

# Medical Image Segmentation and Applications final project: Brain MRI tissue segmentation

Kazi Saeed Alam, Gonzalo Esteban Mosquera Rojas  
saeed.utsha@gmail.com, gonzalo.mosquerarojas2000@gmail.com  
Erasmus Mundus Joint Master's Degree in Medical Imaging and Applications  
University of Girona  
Girona, Spain

## I. INTRODUCTION AND PROBLEM DEFINITION

Segmentation is one of the most important tasks in the field of Medical Imaging since it helps Computer-Aided Diagnosis (CAD hereinafter) systems to improve their performance by focusing on areas of interest depending on the medical application they have been built for. For instance, a CAD framework trained for classifying brain tumors would get more important and distinctive information in the region of the lesion (which requires segmentation), than from normal tissue areas. However, segmentation continues to be a challenging task since it is not yet possible to be done perfectly (which would be the ideal case), and there will always be an error due to several factors, e.g. noise and artifacts due to intrinsic uncertainty of the equipment used or errors during the image acquisition phase, etc. Nowadays, one of the most well-known methods for improving the results of segmentation in the medical imaging domain is by using prior information from the images through an atlas. As the name suggests, the atlas is a map that contains information of the anatomic structure of several individuals mapped into a single image space.

For this project, the subset of the IBSR (Internet Brain Segmentation Repository) database which is of brain magnetic resonance imaging (MRI) scans that have been manually segmented by experts has been used. The number of images used for training, validation and testing are 10, 5 and 3 respectively. The main goal of this project was to use segmentation techniques to segment the 3 different tissues: CSF, White Matter (WM), and Gray Matter (GM). First the each of the training images were registered with respect to each validation/test images and the labels from the training images were propagated according to the deformation field got after image registration. Then using these propagated labels various segmentation techniques like label fusion using majority voting, weighted majority voting, Non-local patch based label fusion and Expectation-Maximization Algorithm were used. The following sections explain the design and implementation of the tasks, as well as the algorithm analysis, results in analysis, and conclusions.

## II. DESIGN AND IMPLEMENTATION OF THE PROPOSED SOLUTION

The design part of the project consists of several tasks which will be discussed in the following sub-sections.

### A. Image Preprocessing

As the images given contained bias, noise and other artifacts. So before jumping directly into segmentation, the images needed to be preprocessed to get rid of these artifacts. The steps used for preprocessing can be shown in a block diagram in Figure 1.

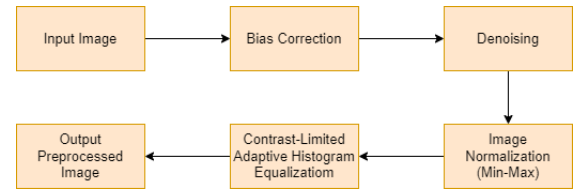


Fig. 1: Block Diagram for Image Preprocessing Steps.

1) *Bias Correction*: Bias correction in brain MRI images is a process that aims to remove intensity inhomogeneities from the image, which can be caused by various factors such as the magnetic field inhomogeneities and radio frequency (RF) imperfections. These intensity inhomogeneities can result in intensity variations in the image, which can affect the accuracy of image analysis and interpretation. Bias correction can be performed using various techniques, such as histogram matching, template matching, and field mapping. The choice of the bias correction technique depends on the specific characteristics of the MRI data and the desired level of accuracy. For this work, N4ITK (N4BiasFieldCorrection), a bias correction algorithm that is implemented in the open-source software package ITK (Insight Segmentation and Registration Toolkit) was chosen. N4ITK is based on the N4 algorithm, which is a variant of the nonparametric nonuniform intensity normalization (N3) algorithm. N4ITK uses a combination of intensity-based and spatial-based information to correct for intensity inhomogeneities in the image.

