

Information Processing in Medical Imaging

MPHYGB06 - Coursework Report

Student Number - 14062340

6th April 2018

Word Count - 3,555

Step 1

1.1 Groupwise registration

The 10 provided T1 weighted template brain MRI images were used to create a groupwise space and mean template image that are representative of the given population. As requested in the question sheet, all the image registration tasks performed in this coursework were carried out using NiftyReg tools. The groupwise registration proceeded as follows (using the script `GroupWiseReg.m`) in 12 discrete registration steps (each of which was composed of 10 separate image registrations), requiring 110 minutes to complete:

- An initialisation step was performed in which each of the 10 provided images (floating image) was registered with the first image (`template_0_img.nii`, which was arbitrarily allocated as the reference image) using a rigid only registration algorithm. This rigid only registration was carried out by calling '`reg_aladin -rigOnly`'. This step was performed to prevent the propagation of bias throughout the groupwise registration toward the first image that was used as a reference.
- For each subsequent registration step, the 10 provided images were used as the floating images. Also, an average of the 10 output warped images was calculated at each registration step (using the function `CalcMeanImage.m`, which makes a call to `reg_average`). This was then used as the reference image for the next registration step; each of these mean images is shown in Figure 1.
- For registration steps 2-6, `reg_aladin` was called using its default rigid and affine transformation model options. An affine transformation model was used for these steps as each image is registered to an *average* image, meaning that no transformation bias toward a specific individual image would occur. For these registration steps, the output affine transformation matrix from the previous step was used to initialise the current step.
- For registration steps 7-12, `reg_f3d` was called to carry out a non-rigid registration. This transformation model was used as it has many more degrees of freedom than the previous two transformation models, and would thus produce an increasingly sharper average image with better alignment at each iteration. For these registration steps, the output affine transformation matrix from step 6 was used to initialise the current step. Initialisation of steps 8-12 using the non-linear transformation parameterisations produced by the *previous* registration step was attempted; however, this proved to be computationally nonviable.
- At each registration step, the normalised cross correlation (NCC) between each of the 10 output warped images and the reference image for that step was calculated and averaged. This was carried out by calling '`reg_measure -ncc`' in the function `CalcMeanNCC.m`. The results of this are shown in Figure 2; when the gradient of this plot approached zero, the respective registration stage was thought to have converged.

The mean image and the 10 non-linear transformation parameterisations produced by the last registration step were the final output of the groupwise registration process (see image 12 in Figure 1). The mean NCC value was 0.65 after the initialisation step, this reached a value of 0.85 after step 6 (after 5 rigid and affine registration steps), and it plateaued at 0.97 after a further 6 non-rigid registration steps. This rate of improvement in the mean NCC value is reflected in the quality of the mean images shown in Figure 1, with images 8 and onward showing the sharpest visual results.

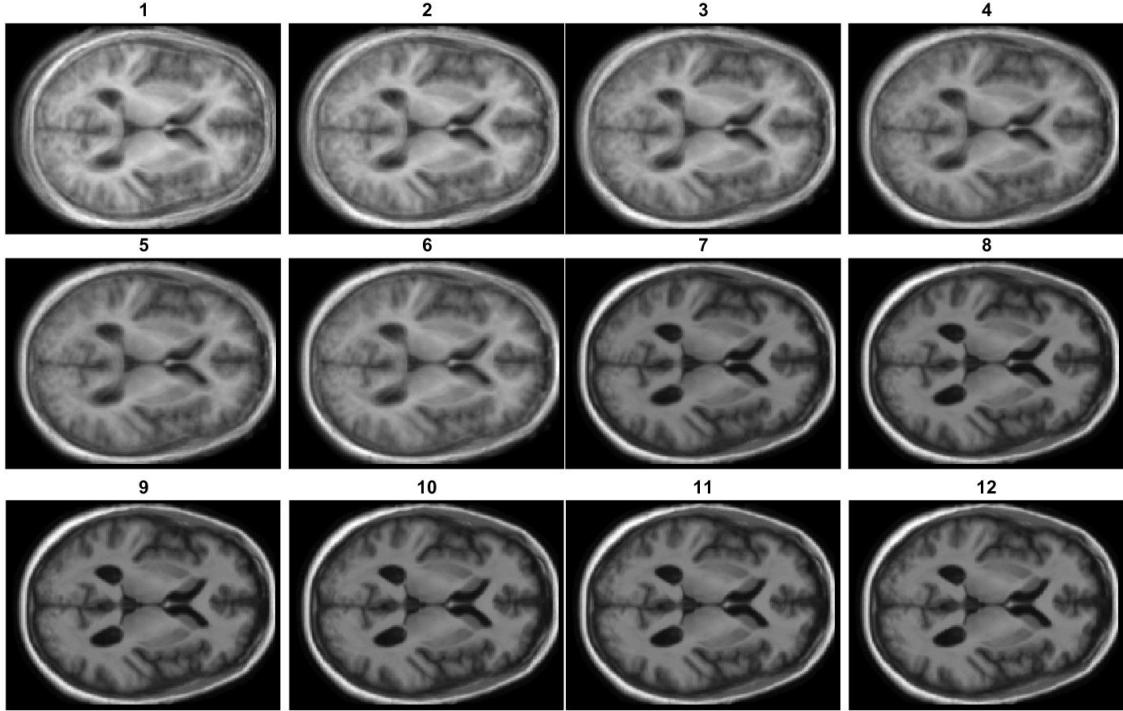


Figure 1: The progression of the groupwise mean image at each registration step. N.B. All 3D volumes are represented by a single orthogonal plane slice in this coursework.

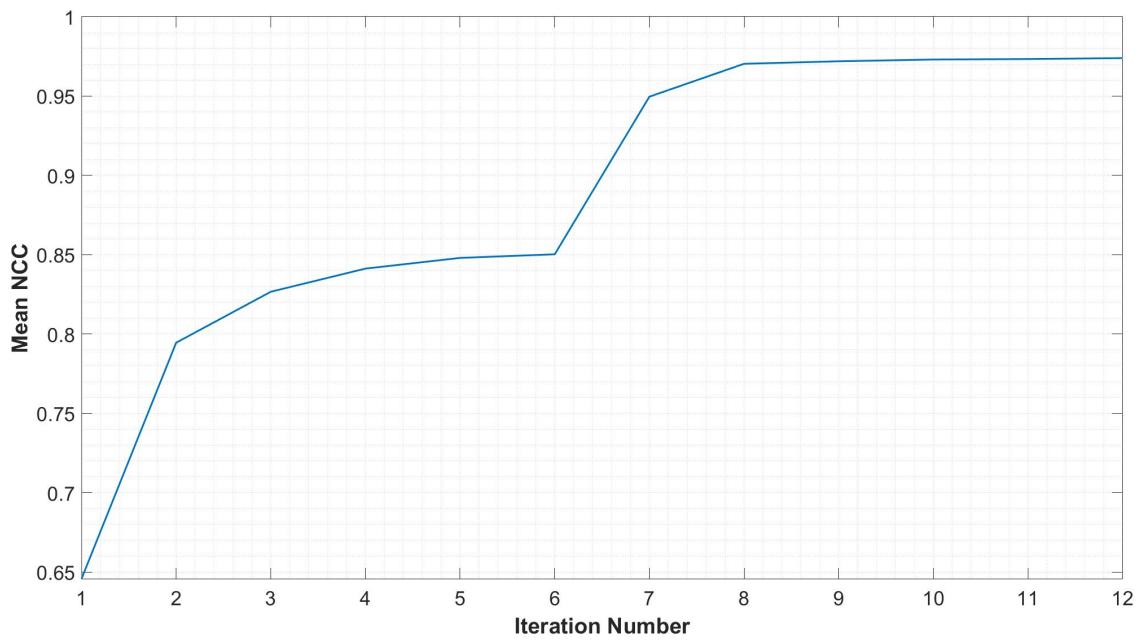


Figure 2: The mean NCC value across all 10 images at each registration step.

