $1 \times 1 \times 1$ classification over all labels. The labels are organized from highest to lowest presence. Miccai data

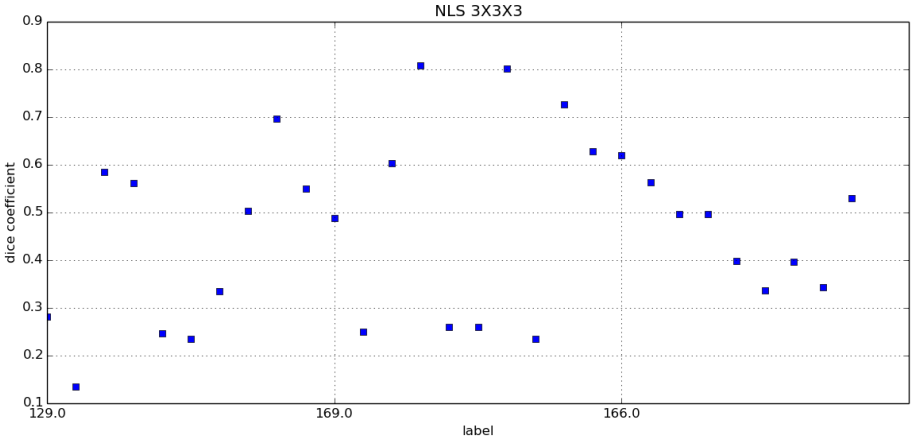

Figure 17: Dice score of NLS $3 \times 3 \times 3$ for all labels. The labels are organized from the highest to lowest presence. Miccai data

Figure 18 shows the calculated weighted mean dice for each performed permutation. It compares both used spatial patch sizes. As table 5 indicates, the smallest patch, $1 \times 1 \times 1$, has a higher average score. The plot shows that the smallest patch has a highest score for all the permutations. Also in the permutations 2, 5 and 12 in which the peaks are the lowest, the decrease is less for the smallest patch.

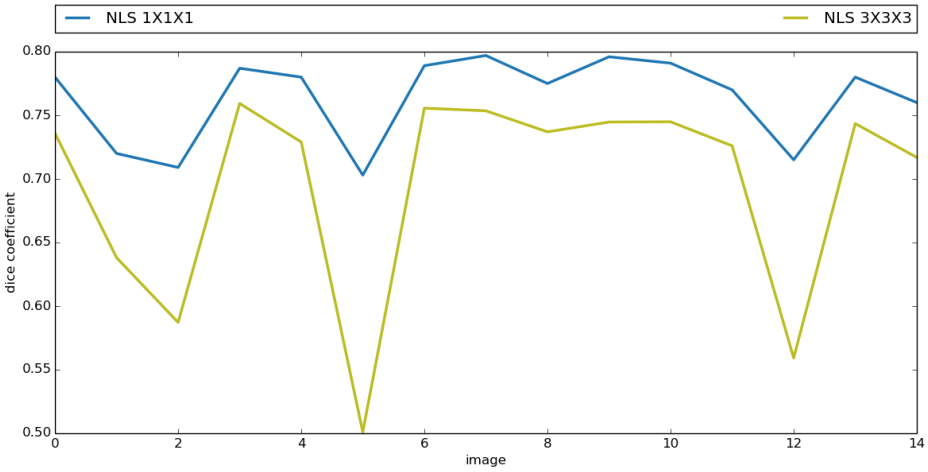

Figure 18: Weighted mean dice score comparative of two different patch sizes over each permutation in NLS. Miccai data.

MGH

Table 6 shows the average weighted and direct mean dice coefficient of the two evaluated approaches in the 10 permutations. Based on the dice scores, the spatial patch with size $1 \times 1 \times 1$ has a better segmentation than $3 \times 3 \times 3$.

	Weighted mean dice coefficient	Direct mean dice coefficient
NLS $1 \times 1 \times 1$	**0.680**	**0.364**
NLS $3 \times 3 \times 3$	**0.618**	**0.313**

Table 6: Weighted and direct mean dice coefficient of NLS with patch sizes: $1 \times 1 \times 1$ and $3 \times 3 \times 3$. MGH Data.

Figure 19 shows three label slices of the same brain, one of the slices is segmentation provided by the experts, the other two are the label estimation of NLS approaches using two different patches.

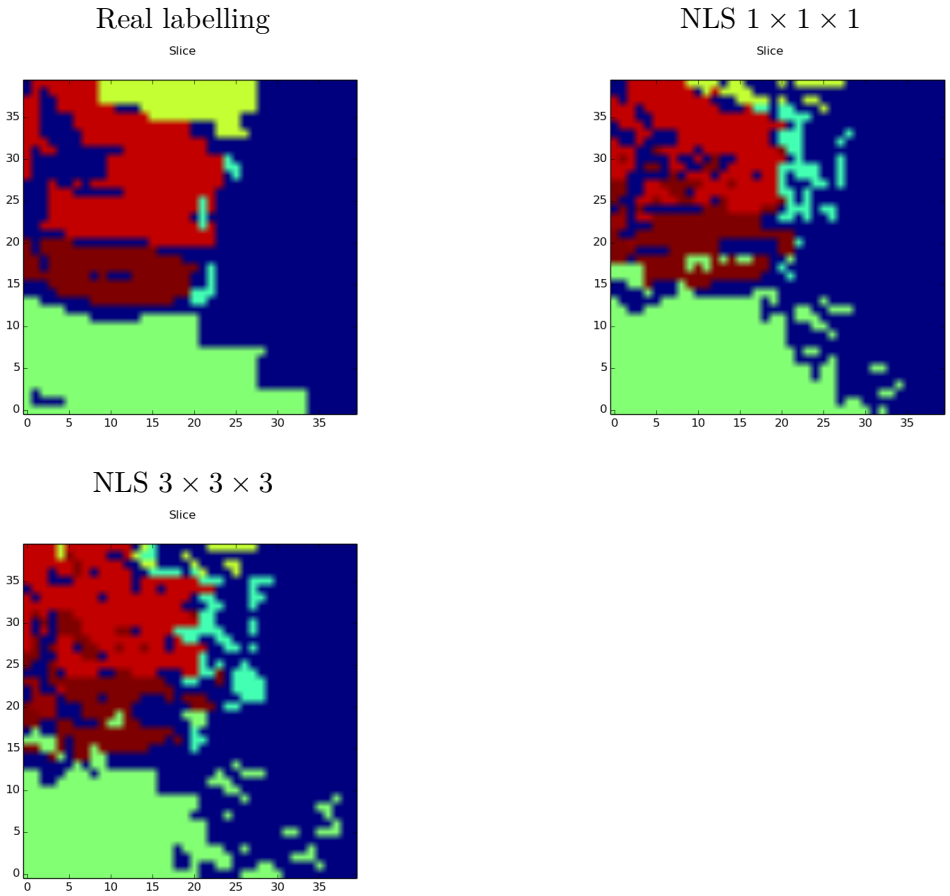

Figure 19: Three slices of the same brain that represents the quality labelling of NLS. MGH Data

Figures 20 and 21 show the average dice coefficient for each of the labels.