2) *Denoising*: As the images might contain some noise, the next step of preprocessing is to perform image denoising. The median filter is a nonlinear image denoising method that can be used to remove noise from images. It works by replacing the intensity value of each pixel with the median intensity value of a set of neighboring pixels. The size of this set of neighboring pixels is determined by the kernel size of the median filter. 'scipy.signal.medfilt' is a function that performs

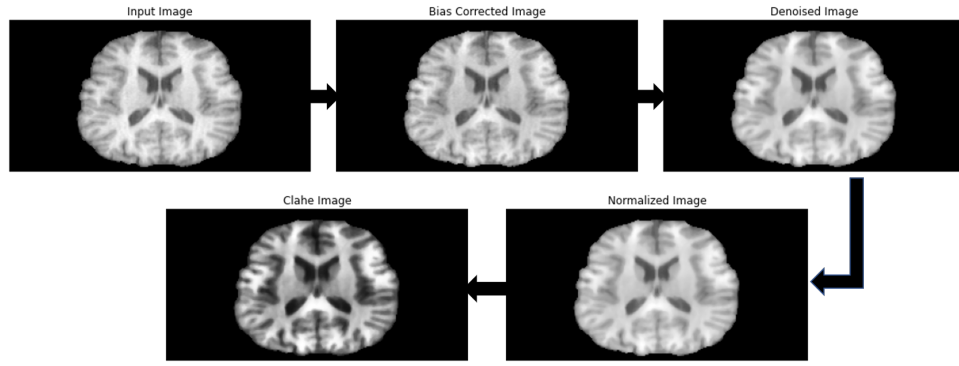


Fig. 2: Visualization of the steps in Image Preprocessing.

median filtering on an image which is also chosen for the project task. It works by replacing each element in the signal or image with the median value of a set of neighboring elements. The size of this set of neighboring elements was kept  $3 \times 3$ .

3) *Normalization*: After denoising the image the next step is to perform image normalization which is a process that scales the intensity values of an image to a specific range, such as  $[0, 255]$  for 8-bit images or  $[0, 1]$  for floating-point images. This is often done to improve the performance of image processing algorithms, as many of these algorithms are sensitive to the scale of the input intensity values. Image intensity values may vary widely across different images, even when these images are taken under similar conditions. Normalization helps to adjust for these variations, so that images can be compared and processed consistently. Image normalization is an important step in many image processing pipelines, as it helps to ensure that images are processed consistently and effectively for that reason it is also applied in this project.

4) *Histogram Equalization*: The final step of the pre-processing pipeline is to perform histogram equalization. To deal with the histogram equalization locally and for more improved results, CLAHE (Contrast-limited adaptive histogram equalization) method was used. It is a variant of histogram equalization that is designed to improve the contrast in an image while limiting the amplification of noise and other image artifacts. It works by dividing the image into small tiles, and applying histogram equalization to each tile independently. The contrast enhancement is limited by a user-specified "clip limit", which determines the maximum number of pixels that can be modified by the equalization process.

CLAHE can be useful for images with low contrast or with non-uniform illumination, as it can help to enhance the contrast in specific regions of the image without over-enhancing noise and other artifacts. 'equalize-adapthist' is a function which was chosen from the `skimage.exposure` module that performs contrast limited adaptive histogram equalization (CLAHE) on an image.

Finally the pipeline of image-preprocessing can be visualized with the following figure 2 which shows different stages

of pre-processing on an input image.

### B. Image Registration

Image registration is the process of aligning two or more images of the same scene taken at different times, from different viewpoints, or by different sensors. The goal of image registration is to determine the spatial transformation that aligns the images so that corresponding points in the images have the same coordinates. Image registration is used in a wide range of applications, including medical imaging, remote sensing, and computer vision. Here as the goal is to create atlas images, the first step for atlas creation is to register the images

For this task, 10 brain images and their corresponding 3D segmentation with three tissue levels (CSF, GM, WM) labels were given. The images were of different sizes and orientation. The first step for atlas creation is registering all the images with the fixed image. Here each of the validation or test set images was considered as the fixed image and all the training images were registered with respect to that.