1.2 Generation of Tissue Probability Maps

Visual inspection of the 10 provided segmentation images revealed the following label correspondences:

- 0 - Non-brain (fat, skin, dura, skull).

- 1 - Cerebro-spinal fluid (CSF).
- 2 - Grey matter (GM).
- 3 - White matter (WM).

The segmentation images were resampled into the groupwise space using the mean template and the non-linear parameterisations obtained from the groupwise registration process (using the script `TPM.m`). This was done by making a call to ‘`reg_resample -inter 0`’. A zero order nearest neighbour interpolation model was used as the segmentation labels are binary values. Once these segmentations had been propagated into the groupwise space, they were concatenated into one 4D matrix, and the number of occurrences of each label at each voxel was counted. This yielded a integer number between 0 and 10 for each label at each voxel in the groupwise space, which was divided by the number of segmentation images (10) to yield the probability of that label occurring at that voxel. The resulting 4 tissue probability maps (TPMs) are shown in Figure 3.

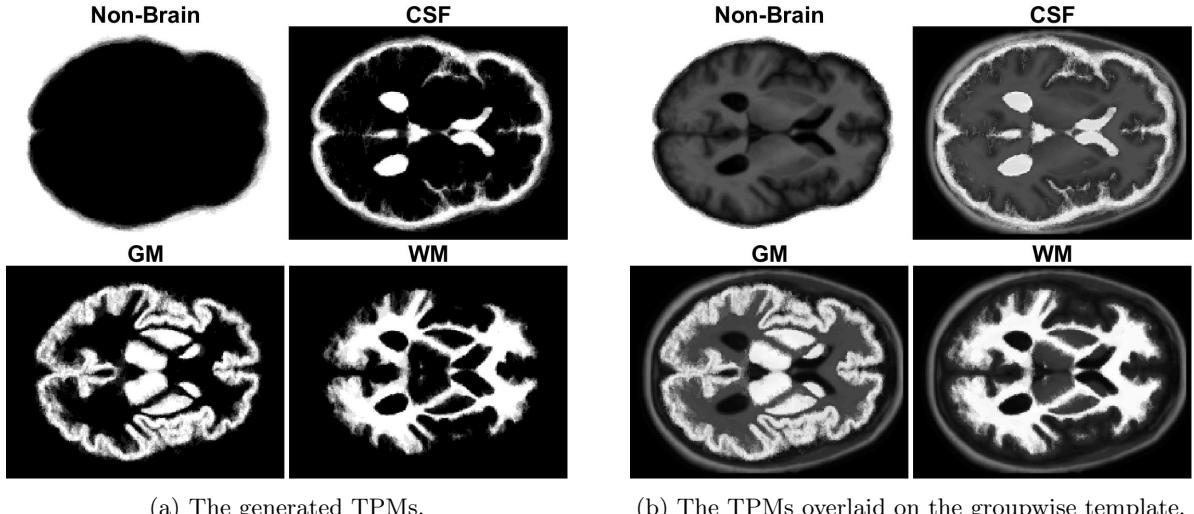


Figure 3: The 4 TPMs generated by propagating the 10 segmentation images into the groupwise space.

As the resampling and thresholding of the 10 segmented images is relatively computationally inexpensive, this section required only 15 seconds to complete. A check was carried out to ensure that the sum of all 4 of the TPMs at each voxel was equal to 1. Verification of the spatial alignment of the TPMs with the groupwise template is demonstrated in Figure 3 (b).

Step 2

2.1 Tissue Probability Map Propagation

The 4 TPMs derived in section 2.1 were propagated from the groupwise space into the space of each of the 20 provided non-segmented brain MRI images. This was done using a two-step registration and resampling process for each image (using the script `TPMProp.m`). In order to improve the quality of the registrations, a binarised version of the groupwise template image (acquired via a call to `reg_tools -bin`) was used as a mask in the floating image space for both of the registration steps.

1. The groupwise mean template (floating image) was globally registered to the non-segmented image (reference image) by calling ‘`reg_aladin`’.
2. The affine matrix obtained from the above step was used to initialise the non-rigid registration (using the same floating and reference images as the first step) by calling ‘`reg_f3d`’. A visual summary of these first 2 steps is shown in Figure 4 (a).

3. The non-linear transformation parameterisation from the above step (which also contains the affine transformation from the first step) was then used to propagate the 4 TPMs (floating image) into the space of the non-segmented image (reference image). This was done by making a call to ‘`reg_resample -inter 1`’. A first order linear interpolation model was used in order to produce probabilities in the range of 0 to 1. When resampling a non-brain TPM, the argument ‘`-pad 1`’ was used to set the interpolation padding value to 1. For the other three TPMs, the argument ‘`-pad 0`’ was used to set the interpolation padding value to 0. A visual summary of this step is shown in Figure 4 (b).

This process yielded 4 propagated TPMs for each of the 20 non-segmented MRI images (i.e. 80 TPMs in total) with a run-time of 28 minutes. Again a check was carried out to ensure that the 4 TPMs associated with each image summed to 1 at each voxel. The mean NCC between each reference image and the output warped image of the second registration step was 0.76 across all 20 images, with a range of 0.63 to 0.87.