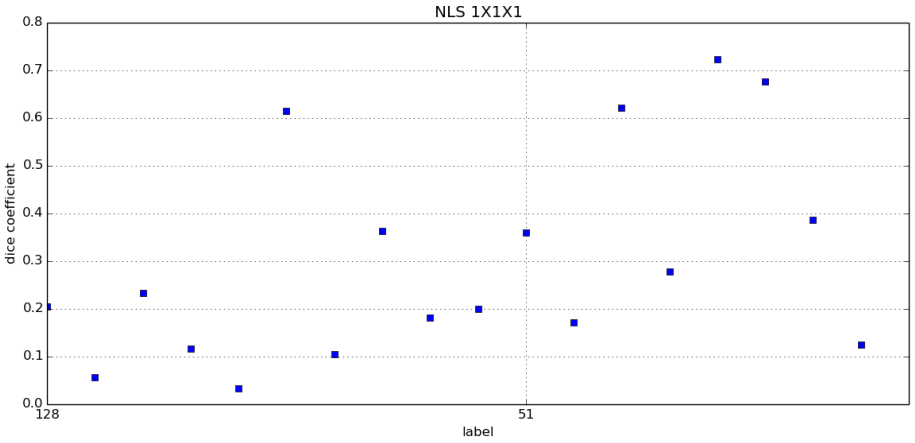

Figure 20: Dice score of NLS $1 \times 1 \times 1$ classification over all labels. The labels are order from higher presence to lower. MGH data

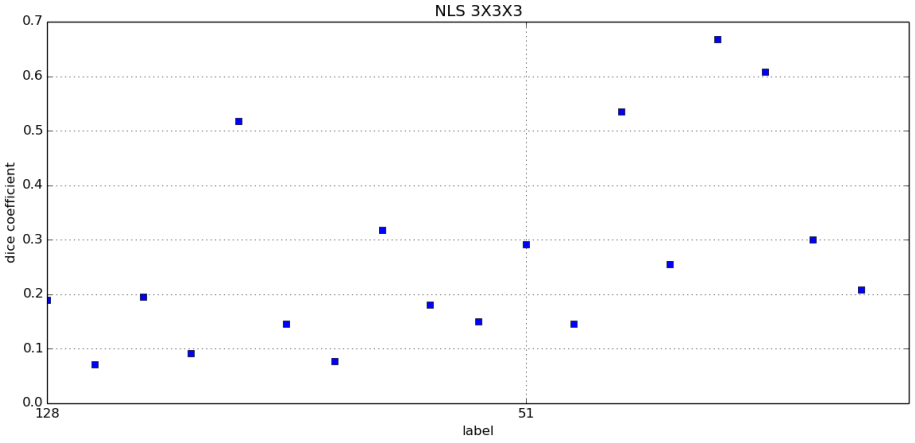

Figure 21: Dice score of NLS $3 \times 3 \times 3$ for all labels. The labels are order from higher presence to lower. MGH data

Figure 22 shows the weighted mean dice score for each permutation in the two evaluated spatial patch sizes. In both spatial patches, the first four permutations have a lower score than the total weighted mean dice score. As the table indicates, the smallest spatial patch has the best score segmentation for all evaluated permutations.

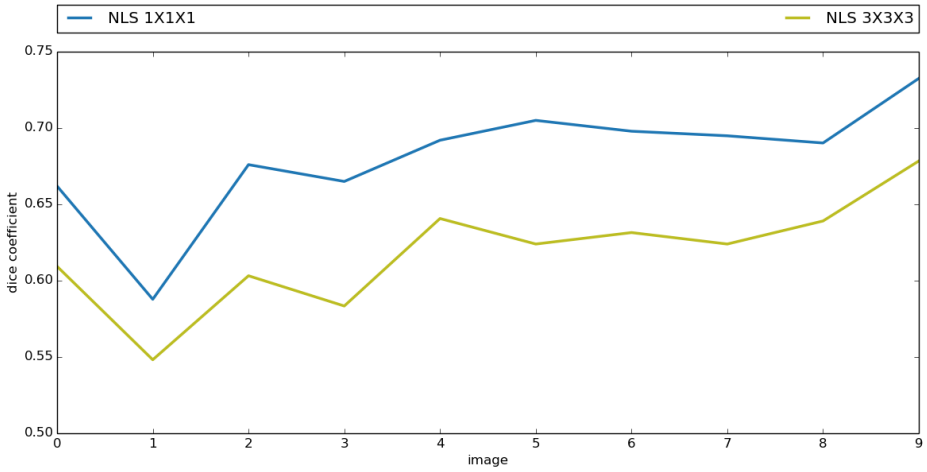

Figure 22: Weighted mean dice score comparative of two different patch sizes over each permutation in NLS. MGH data.

5.5 Attribute Similarity and Mutual-Saliency Weighting

Attribute Similarity and Mutual-Saliency Weighting (ASMSW) approach has been evaluated in both datasets. The first order neighbourhoods has been selected as the `core neighbourhood` and the second order neighbourhoods as `peripheral neighbourhood` for each target voxel. Then the expectation probability for all labels has been maximized for each target voxel, as it is explained in the section 4.5. As in other experiments, the shapes of both data-sets have been reduced to perform the experiments within a reasonable time. The evaluated images consists on the slice 50 to 99 from the original data sets. The evaluated shape of the images is $49 \times 49 \times 49$. The experiments results are shown in the next subsections.