A toolbox named *Elastix* [1] was used for the registration of images. It is open-source software, based on the well-known Insight Segmentation and Registration Toolkit (ITK). The software consists of a collection of algorithms that are commonly used to solve (medical) image registration problems for which it was also chosen in this project. To use *elastix*, 'sitk' (Simple Image Toolkit), a collection of open-source software tools for image analysis and processing, was used. It was used to perform image registration by importing the 'sitk' module and using the various image registration functions it provides. The registration was then executed, and the resulting registered image was written to a file. Both rigid and non-rigid parameters were used for registration. Brain 3D MRI Parameter file 0010 from Elastix Model Zoo was chosen as it offers affine and B-Spline information using mutual information as a similarity metric and can be used for interpatient image registration.

1) *Volume Resampling*: Volume resampling in registration refers to the process of resizing or resampling a volume image, typically as part of a process to align or register two or more images. This can be done for a variety of reasons, such as

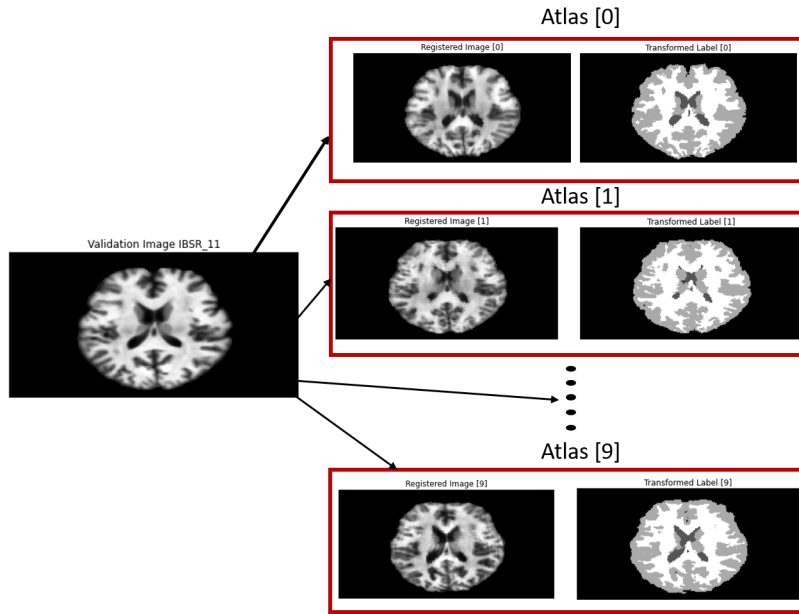


Fig. 3: Multi-Atlas Creation for IBSR11 Data.

to correct for differences in resolution or voxel size between the images, or to improve the accuracy of the registration. There are a number of methods that can be used to resample a volume, including interpolation methods like nearest neighbor, linear, or cubic interpolation. In this project, to improve the performance of image registration, resampling was done using ‘sitk’ module and Linear interpolation.

2) *Label Propagation*: To be able to build probabilistic atlas, the labels are needed to be registered also with respect to the label of the fixed image. Label propagation is a technique that can be used for image registration in cases where the images being registered have distinct, labeled regions of interest. The idea is to propagate the labels from the fixed image to the moving image, using the spatial transformation obtained from the image registration process.

So, after the registration process of all the images with respect to each of the validation images, the resulting image after registration as well as the deformation field that was used to transform the original image were generated. As there were two transforms (affine and elastic) that were used sequentially, the last transformation matrix hold the both transformation and can be used to transform the labels in the same way as the intensity images. ‘sitk’ module also provide the use of transformix and related functions.

3) *Multi-Atlas Creation*: The creation of a multi-atlas refers to the process of combining multiple atlases created after label propagation process discussed above into a single atlas that can be used for image segmentation. A multi-atlas is created by registering multiple atlases to a common reference space, and then combining them into a single atlas by averaging or fusing the segmentations or labels in some way.