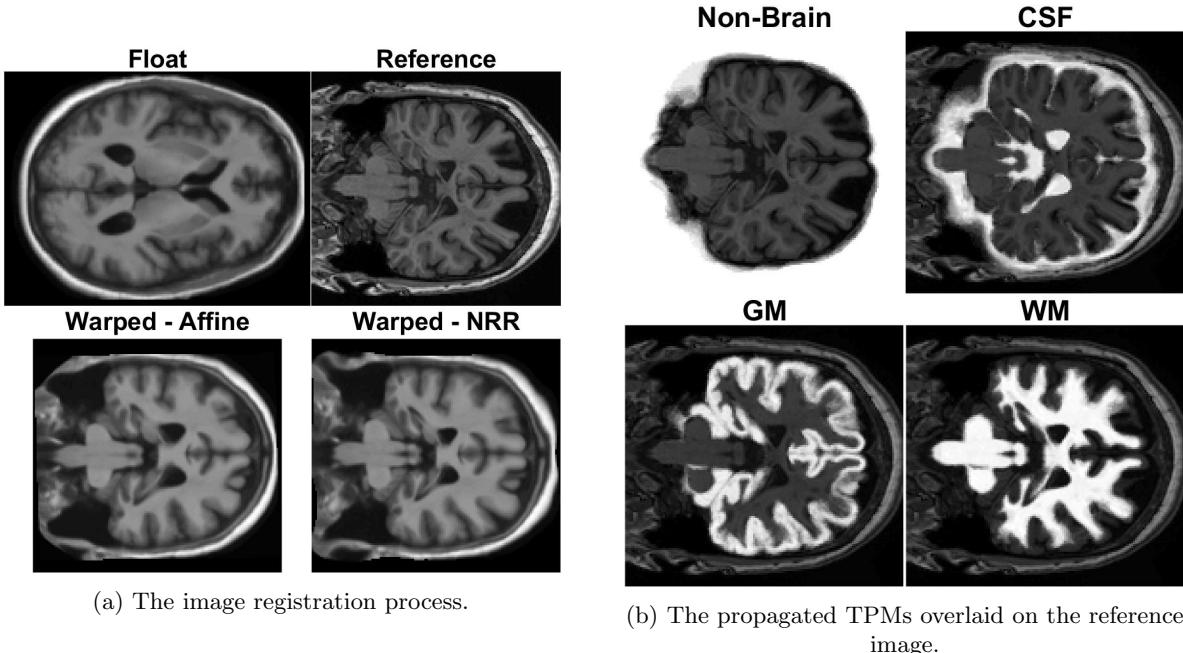


Figure 4: A visual summary of the propagation of the groupwise TPMs in to the space of `img_9.nii`.

2.2 Gaussian Mixture Model with Expectation Maximisation

A Gaussian Mixture Model (GMM) was implemented in order to segment the 20 non-segmented brain MRI images into the 4 tissue classifications described in section 1.2. The model was optimised using an Expectation Maximisation (EM) scheme, which made use of the TPMs generated in section 2.1 as tissue priors to initialise the segmentation with *a priori* information. The script `GMM.EM.m` receives an image to segment, its 4 tissue priors, a convergence threshold and a specified number of maximum iterations, and outputs the 4 segmentation images. A convergence threshold of 1×10^{-5} and 30 maximum iterations were the default parameters used for all segmentations in this coursework. For example, segmentation of `img_9.nii` took 8 iterations and 6 seconds to complete using these settings (Figure 5).

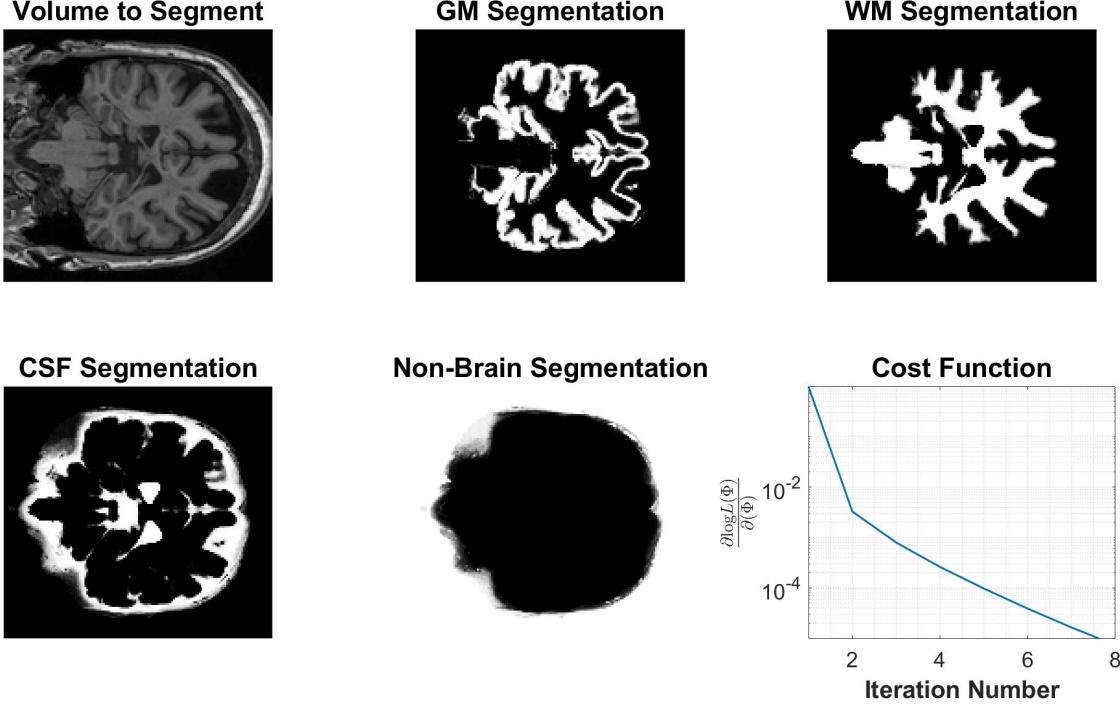


Figure 5: A visual summary of the segmentation of `img_9.nii`, using *a priori* information.

In general, an EM algorithm is one which finds maximum likelihood (ML) estimates of parameters in situations with missing data. Each component of the GMM framework used in this coursework, and the motivation for its use, is described below. This implementation assumes that every voxel intensity is independent, and that it is chosen at random from one of 4 classes.

- The input volume and its 4 priors were reshaped from 3D matrices to 1D vectors in order to speed up the implementation.
- y_i represents the intensity at pixel i , which is a 1D vector of length corresponding to the number of voxels in the volume to be segmented. The underlying concealed segmentation is denoted by z_i , where z describes the tissue type at voxel i . If there are 4 tissue types, then $z_i = e_k$ for some integer value of k in the range $1 \leq k \leq 4$, and where e_k is a unit vector whose k th element is 1, with the remainder of its elements being zero.
- Each tissue class, k , is modelled as a Gaussian distribution $G_{\sigma_k}(y_i - \mu_k)$, with a mean μ_k and standard deviation σ_k , and Φ_k refers to the parameter set $\{\mu_k, \sigma_k\}$. The mean and variances of the priors were used to initialise Φ for each class.
- The iterative part of the algorithm then begins. The expectation step updates the classification estimation, defining the current probability estimate that voxel i belongs to class k at iteration number $(m + 1)$

$$p_{ik}^{(m+1)} = \frac{f(y_i | z_i = e_k, \Phi_y^{(m)}) f(z_i = e_k)}{\sum_k f(y_i | z_i = e_k, \Phi_y^{(m)}) f(z_i = e_k)}. \quad (1)$$

Here the numerator represents the product of the Gaussian probability density function (PDF) and the prior likelihoods for each class (determined from the TPMs in this context). The denominator, representing a normalisation of the posterior probability, is the sum of all probabilities across all classes at each voxel.

- The maximisation step then involves updating the estimation of the parameter set Φ for each tissue class using the equations

$$\mu_k^{(m+1)} = \frac{\sum_i p_{ik}^{(m+1)} y_i}{\sum_i p_{ik}^{(m+1)}}, \quad (2)$$

and

$$\left(\sigma_k^{(m+1)}\right)^2 = \frac{\sum_i p_{ik}^{(m+1)} \left(y_i - \mu_k^{(m+1)}\right)^2}{\sum_i p_{ik}^{(m+1)}}. \quad (3)$$

- Convergence of the algorithm is then assessed according to the cost function

$$\frac{\partial \log L(\Phi)}{\partial \Phi} = 0. \quad (4)$$

The numerator here represents the sum of the logarithms of the overall probability density function for y_i across all classes, the maximisation of which is sought for successful segmentation. In reality, if this cost function was less than or equal to the specified tolerance, or if the number of iterations exceeded the specified maximum, then the algorithm was instructed to converge.