Miccai

Table 7 shows the weighted mean dice coefficient and the direct mean dice coefficient. The results are similar to the MAP approaches, table 1. The direct mean dice is lower in ASMSW, because some labels with lower presence in the segmentation have an inferior score.

	Weighted mean dice coefficient	Direct mean dice coefficient
ASMSW	**0.802**	**0.621**

Table 7: Weighted and direct mean dice coefficient of ASMSW approach . Miccai data.

Figure 23 shows two slices of the same brain, one slice is the real segmentation of the brain, the other one is the segmentation done by ASMSW approach.

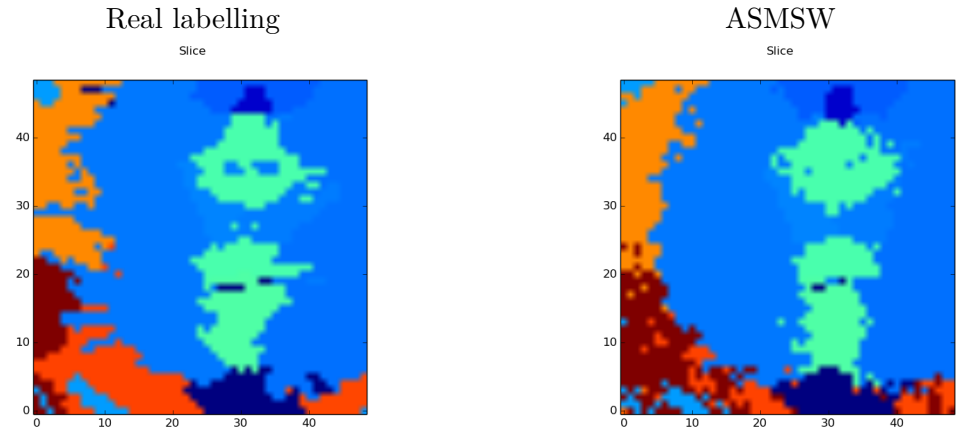

Figure 23: Slice of the same brain presenting the quality labelling of ASMSW. Miccai data

Figure 24 shows the dice score coefficient of all label in the 15 permutations. There is not a clear pattern, as there is in maximum likelihood or maximum a posteriori approaches, that the dice score is influenced by the highest presence of the label in the image. This is not such case, because the estimation is not based on a probability density function, it is based on the calculating of the similarity of the target adjacent voxels in the image and in the atlas.

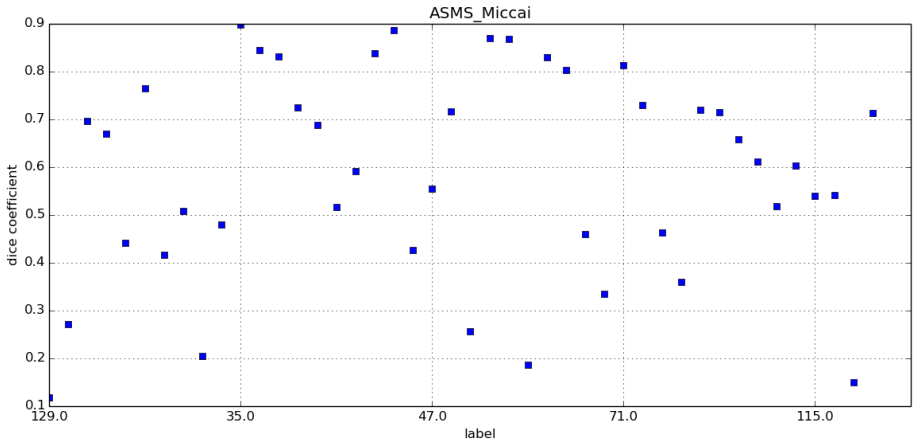

Figure 24: Dice score ASMS for all labels. The labels are organized from the highest to lowest presence. Miccai data

Figure 25 shows the weighted dice coefficient for each of the carried out permutations in the *test-one-leave-out* strategy. In the permutations 1,2,5 and 12, there is a clear under estimation of the ground truth, they correspond to the images 2,3,6 and 13. The dice score coefficient of these permutations is under is below the average, 0.802 (table 7).

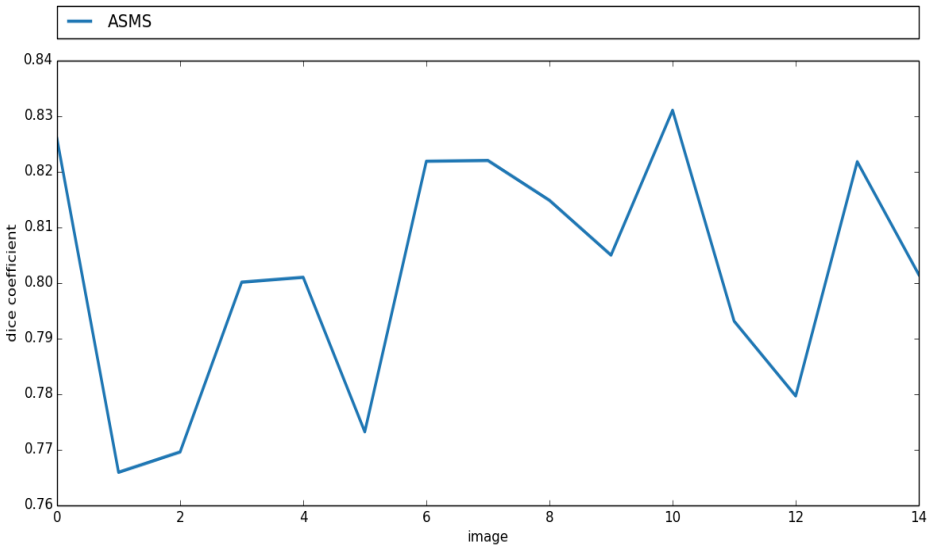

Figure 25: Weighted dice score comparative over each permutation in ASMS. Miccai data.

MGH

Table 8 shows the weighted mean and direct mean dice coefficient over the 10 permutations. The results are similar to MRF and ML approaches, even though the score is below the scores of MAPs and probabilistic atlas approaches as presented in table 2 and table 4.