Here after the label propagation process, each of the val-

idation images has a set of 10 registered images with their 10 transformed labeled images from the training data. This is the way we get multi atlas representation, an example for one validation image can be shown in figure 3.

### C. Image Segmentation

Now that multi atlas creation is done, the next step is to combining them into a single atlas by averaging or fusing the segmentations or labels in some way which will provide the segmented image. For segmentation or label fusion task, three techniques were used namely, majority voting, weighted majority voting and non local patch-base label fusion method. Each of the methods will be discussed briefly in the following sections.

## III. ALGORITHM ANALYSIS

### A. Majority Voting

The majority voting label fusion method [2] is a straightforward approach to combining the labels or segmentations from multiple atlases or sources into a single label map. It is based on the idea of selecting the most common label among the input sources as the final label for each voxel. This method can be applied to any number of input sources, and is particularly useful when the input sources are in good agreement.

To implement the majority voting label fusion method, each voxel in the final label map was initialized to the label that appears most frequently among the input sources. Among the ten atlases provide labels for a particular voxel, and five of them label the voxel as ‘CSF’ while two labels it as ‘background,’ and the rest other tissues the majority voting label fusion method will assign the label ‘CSF’ to the voxel in the final label map.

One advantage of the majority voting label fusion method is its simplicity. It requires minimal computation and can be implemented efficiently, making it well-suited for use in large-scale image processing tasks. It is also relatively robust to noise and outliers, as it relies on the majority of the input sources rather than any single source.

However, the majority voting label fusion method can be sensitive to the balance of labels among the input sources. If the input sources are highly variable or unreliable, the method may not perform well, as it can be swayed by a small number of outliers. In these cases, more advanced label fusion methods may be more appropriate. The process of weighted majority voting can be seen in Figure 4.

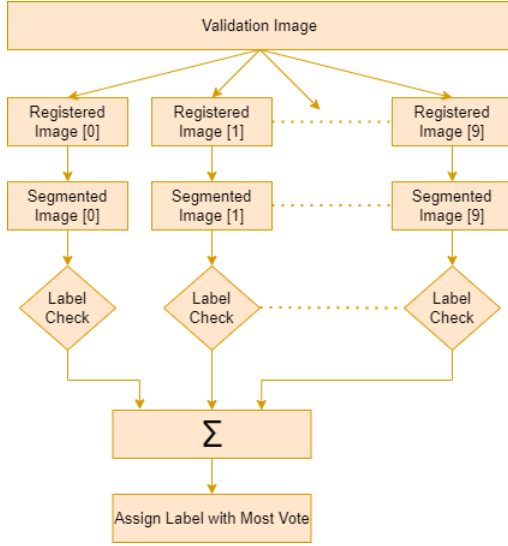


Fig. 4: Majority Voting Flow Chart.

### B. Weighted Majority Voting

Weighted majority voting [3] is a variant of the majority voting label fusion method that allows for the incorporation of weights or confidence values for each input source. This can be useful when the input sources are known to have different levels of reliability or quality, or when certain sources are more important than others for a particular application.

To implement weighted majority voting, each voxel in the final label map was initialized to the label that appears most frequently among the input sources, with each source's contribution to the vote being weighted by a user-defined confidence value. In this project, Structural Similarity Index Measure was used to give weights. It is a measure of the similarity between two images. It is based on the idea that the structural information in an image, such as the arrangement of edges and other features, is more important for perception than the exact pixel values. SSIM is calculated as a value between -1 and 1, where a value of 1 indicates a perfect match between the two images and a value of -1 indicates a complete mismatch. `structural_similarity()` function from the `skimage.measure` module was used to measure structural