- After convergence, the class probabilities, p_{ik} , were reshaped from 4 1D vectors back to 4 3D matrices, in which the value at each voxel represents the probability of voxel i belonging to class k . A check was performed to ensure that the sum of probabilities at each voxel across all 4 classes was equal to one for each segmentation.

Estimation of the tissue segmentation (Equation 1) is therefore interleaved with estimations of the model parameters that maximise the likelihood of generation of complete data (Equations 2 and 3) in an iteratively performed two step process.

To motivate the use of anatomical priors, segmentation without them was performed. In this case Φ was initialised randomly for each tissue class, and $f(z_i = e_k)$ was initialised as a vector of ones. The algorithm converged in 11 seconds after 18 iterations. As can be seen from Figure 6, the 4 segmentation images produced do not correspond to the 4 desired tissue classes. This demonstrates the difficulty of segmenting and distinguishing each tissue type without having prior spatial information. Using prior information therefore fully automates the segmentation and avoids the need for manual initialisation. The use of prior information also makes the algorithm more robust in the face of severe bias fields¹ (see Section 2.4).

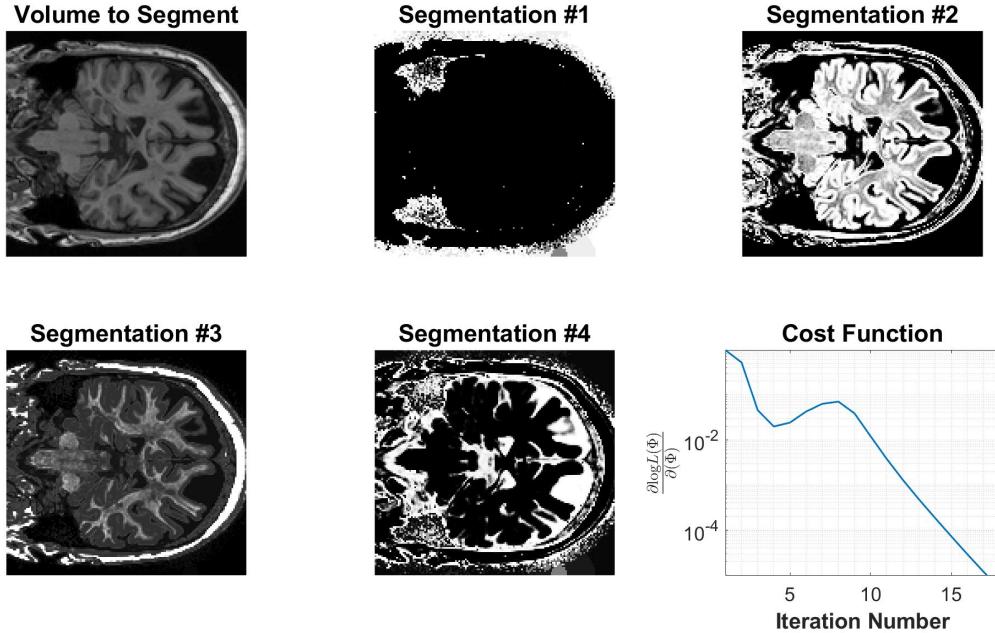


Figure 6: A visual summary of the segmentation of `img_9.nii`, without using *a priori* information.

2.3 Embedding a Markov Random Field

A Markov Random Field (MRF) was then embedded into the segmentation algorithm in order to introduce spatial smoothness and noise reduction. When considering K different tissue classes and 3D data, a first-order MRF takes into account the 6 nearest neighbours of each voxel. The probability that voxel i belongs to tissue class k is reduced if it is surrounded by voxels with a high probability of belonging to a class other than k . This process therefore facilitates the natural segmentation of the brain into different tissue types.

The MRF energy function, U_{MRF} , for voxel i and tissue class k can be calculated as

$$U_{\text{MRF}}(e_k | p_{\mathcal{N}_i}, \Phi_z) = \sum_{j=1}^K \left(\sum_{i=\mathcal{N}_i^G} p_{ij} G_{kj} \right), \quad (5)$$

where the matrix G , for $K = 4$, is the energy functional

$$G = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \quad (6)$$

and \mathcal{N}_i is the first-order neighbourhood of voxel i . It is important to note that this implementation assumes isotropic voxel dimensions, and does not consider the out-of-plane effects arising from anisotropic voxel dimensions, as described by Van Leemput *et al.*²

For the first iteration of the segmentation algorithm the MRF for each class was initialised as 1D vector of ones, which was then updated at each maximisation step. The calculation of the MRF at each voxel requires knowledge of the spatial relationship between voxels, and therefore the 1D p_{ik} vector was reshaped into a 4D matrix before the calculation of each update, and it was then reshaped back into a 1D vector after each update.

This MRF update calculation was implemented in the function `UMRF.m`, which generates a $3 \times 3 \times 3$ zeros matrix, with the 6 elements connected to the centre element converted to ones. This kernel representation of \mathcal{N}_i was then convolved with p_{ik} for all values of k apart from the value of k invoking the MRF update, before being summed across all classes. This represents an implicit (and more computationally efficient) application of the energy functional G .

The MRF update can be incorporated into the expectation part of the algorithm by multiplying the numerator of Equation 1 by

$$e^{-\beta U_{\text{MRF}}(e_k | p_{\mathcal{N}_i^{(m)}}, \Phi_z^{(m)})}, \quad (7)$$

where β is a regularisation parameter which controls the contribution of the MRF to class probabilities at each expectation step.

The implementation of an MRF added a significant computational cost to the segmentation process. This is demonstrated by setting $\beta = 0$, which effectively produced the same results as shown in Figure 5 and in the same number of iterations, but in a longer time of 15 seconds rather than 6 seconds. Using a value of $\beta = 0.5$ (Figure 7) produced segmentations that are noticeably smoothed compared to Figure 5, and the number of iterations required is also increased. Increasing this value to $\beta = 2.0$ (Figure 8) produced segmentations that are quite ‘blocky’ and perhaps over-smoothed, and the number of iterations required is also increased further.