	Weighted mean dice coefficient	direct mean dice coefficient
ASMSW	**0.716**	**0.518**

Table 8: Weighted and direct mean dice coefficient of ASMSW Weighting approach . MGH Data.

Figure 26 shows the same slice for the segmentation of ASMSW and the ground truth of the slice.

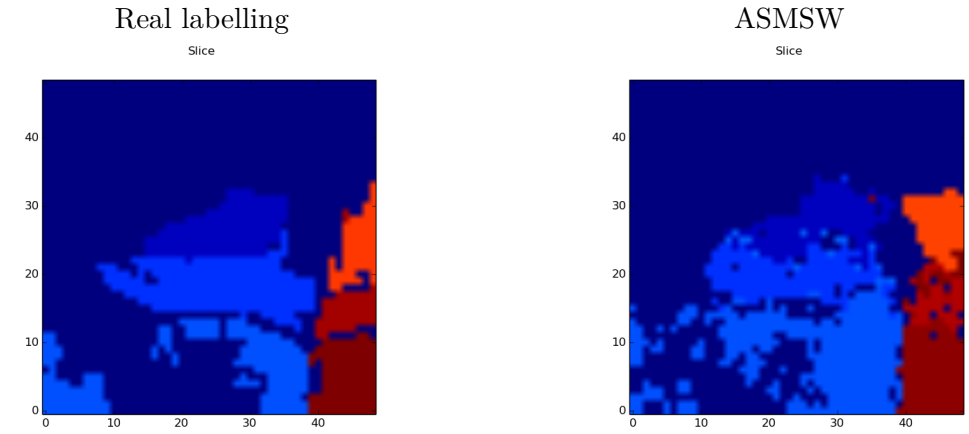

Figure 26: Slice of the same brain representing the quality labelling of ASMSW. MGH Data

Figure 27 shows the dice score of the 22 labels in the reduced brain image.

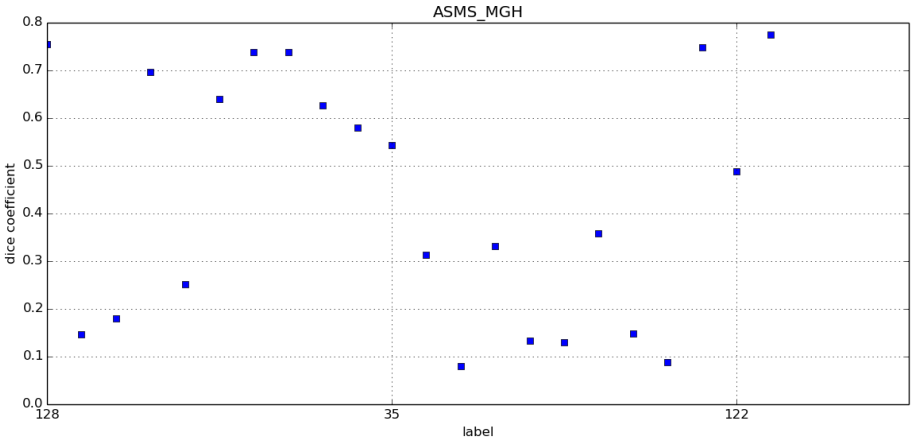

Figure 27: Dice score ASMSW for all labels. The labels are organized from the highest to lowest presence. MGH data

Figure 28 shows the weighted dice score for each permutation. There are three lowest peaks, in the permutations 1, 4 and 8, the score is below the average 0.716. These permutations correspond to test the images 2, 5 and 9 respectively.

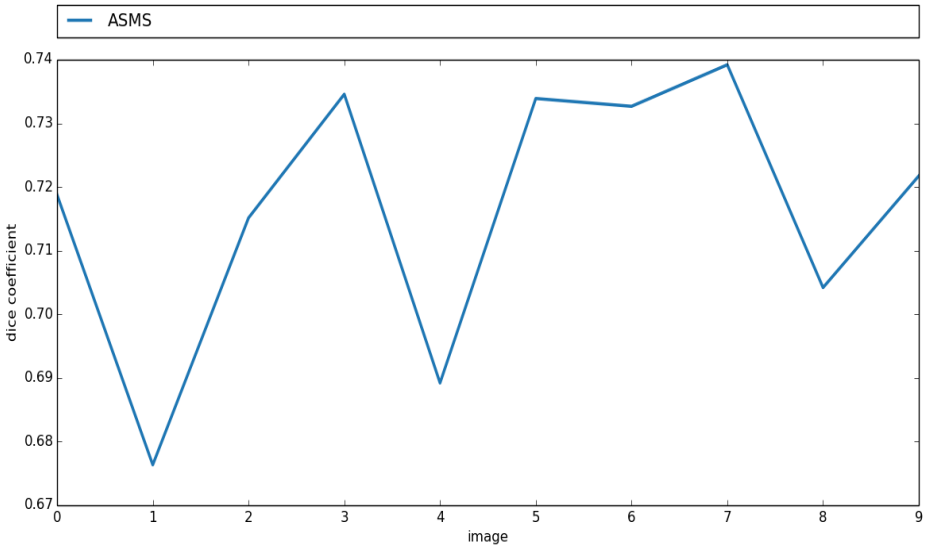

Figure 28: Weighted dice score comparative over each permutation in ASMSW. MGH data.

5.6 Support Vectors Machine

Support Vectors Machine(SVM) has been approached by slicing the image atlas into multiple 1-dimension slices. Each slice has been individually trained and tested using a linear SVM algorithm from the library *sckit-learn* [3]. Then all 1-dimension slices have been merged in order to get the 3-dimensional segmented image. A *one-against-one* strategy has been used for multi-label classification. Each slice has been divided into patches, in order to include 3-dimensional neighbourhood voxel information. The model has been fitted with n patches as samples. A sample consists on $size(patch)+1$ features, the intensities of voxels in the patch and the position of the patch in the 1-dimension slice. The target consists on the label information for each voxel. SVM has been tested with a *test-one-leave-out* strategy from all possible permutations in the atlas. One image is used as test image and the remaining are used as training sets in order to fit a model in SVM. Afterwards, the test image is divided in the same amount of 1-dimensions slices, as in the training model, and the corresponding label for each voxel in every SVM model is predicted.

The overall process is:

1º Create 1-dimensional slice in the train set images.