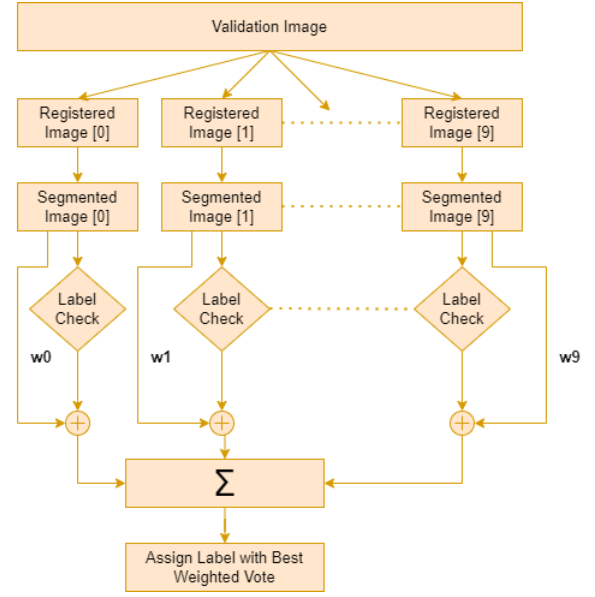


Fig. 5: Weighted Majority Voting Flow Chart.

similarity and takes two images as input and returns the value between -1 to 1.

For example, if ten atlases provide labels for a particular voxel, and five of them label the voxel as 'background' while the five others label it as 'CSF,' the weighted majority voting method will assign the label 'background' to the voxel in the final label map if the weights of the 'background' sources are greater than the weight of the "foreground" source.

One advantage of the weighted majority voting method is its flexibility. By allowing the user to specify confidence values for each input source, the method can be customized to reflect the relative importance or reliability of each source. This can be useful in cases where some input sources are known to be more reliable than others, or where certain sources are more relevant to the application at hand.

However, the weighted majority voting method can be sensitive to the balance of weights among the input sources. If the weights are not chosen carefully, the method may be swayed by a small number of sources with high weights, even if those sources are unreliable. It is also more computationally complex than the majority voting method, as it requires the computation of weights for each input source. The process of weighted majority voting can be seen in Figure 5.

### C. Non-local Patch-Based Label Fusion

The non-local patch-based label fusion method [4] is a technique for combining or merging the labels or annotations from multiple sources for a given dataset. This approach involves comparing patches or small regions within the images to find correspondences between the labels from different sources.

The method makes use of non-local means, which is a technique for denoising images by comparing patches within the image and replacing pixels with the average of the patch



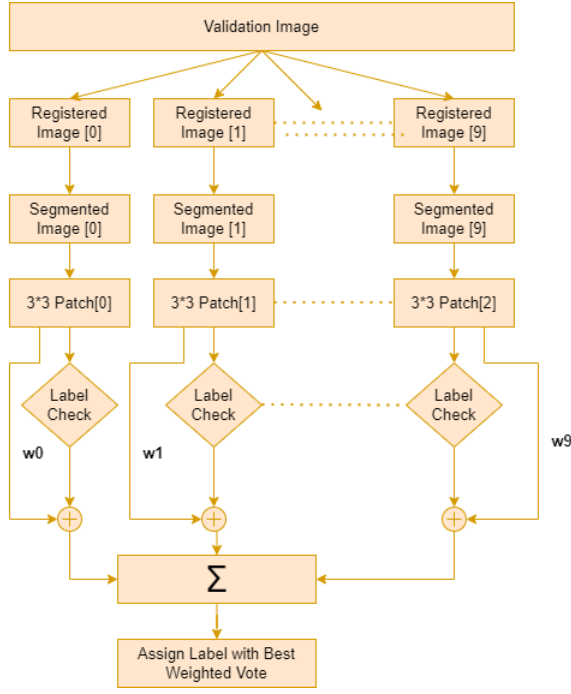


Fig. 6: Non-Local Patch-Based Label Fusion Flow Chart (For One Patch).

values. In the case of label fusion, non-local means is applied to the labels rather than the image itself, and the resulting label averages are used as the fused labels. This approach can be effective at reducing noise and improving the accuracy of the fused labels, especially when the labels from different sources are in good agreement with each other. Label fusion can be useful in a variety of scenarios, such as when multiple annotators are labeling the same dataset and the labels need to be combined, or when multiple models are trained to label a dataset and the predictions need to be combined. The non-local patch based label fusion method offers a way to effectively combine labels from multiple sources, helping to improve the overall accuracy and reliability of the resulting labels.