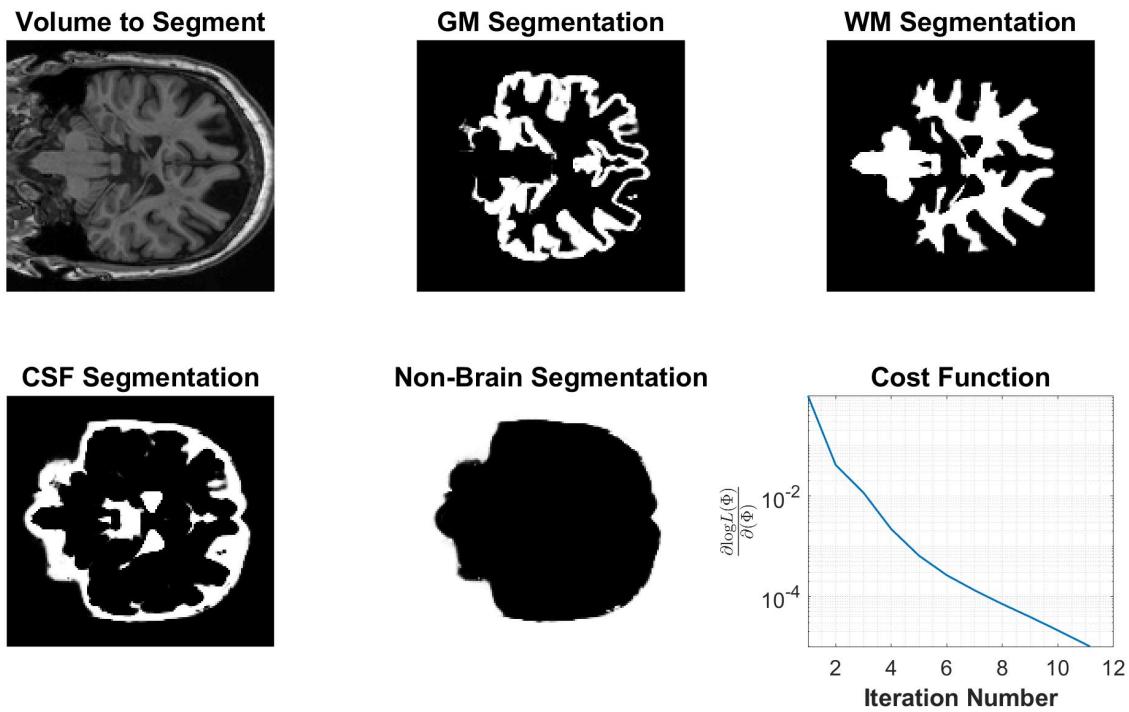


Figure 7: A visual summary of the segmentation of `img_9.nii`, using *a priori* information and embedding an MRF with $\beta = 0.5$ (Convergence achieved in 12 iterations and 22 seconds).

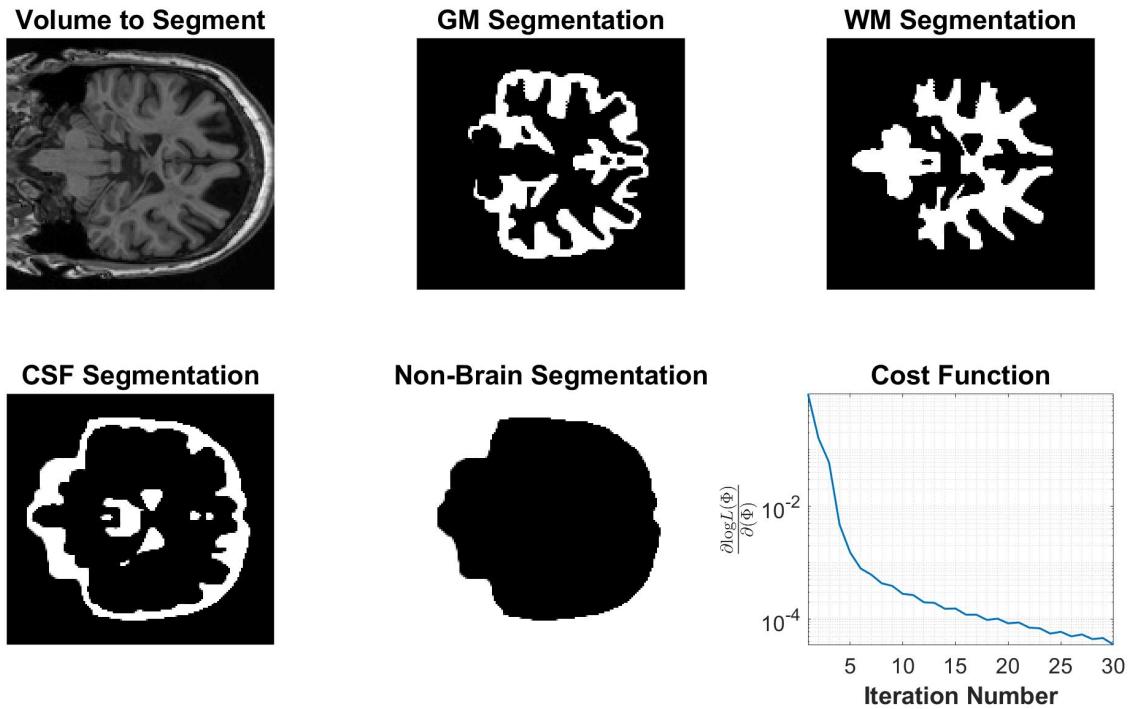


Figure 8: A visual summary of the segmentation of `img_9.nii`, using *a priori* information and embedding an MRF with $\beta = 2$ (Forced convergence after 30 iterations and 60 seconds).

2.4 Bias Field Correction

In order to correct for magnetic field intensity non-uniformity (INU), a bias field correction component was added to the probabilistic model used by the GMM segmentation algorithm. The bias field, BF , is modelled as a linear combination of smoothly varying polynomial basis functions of order N according to

$$BF = \sum_{j=1}^J c_j \phi_j(x_i), \quad (8)$$

where J represents the number of basis functions, x_i represents the spatial coordinates of voxel i , and $C = \{c_1, c_2, \dots, c_J\}$ denotes the bias field parameters. The basis functions are represented by the matrix A , which has a number of rows corresponding to the number of voxels in the image data, and a number of columns equal to J , where, for a 3D basis function,

$$J = \left(\frac{1+N}{1} \right) \times \left(\frac{2+N}{2} \right) \times \left(\frac{3+N}{3} \right). \quad (9)$$

The function `getMatrixA.m` receives an image's 3D dimensions (in terms of voxels) and a value of N (either 0, 1, 2, 3, or 4) and returns the appropriate matrix representation of A . A fourth order maximum was chosen as this was the order at which Van Leemput *et al.*¹ typically detected convergence in the accuracy of their segmentation algorithm. The same authors note that MRI bias fields are known to be multiplicative, and therefore the image intensities, y_i are log transformed at the start of the GMM algorithm in order to make the bias additive. The bias field is also initialised to be a vector of all zero values at this point.

The class probabilities used in the expectation step (Equation 1) are then modelled as

$$f(y_i | z_i = e_k, \Phi_y^{(m)}) = G_{\sigma_k^{(m)}}(y_i - \mu_k^{(m)} - BF^{(m)}). \quad (10)$$

The means and variances, as calculated in the maximisation step (Equations 2 and 3), are then modelled as

$$\mu_k^{(m+1)} = \frac{\sum_i p_{ik}^{(m+1)} (y_i - BF^{(m)})}{\sum_i p_{ik}^{(m+1)}}, \quad (11)$$

and

$$(\sigma_k^{(m+1)})^2 = \frac{\sum_i p_{ik}^{(m+1)} (y_i - \mu_k^{(m+1)} - BF^{(m)})^2}{\sum_i p_{ik}^{(m+1)}}. \quad (12)$$

The bias field parameters are then updated at each iteration according to

$$C^{(m+1)} = (A^T W^{(m+1)} A)^{-1} (A^T W^{(m+1)} R^{(m+1)}), \quad (13)$$

where W is a diagonal sparse matrix (implemented using MATLAB's inbuilt `spdiags` function) of the voxel weights of the residue image, w_i , according to

$$W^{(m+1)} = \text{diag}(w_i^{(m+1)}), \quad (14)$$

where

$$w_i^{(m+1)} = \sum_{k=1}^K w_{ik}^{(m+1)}, \quad (15)$$

and where

$$w_{ik}^{(m+1)} = \frac{p_{ik}^{(m+1)}}{(\sigma_k^{(m+1)})^2}. \quad (16)$$

The vector R is a residue image, and represents the difference between the image data, y_i , and the estimated bias corrected data, \tilde{y}_i (i.e. an estimation of the signal in the absence of a bias field), and is calculated as

$$R^{(m+1)} = y_i - \tilde{y}_i^{(m+1)}, \quad (17)$$

where

$$\tilde{y}_i^{(m+1)} = \frac{\sum_{k=1}^K w_{ik}^{(m+1)} \mu_k^{(m+1)}}{\sum_{k=1}^K w_{ik}^{(m+1)}}. \quad (18)$$

Equation 13 therefore represents a weighted least-squares fit of A to the residue image. This can be made explicitly clear by considering the bottom row of Figure 9, which shows the original image (y_i), the predicted signal in the absence of bias (\tilde{y}_i), the residue image (R), the weights of each of the voxels of this residue image (w_i), and the corrected image (i.e. $y_i - BF$). The top row of the same figure also demonstrates the spatial variation of the bias field for various differing orders of complexity. In a perfect model, the residue image would be equal to the bias field with some level of noise corruption.