2º Create a SVM model for each slice with the intensity and the voxel location in the

[3]linear SVM: http://scikit-learn.org/stable/modules/svm.html

patch.

3º Create 1-dimensional slice in the test set images.

4º Calculate the expected label for each voxel in the image's slices using the corresponded SVM trained model.

5º Merge all segmented 1-slice image in order to get the complete 3-dimensional image.

Finally, the dice score coefficient for the tested images in each permutation is calculated.

Linear SVM approach has been tested with two different patch size. In one, the size is 1 and it does not include any neighbourhood. The other test includes the first order neighbours and thus the amount of voxels inside the patch is 28.

Shape reduced Due to the high shape of the images and the complexity of the algorithms, the shape for each image in the atlas has been reduced to $49 \times 49 \times 49$ by taking only the slices $50 - 99$ in the three dimensions.Through this adaptation has made the running and testing of the methods in a reasonable period of time. That has been achievable because the estimation of each voxel is influenced by its intensity, atlas label information and also by whether it used neighbourhood information, and the estimation is not influence by a global estimator.

Miccai Data

Table 9 shows the direct and weighted dice mean over all classes and test permutations.

	Weight mean dice coefficient	Direct mean dice coefficient
one voxel	**0.770**	**0.602**
first order	**0.527**	**0.351**

Table 9: Dice coefficient of SVM for one voxel and first order voxels size patch. Miccai Data.

Figure 29 shows the mean dice score coefficient in the 15 permutations of all estimated labels, the labels are organized from the highest to the lowest presence in the segmented brain images.

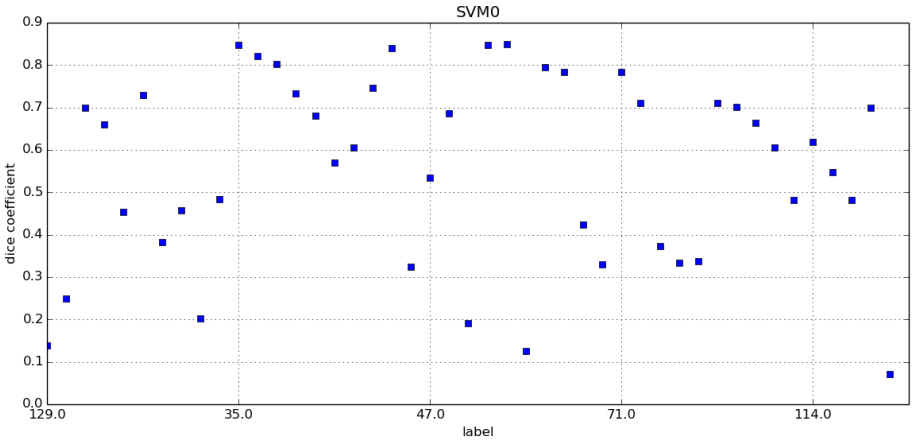

Figure 29: Dice score of SVM classifications over all labels. Miccai data

Figure 30 compares the weighted dice score of the two SVM evaluated versions. One version has a patch that is equal to one, just one voxel per patch. The other version includes the target voxel and the first order voxel in the 3-dimensional space in the patch. As in the other evaluated methods, there is underestimation in the permutations 5 and 12, that corresponds to the images 6 and 13 in the atlas, where the weighted dice scores are below 0.7 for the best SVM evaluated version, SVM with on voxel in the patch. Furthermore it is clear that the SVM version that does not include neighbourhood voxel intensities performs better than the one that does it, since in all permutations the dice score is higher.

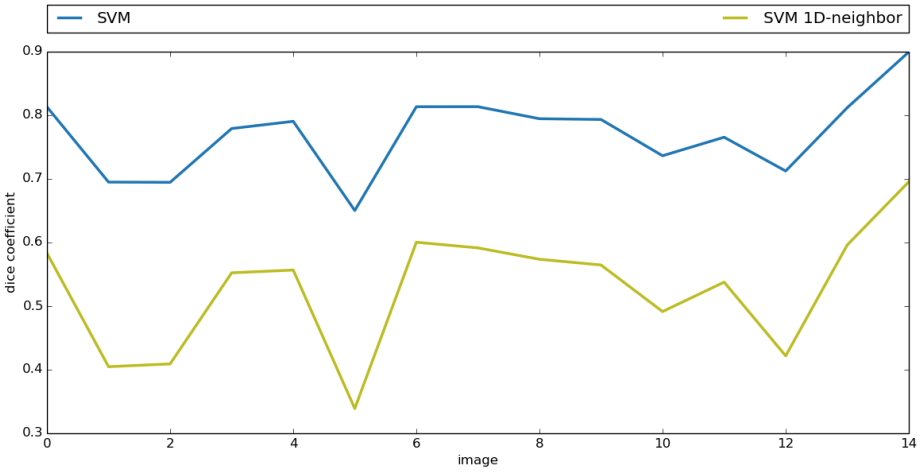

Figure 30: Weighted dice score comparative of two different patch's size over each permutation.Miccai data

Figure 31 shows three the label slices of the same brain, one slice is the ground truth. The rest are the SVM segmentation, the *SVM-one voxel feature* is relatively similar to the real labelling. Meanwhile, *SVM-first order neighbourhood features* has a really poor segmentation. The volumes are diffuse in the slice image, there is not visual correlation with the ground truth. This image slice and the dice score, table 9, show that *SVM-first order neighbourhood version* has not approached well the feature selection in order to build the SVM model.