In non-local patch based weighting, small patches or regions within the image are compared to each other to compute weights or importance values for each pixel. In this task, MSE (Mean squared error) was measured between patches of size  $3 \times 3$  of each validation image and its corresponding registered images to assign the weights. These weights were then used to guide the segmentation process, for example by assigning higher weights to pixels that are more similar to other pixels within the same region. The process for a single patch can be seen in Figure 6.

#### D. Deep Learning approach: 2D UNET

Finally, in order to generate a thorough comparison of different segmentation methods, a Deep Learning approach was also adopted. This was done through the use of both a 2D UNET architecture.

The UNET is a Deep Convolutional Neural Network initially

proposed for the segmentation of biomedical images. It consists of a contracting and an expansion path that together form the shape of an U, and that is the reason behind its name. The first path is a Convolutional Neural Network with successive convolutions followed by a nonlinear activation function (ReLU) and max pooling operations [5]. It is called contracting because in each stage of the path the feature space gets reduced, since only the maximum response from every  $2 \times 2$  sliding window is preserved. The goal of the expansion path is to create a higher resolution segmentation map, and get back to the original size of the image. This is done through successive transposed convolutions, also known as upconvolutions. This upsampling procedure makes the feature space bigger in every level of the path. In each stage of the expansion path the upsampled representations are concatenated with the high resolution features from its corresponding level in the contracting path.

#### E. 2D UNET Architecture:

The architectures used for the 2D UNET is the following:

TABLE I: 2D UNET architecture

Layer	Number of Filters	Filter size	Activation
Conv2D	32	(3,3)	ReLU
MaxPooling 2D	-	-	-
Conv2D	64	(3,3)	ReLU
MaxPooling 2D	-	-	-
Conv2D	128	(3,3)	ReLU
MaxPooling 2	-	-	-
Conv2D	256	(3,3)	ReLU
UpSampling2D	-	-	-
Concatenate	-	-	-
Conv2D	128	(3,3)	Relu
UpSampling2D	-	-	-
Concatenate	-	-	-
Conv2D	64	(3,3)	ReLU
UpSampling2D	-	-	-
Concatenate	-	-	-
Conv2D	32	(3,3)	Relu
Conv2D	4	(1,1)	Softmax

As seen in the II, this is a simple 2d UNET with ReLU activation function and a final layer with softmax activation to give the final prediction for the 4 different classes to be segmented. Other important parameters taken into account for the training of the UNET were the following:

- Patch size = (64,64)
- Patch stride = (32,32)
- Batch size = 64
- Patience (stop learning criterion): 10
- Optimizer: Stochastic Gradient Descent
- Loss: Categorical cross entropy
- Epochs for training : 200 (reached 153 with early stopping).

#### IV. EXPERIMENTAL SECTION AND RESULT ANALYSIS

As for the experiments, after creating multi-atlas, three label fusion techniques were used to combine the multi atlases,

namely: Majority Voting, Weighted Majority Voting, Non-Local Patch-Based Label Fusion method. Whereas, the results obtained for the first two methods had very subtle differences and for the later, slightly low. The following sections present both the quantitative and qualitative results regarding the performance of each of the segmentation methods tried in this work. As a comparison metric Dice Score was chosen.

#### A. Tabular Analysis

- Table II, presenting the dice score results for the Majority Voting method.
- Table V, presenting the dice score results for the Weighted Majority Voting method.
- Table IV, presenting the dice score results for the Non-Local Patch-Based method.
- Table VI, presenting the Mean dice Score for each one of the methods used.
- Table VII, presenting the standard deviation of the dice scores for each one of the methods used.