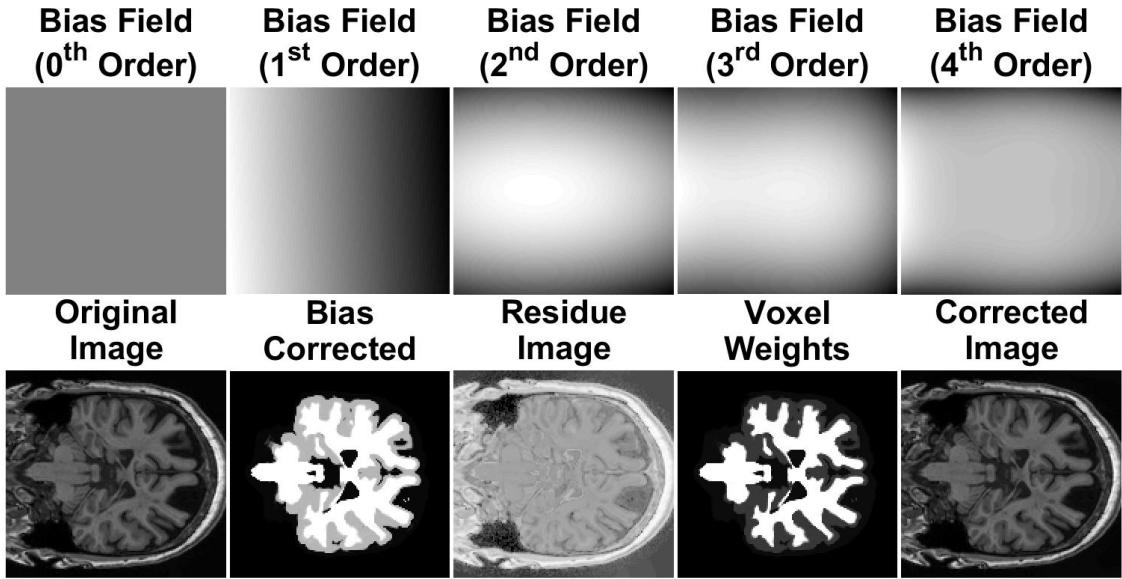


Figure 9: Bias field estimations at convergence for various values of N (top row). The remaining bias field parameters at convergence for $N = 4$ (bottom row). (All images refer to segmentation of `img_9.nii` and use $\beta = 0.5$; convergence achieved in 24 iterations and 170 seconds for $N = 4$).

Incorporating this bias field correction component added significant computational cost to the segmentation process. Table 1 describes the performance of the algorithm when segmenting `img_9.nii` with a value of $\beta = 0$.

Table 1: Effect of bias field order on computational performance.

Bias Field Order	Number of Iterations	Convergence Time (s)
0	8	24
1	8	30
2	22	90
3	22	108
4	19	140

2.5 Optimisation of Implementation Parameters

The groupwise TPMs derived in Section 1.2 were transformed into the space of `template_2_img.nii` (using the same process described in Section 2.1), before being used as priors to segment the same image. This was done iteratively for every possible pairwise combination of $\beta = [0.0:0.1:2.0]$ and $N = [0:1:4]$. At the end of each of these 105 test segmentations, the 4 probabilistic segmentations were converted into 1 categorical segmentation (V_{test}), which was then compared to the ground truth segmentation `template_2_seg.nii` (V_{gt}). This was done using a DICE overlap metric, according to

$$\text{DICE}^k = \frac{2V_{\text{test},\text{gt}}^k}{V_{\text{test}}^k + V_{\text{gt}}^k}, \quad (19)$$

where, for tissue class k , the numerator represents the number of overlapping voxels between the two images, and the denominator represents the sum of voxels in each image. This produced a score between 0 (no similarity at all) and 1 (perfect agreement) for CSF, GM and WM tissue classes. These three classes were then weighted by their relative abundance in V_{gt} in order to produce an overall score between the two images. These results are summarised in Figure 11 and Table 2. This whole process was performed within the script `GMM_EM_Optimise.m`, with a total run-time of 223 minutes.

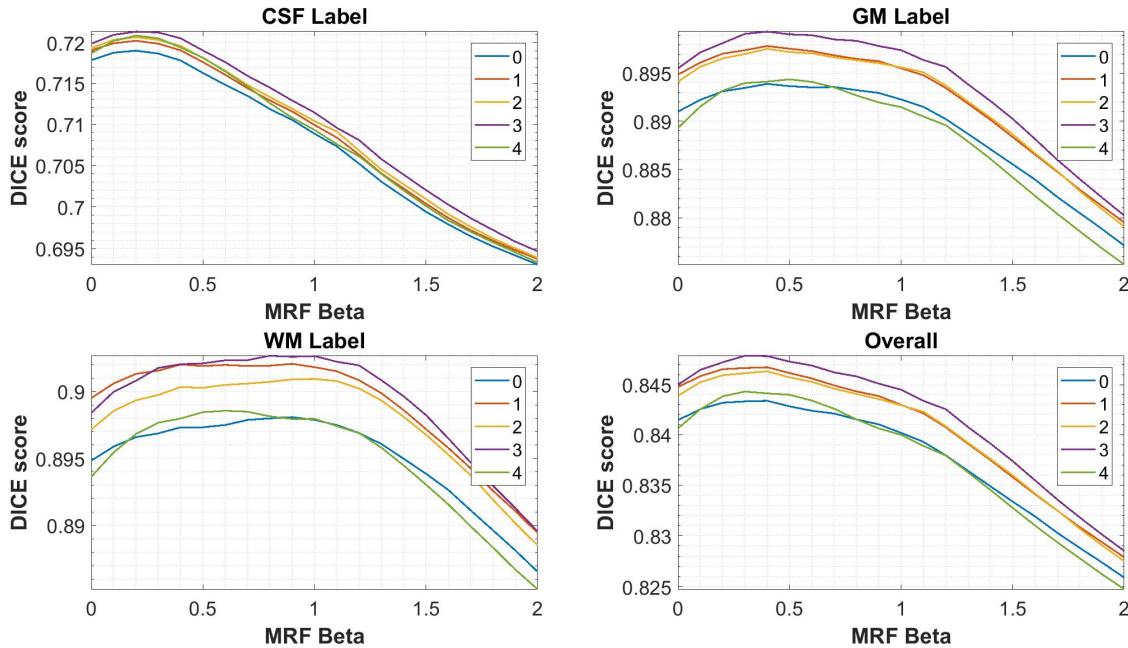


Figure 10: DICE scores for each tissue class and overall, for the range of β and N values tested. The legends indicate the order of the bias correction field.