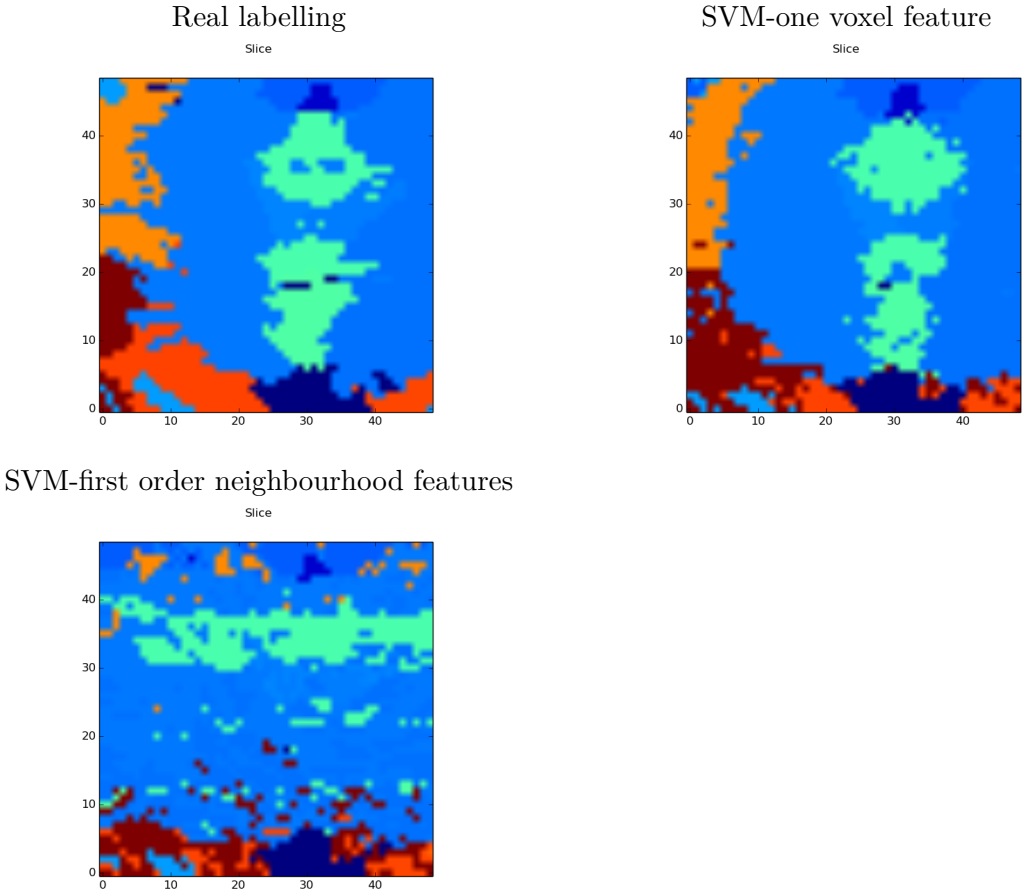

Figure 31: Slice of the same brain representing the quality labelling of SVM. Miccai data

MGH

Table 10 shows the direct and weighted dice mean over all classes and test permutations.

	Weight mean dice coefficient	Direct mean dice coefficient
one voxel	**0.705**	**0.501**
first order	**0.554**	**0.310**

Table 10: Dice coefficient of SVM for one voxel and first order voxels size patch. MGH Data.

Figure 32 shows the mean dice score coefficient in the 10 permutations of all estimated labels, the labels are organized from the highest to lowest presence in the segmented brain images.

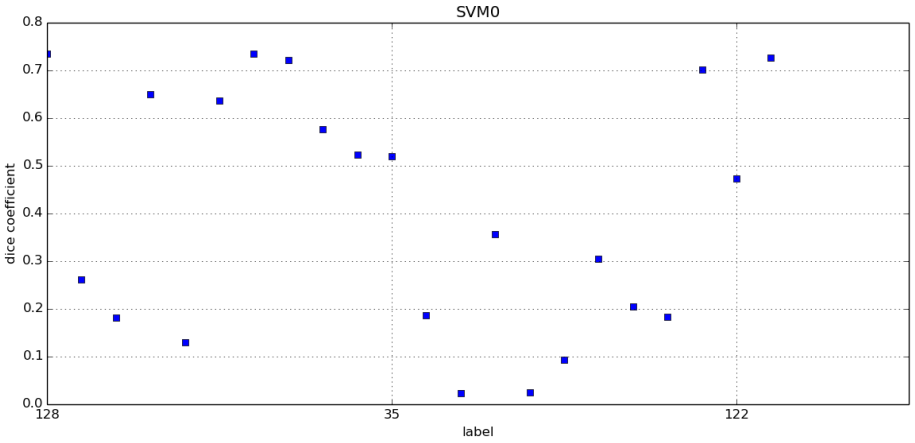

Figure 32: Dice score of SVM classifications over all labels. MGH data.

Figure 33 compares the weighted dice score of the two SVM evaluated versions. One version with patch equal to one, just one voxel per patch. The other version that includes the target voxel and the first order voxel in the 3-dimensional space in the patch . The permutations 5 and 8 that corresponds to the images 6 and 9 in the MGH atlas, are under the weighted dice coefficient, 0.705, of SVM with one voxel patch. These under estimations match with the under estimations in the expectation-maximization approaches.

Figure 34 shows three label slices of the same brain, one slice is the ground truth and the other two are the label segmentation of two evaluated SVM. *SVM-first order neighbourhood features* slice has discontinuity in the segmentation, some of labels are spread across the slice image and do not define a volume.

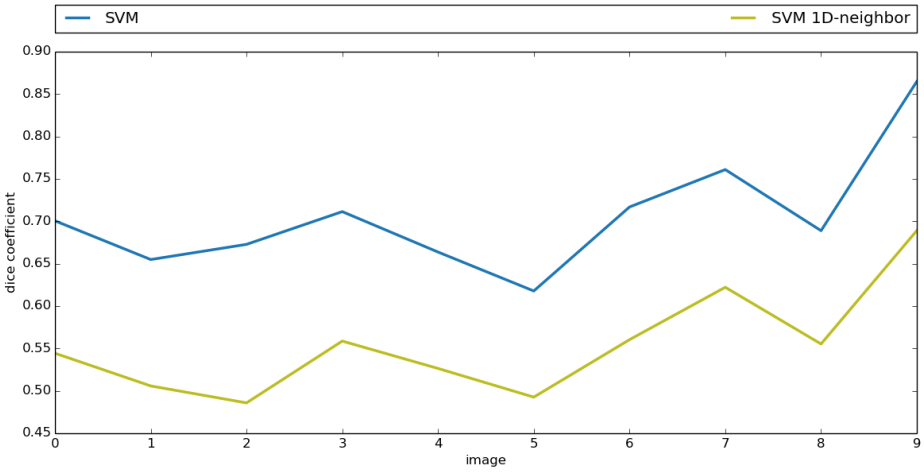

Figure 33: Weighted dice score comparative of two different patch's size over each permutation. MGH data.