Table 2: The values of β and N corresponding to the maximum DICE score obtained for each of the 3 tissue classes and a weighted overall score.

Tissue Label	Weighting Factor	Maximum DICE Score	β	N
CSF	0.2561	0.7214	0.2	3
GM	0.4110	0.8994	0.4	3
WM	0.3330	0.9027	0.8	3
Overall	1.0000	0.8479	0.3	3

The aim here is to maximise the *overall* similarity between a test segmentation and its ground truth counterpart, and therefore values of $\beta = 0.3$ and $N = 3$ were chosen as the optimal implementation parameters.

ters with which to segment the 20 provided non-segmented images. The results of this process, requiring 41 minutes to complete, are shown in Figure 11. However, if one wished to maximise the similarity of a *specific* tissue class, then a different β value could be used according to the results shown in Table 2.

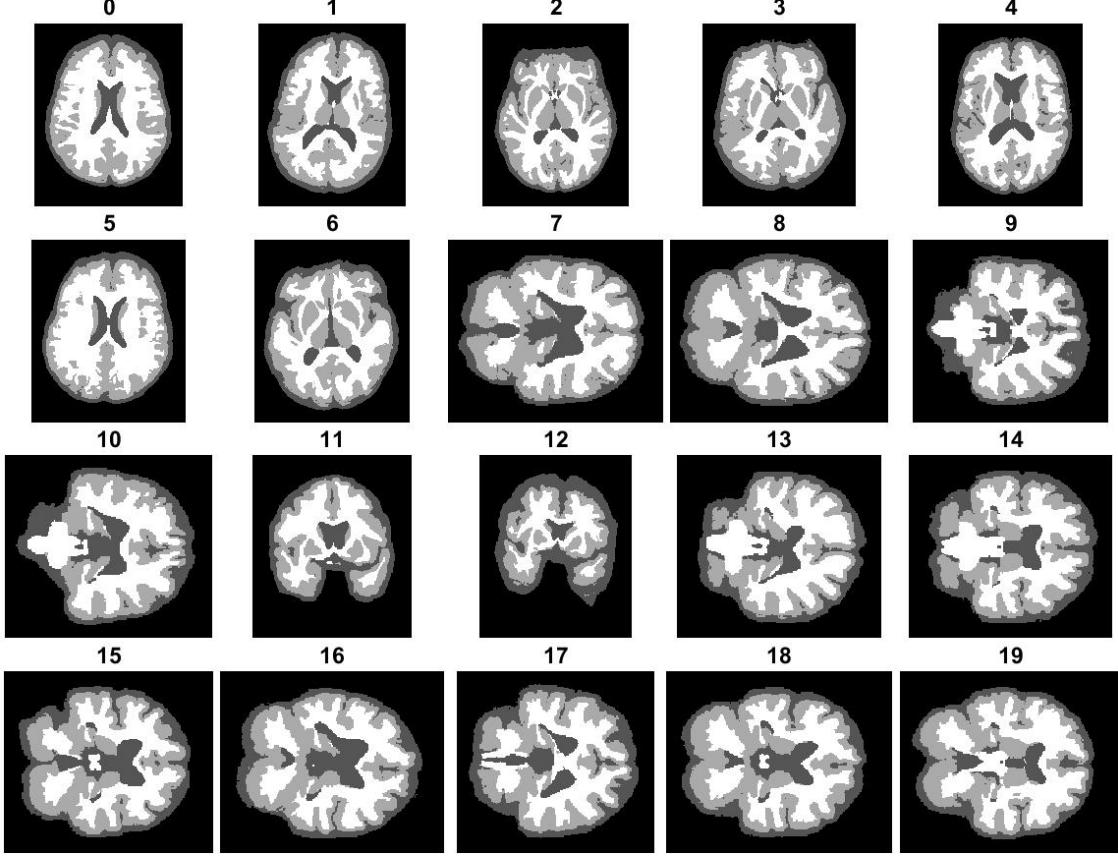


Figure 11: Segmentation of the 20 unseen images using optimised implementation parameters.

This process has selected parameters that optimise the segmentation `template_2_seg.nii`, and we have *assumed* that this optimisation generalises to the testing data-set of the 20 unseen non-segmented images. Given that the implementation parameters have only been optimised for *one* image, this is highly unlikely. However, this process could be made more robust by collecting similar data for the remaining 9 template images, and using a value of β and N that collectively optimises the segmentation of all 10 template images. Even then, there is no guarantee that these parameter values will translate well from the training data-set to the testing data-set. Indeed, we do not know if these two sets of images are drawn from the same population. There is also the additional computational cost to consider, as, given the run times encountered above, groupwise optimisation could require roughly 37 hours of additional continuous serial computer processing.

Step 3

3.1 Statistical Analysis

The script `StatAnalysis.m` loops through all of the above 20 segmented images and counts the number of CSF, GM and WM voxels in each image, as well as computing the volume of each voxel (in cm^3). The relationships between brain volume (i.e. GM and WM) and age, as well as between GM/WM ratio and age, were modelled as linear. This was based on the findings of Good *et al.*,³ who reported a linear reduction in GM volume with age, but no significant change in WM volume with age. All statistical analyses were carried out by the function `linearStatAnalysis.m`, which was written such that it receives two sets of data

and returns the two coefficients of a linear regression analysis, the Pearson correlation coefficient (r), and a p -value produced by a two-tailed t -test, using the test statistic

$$t^* = r \sqrt{\frac{n-2}{1-r^2}}, \quad (20)$$

where n is the number of observations. The p -value is then

$$p = 2 \times P(T < -|t^*|), \quad (21)$$

where T follows a t distribution with $(n-2)$ degrees of freedom. t statistics were used here due to the relatively small sample size. Please refer to the function `linearStatAnalysis.m` for precise details of how the linear regression and correlation coefficients were computed.

The results of the analysis of the relationship between age and brain volume, as well as between age and GM/WM ratio, are shown in Figures 12 and 13.

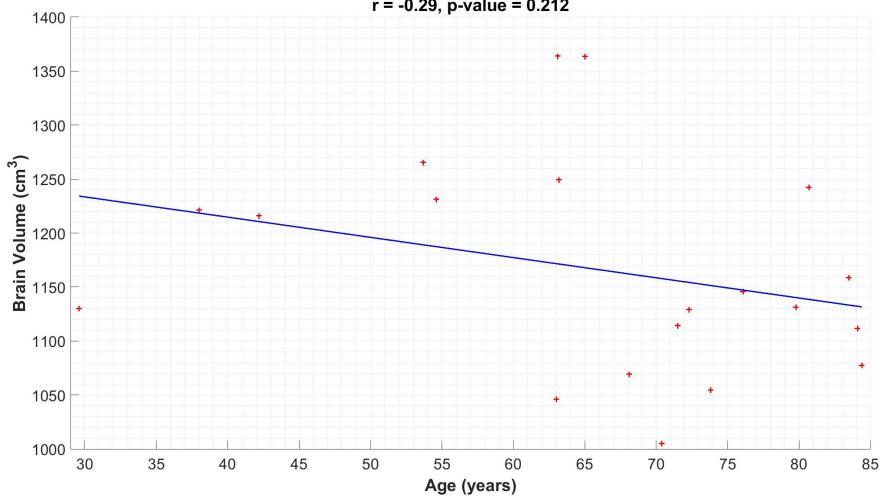


Figure 12: The relationship between brain volume and age.

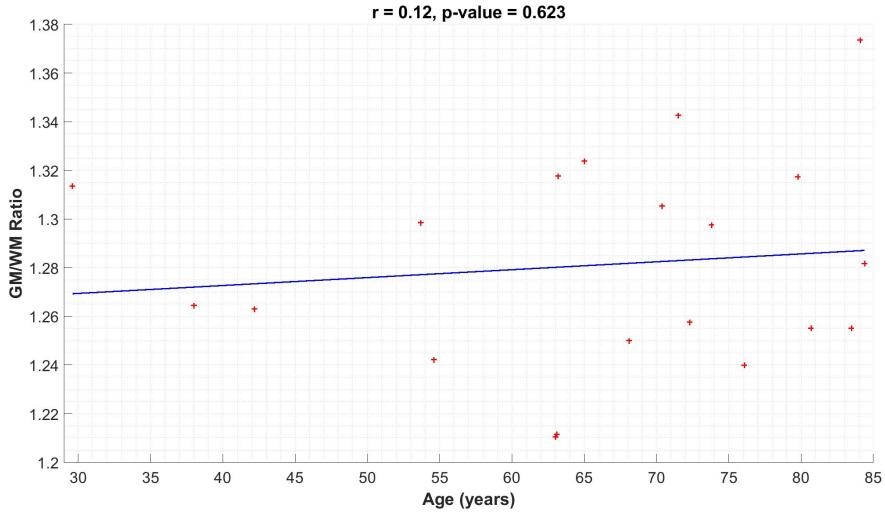


Figure 13: The relationship between GM/WM ratio and age.

Even though a negative correlation between brain volume and age is identified in Figure 12, the correlation is weak and the p -value exceeds a threshold 5% significance level. The relationship between GM/WM ratio

and age shows almost no linear correlation (r is close to 0), and has a clearly insignificant p -value. This is in keeping with the findings of Ge *et al.*,⁴ who found no significant effect of age on GM/WM ratio in 54 healthy subjects between the ages of 20 and 86.

In order to control for the variety in head size in the sample population, the brain volumes were then normalised by total intracranial volume (TIV) for each subject. As can be seen from Figure 14, the negative linear correlation is now much stronger, and the relationship is also statistically significant at a 5% threshold level of significance.

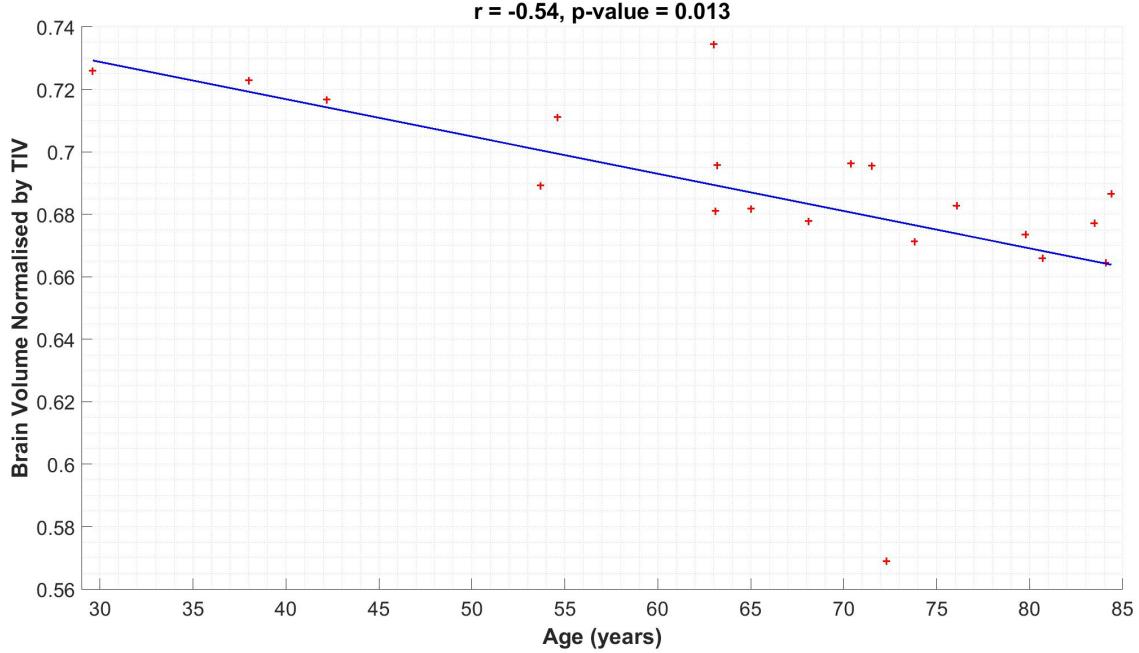


Figure 14: The relationship between brain volume (normalised by TIV) and age.

Therefore, it can be concluded from this analysis that there is a significant negative linear correlation between normalised brain volume and age in the test population. However, no significant linear correlation between GM/WM ratio and age was found in this population. The analysis is limited by the fact that the selection of segmentation implementation parameters is biased toward one sample. Also, where statistically significant relationships were not found, this study could have been made more powerful by increasing the sample size of the test population. Additionally, using non-linear models (e.g. quadratic) may have yielded stronger correlations and more significant results.⁵

References

- [1] Koen Van Leemput, Frederik Maes, Dirk Vandermeulen, and Paul Suetens. “Automated Model-Based Bias Field Correction of MR Images of the Brain”. In: *IEEE Transactions on Medical Imaging* 18.10 (1999), pp. 885–896.
- [2] Koen Van Leemput, Frederik Maes, Dirk Vandermeulen, and Paul Suetens. “Automated Model-Based Tissue Classification of MR Images of the Brain”. In: *IEEE Transactions on Medical Imaging* 18.10 (1999), pp. 897–908.
- [3] Catriona D. Good, Ingrid S. Johnsrude, John Ashburner, Richard N.A. Henson, Karl J. Friston, and Richard S.J. Frackowiak. “A voxel-based morphometric study of ageing in 465 normal adult human brains”. In: *NeuroImage* 14.1 I (2001), pp. 21–36.

- [4] Yulin Ge, Robert I Grossman, James S Babb, Marcie L Rabin, Lois J Mannon, and Dennis L Kolson. “Age-related total gray matter and white matter changes in normal adult brain. Part I: volumetric MR imaging analysis.” In: *AJNR. American journal of neuroradiology* 23.8 (2002), pp. 1327–1333.
- [5] Débora Terribilli, Maristela S. Schaufelberger, Fábio L S Duran, Marcus V. Zanetti, Pedro K. Curiati, Paulo R. Menezes, Márcia Scazufca, Edson Amaro, Cláudia C. Leite, and Geraldo F. Busatto. “Age-related gray matter volume changes in the brain during non-elderly adulthood”. In: *Neurobiology of Aging* 32.2 (2011), pp. 354–368.