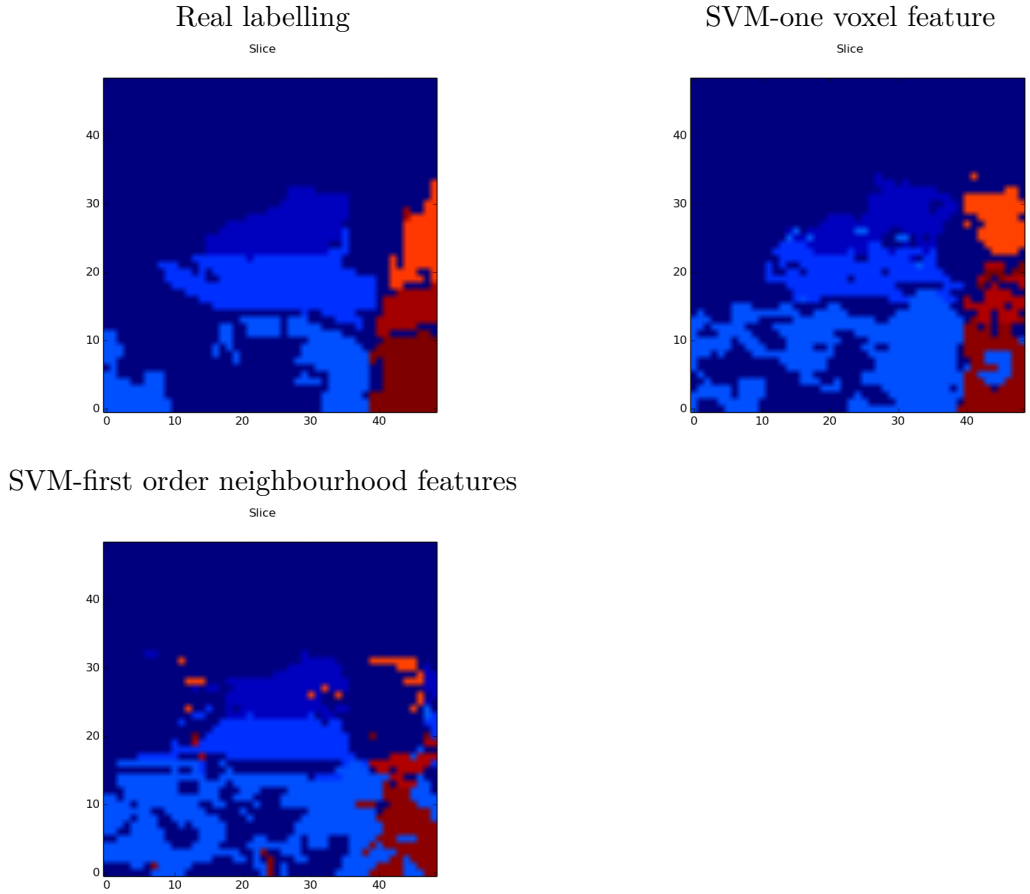

Figure 34: Slice of the same brain representing the quality labelling of SVM. MGH data

6 Conclusion

Different segmentations approaches have been presented throughout this document, Expectation-Maximization(EM) methods, Attribute Similarity and Mutual Saliency Weighting(ASMSW), and Support Vector Machine(SVM). EM covers different estimators and different strategies in order to estimate a label for each voxel in the image. EM has been extend to add spatial consistency based on the neighbourhood information such as MRF and NLS. All the methods have been evaluated in order to calculate a dice score coefficient that measures the overlap of the segmented volumes and the real label segmentation. Two data sets, MICCAI and MGH with 15 and 10 samples respectively, have been used with a registration method presented by Hansen and Hansen [4].

EM is an algorithm that iterates until converge in a maximum probability that defines a label for a target voxel. A mixture model is first introduced, it is based on Gaussian distributions from all set labels and intensities in the given atlas. Maximum Likelihood(ML) estimator defines a label to a voxel which the mass function is higher for the intensity voxel over the set of Gaussian distribution. MRF method expands this ML in order to introduce a spatial consistency over the neighbourhood voxels. Maximum a Posteriori(MAP) also expands ML, it introduces a priori probability. NLS introduces a non local spatial approach. ASMSW measures the similarity of the neighbourhood in order to create a probability with most similar voxels across the atlas.

Lastly, Support Vector Machine(SVM) has been fitted to build a segmentation method. The proposed method slices each image from the data base in the same amount of slices, then trains a model for each slice and predicts the target label for each model. Finally, it merges all slices to reconstruct the original database. Two versions have been presented in order to create the SVM models, one version that only includes information of the intensity and position of the voxel in the slice, and another version that creates neighbourhood patches in order to extend the SVM model features to the three dimensional neighbourhood space. This second version shows poor results in the performed experiments, better feature selection will be desirable in future work such as introducing similarity measures over the neighbourhood patch like sum of square differences.

Table 11 and table 12 shows the dice scores for all evaluated methods in both data-sets Miccai and MGH. MAPs approaches have a better segmentation for the rest evaluated methods. SVM and NLS version with large spatial patch perform the worst segmentation for both cases. MAP approaches compare to ASMSW have similar results, the main difference is the running time. ASMSW is more complex than MAPs, and it takes much time to complete the segmentation. There is a lightly difference between ML and MRF, but it is not relevant. In the case of MRF, the included spatial consistency does not contribute to a better segmentation.

	Weight mean dice coefficient	Direct mean dice coefficient
MAP-1n	**0.801**	**0.681**
MAP	**0.801**	**0.680**
ASMSW	**0.802**	**0.621**
Anatomical prob. atlas	**0.796**	**0.672**
ML	**0.768**	**0.622**
MRF	**0.766**	**0.618**
SVM one voxel	**0.770**	**0.602**
NLS $1 \times 1 \times 1$	**0.763**	**0.591**
NLS $3 \times 3 \times 3$	**0.695**	**0.496**
SVM first order	**0.527**	**0.351**

Table 11: Weighted and direct mean dice coefficient for all evaluated methods. Miccai Data.

	Weight mean dice coefficient	Direct mean dice coefficient
MAP-1n	**0.748**	**0.602**
MAP	**0.744**	**0.596**
Anatomical prob. atlas	**0.739**	**0.589**
ASMSW	**0.716**	**0.518**
ML	**0.718**	**0.560**
MRF	**0.715**	**0.556**
SVM one voxel	**0.705**	**0.501**
NLS $1 \times 1 \times 1$	**0.680**	**0.364**
NLS $3 \times 3 \times 3$	**0.618**	**0.313**
SVM first order	**0.554**	**0.310**

Table 12: Weighted and direct mean dice coefficient for all evaluated methods. MGH Data.

These results are important because they present that it is possible to achieve a better segmentation based on the similarity of formed local clusters in the image. Besides in the case of the data-set MGH, the results show that the segmentation improves over the template based labelling,5.1.1 , in the approaches: MAPS, Anatomical probabilistic atlas, ASMSW, ML, MRF and SVM-one voxel.

It is interesting to certify the results in future work by using the whole shape in the approaches ASMSW, SVM and NLS. Furthermore it would be possible to reduce the amount of labels by fusing the labels, in this way there is more information for each new created label in the segmentation methods and less chance of error.

References

[1] G.E. Christenses, H.J. Johnson, *Consistent Image Registration*, IEEE Transactions on medical imaging, vol. 20, no. 7, 2001.

[2] S.Darkner, J.Sporring, *Locally Orderless Registration*, IEEE Transactions on pattern analysis and machine learning, vol. 20, 2012

[3] S.Darkner, Akshay Pai, Matthew G. Loprot, J.Sporring, *D3:Discrete Diffeomorphic Deformatios for Medical Image Registration*, 2016.

[4] Casper Hansen, Christian Hansen, *Benchmarking of template strategies*, Project reasearch at University of Copenhaguen, 2015.

[5] C. Studholme, D.L.G. Hill , D.J. Hawkes, *An overlap invariant entropy measure of 3D medical image aligment*, Pattern Recognition, vol. 32, no.1, pp. 71-86, 1999.

[6] Christian Ledig , Robin Wolz , Paul Aljabar , Jyrki Lötjönen, Rolf A. Heckemann,Alexander Hammers, Daniel Rueckert, *Multi-class brain segmentation using atlas propagation and EM-Based refinement*, IEEE,2012 (896-899)

[7] M. Jorge Cardoso and Matthew J. Clarkson and Gerard R. Ridgway and Marc Modat and Nick C. Fox and Sebastien Ourselin, *LoAD:A locally adaptive cortical segmentation algorithm*, NeuroImage, 56, 3, pages 1386-1397. 2011

[8] Aubert-Broche, B., Griffin, M., Pike, G.B., Evans, A.C., Collins, D.L. *Twenty new digital brain phantoms for creation of validation image data bases.*, IEEE Trans. Med.Imaging 25 (11), 1410–1416 (Nov). 2006.

[9] Pham, D.L. *Robust fuzzy segmentation of magnetic resonance images.*, Computer-Based Medical Systems pp. 127–131 (Jan) .2002

[10] Ashburner, J., Friston, K.J. *Unified segmentation*, Neuroimage 26 (3), 839–851 (Jan). 2005.

[11] Zhang, Y., Brady, M., Smith, S.M. *Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm.*, IEEE Trans. Med. Imaging 20 (1), 45–57. 2001.

[12] Christian Ledig, Rolf A. Heckemann, Paul Aljabar, Robin Wolz, Joseph V.Hajnal, Alexander Hammers, Daniel Rueckert, *Segementation of MRI brain scans using MALP-EM*, MICCAI Grand Challenge and Workshop on Multi-Atlas Labeling , 2012

[13] D. Rueckert, L. I. Sonoda, C. Hayes, et al. *Nonrigid registration using free-form deformations: Application to breast MR images*, IEEE TMI, vol. 18, no. 8, pp.712–721, 1999.

[14] M. Modat, G. R. Ridgway, Z. A. Taylor, et al. *Fast free-form deformation using graphics processing units*, Computer Methods and Programs in Biomedicine, vol.98, no. 3, pp. 278–284, 2010.

[15] X. Artaechevarria, A. Munoz Barrutia, and C. Ortiz de Solorzano *Combination strategies in multi-atlas image segmentation: Application to brain MR data*, IEEE TMI, vol. 28, no. 8, pp. 1266–1277, 2009.

[16] K. Van Leemput, F. Maes, D. Vandermeulen, and P. Suetens *Automated model-based tissue classification of MR images of the brain*, IEEE TMI, vol. 18, no. 10, pp. 897–908, 1999.

[17] Andrew J.Asman, Bennett A.Landman, *Characterizing Spatially Varying Performance to Improve Mutil-Atlas Multi-Label Segmentation*, MICCAI Grand Challenge and Workshop on Multi-Atlas Labeling , 2012

[18] Simon K. Warfield, Kelly H. Zou and William M.Wells *Simultaneous Truth and Performance Level Estimation (STAPLE): An Algorithm for the Validation of Image Segmentation*, IEEE TMI, 2004

[19] Andrew J. Asman, and Bennett A. Landman *Non-Local Statistical Label Fusion for Multi-Atlas Segmentation*, Medical Image Analysis, 2012

[20] Jeffrey Dean and Sanjay Ghemawat *MapReduce: simplified data processing on large clusters*, Commun. ACM 51, 1, 107-113. 2008.

[21] Yangming Ou, Jimit Doshi, Guray Erus, and Christos Davatzikos *Attribute Similarity and Mutual-Saliency Weighting for Registration and Label Fusion*, MICCAI Grand Challenge and Workshop on Multi-Atlas Labeling , 2012

[22] Yangming Ou, Aristeidis Sotiras, Nikos Paragios, Christos Davatzikos *DRAMMS: Deformable registration via attribute matching and mutual-saliency weighting*, Medical Image Analysis Volume 15 , Issue 4 , pp. 622 - 639 . 2010

[23] Buades, A., Coll, B., Morel, J.M *A non-local algorithm for image de-noising*, Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 60-65. 2005

[24] Coupé, P., Manjkn, J.V., Fonov, V., Pruessner, J., Robles, M., Collins, D.L. *Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation*, Neuroimage Volume 54, pp. 940-954.

[25] Christopher M.Bishop *Pattern recognition and machine learning*, Springer, Chapter 9, pp.423-455. 2006

[26] Christopher M.Bishop *Pattern recognition and machine learning*, Springer, Chapter 7, pp.325-344. 2006