1) *Comparison of multi-atlas based methods:* For the Patient IBSR\_11, the best-performing algorithm is Non-Local Patch-Based Label Fusion method as for the dice score for all the tissues, For the Patient IBSR\_12 and IBSR\_13, the best-performing algorithm is Non-Local Patch-Based Label Fusion method as for the dice score for all the tissues, Majority and Weighted-Majority Voting Performed quite well but the performance for non-local was lower. For the rest of the two patients, Non-Local Method outperformed the other two methods.

If we also see the mean and standard deviation per each tissue in the model, it shows that the majority and weighted-majority voting model's performance was better than non-local although the results for non-local outperformed the other two methods in the case of three images out of five. But for image 2 and image 3, the performance of the non-local method was very low compared to the other two methods which affect the overall performance. This might be due to poor registration as the creation of multi-atlas is dependent on image registration. As we explored only one parameter from elastix zoo (Par0010) which might be the reason. There can be other parameters that would have better registration and might improve the results. So, according to this task, the majority voting model was the best-performing label fusion method. Also, the computation time for majority voting, weighted majority voting and non-local methods were 15 minutes, 20 minutes and 1 hour respectively for segmenting each of the five validation images. So according to computation expense wise, majority voting was the simpler one.

2) *Comparison taking into account all methods:* When taking into account the performance of all methods, it can be easily seen in table VI that 2D UNET greatly outperforms all the rest of multi-atlas based methods for all tissues. The differences are bigger for grey matter and white matter. Therefore, this was the approach selected for generating the final segmentation masks for the challenge.

TABLE II: Dice score per tissue for each of the validation images with multi-atlas majority voting segmentation

Patient ID	Dice CSF	Dice GM	Dice WM
IBSR_11	0,7758	0,7852	0,8238
IBSR_12	0,8143	0,7991	0,8292
IBSR_13	0,7601	0,7863	0,8715
IBSR_14	0,802	0,8058	0,867
IBSR_17	0,8636	0,7765	0,8702

TABLE III: Dice score per tissue for each of the validation images with multi-atlas weighted voting segmentation

Patient ID	Dice CSF	Dice GM	Dice WM
IBSR_11	0,7854	0,7994	0,8244
IBSR_12	0,799	0,8001	0,8239
IBSR_13	0,7594	0,7798	0,8627
IBSR_14	0,8069	0,8091	0,8643
IBSR_17	0,8514	0,7798	0,8694

TABLE IV: Dice score per tissue for each of the validation images with patch-based segmentation method.

Patient ID	Dice CSF	Dice GM	Dice WM
IBSR_11	0,8257	0,8774	0,8633
IBSR_12	0,6528	0,7054	0,7568
IBSR_13	0,6307	0,7123	0,7933
IBSR_14	0,8602	0,867	0,8927
IBSR_17	0,8826	0,8325	0,8923

TABLE V: Dice score per tissue for each of the validation images with 2D UNET

Patient ID	Dice CSF	Dice GM	Dice WM
IBSR_11	0,8028	0,9201	0,9067
IBSR_12	0,8497	0,9048	0,9009
IBSR_13	0,7704	0,8703	0,9102
IBSR_14	0,8438	0,9109	0,9249
IBSR_17	0,8566	0,8625	0,9135

TABLE VI: Mean dice score per tissue

Segmentation method	Mean dice score per tissue		
	CSF	GM	WM
Majority voting	0,80316	0,79058	0,85234
Weighted voting	0,80042	0,79364	0,84894
Patch-based	0,7704	0,79892	0,83968
2D-UNET	0,8247	0,8937	0,9112

TABLE VII: Standard deviation of dice score per tissue

Segmentation method	Standard deviation		
	CSF	GM	WM
Majority	0,035724	0,010486	0,021218
Weighted	0,03017	0,011807	0,020362
Patch-based	0,106824	0,075061	0,05506
2D-UNET	0,033	0,023	0,008

#### B. Image Analysis

1) *Comparison of multi-atlas based methods::* From Figure 8, we can see the tissue segmentation for each of the three label fusion segmentation methods compared to the ground truth. For example, Patient IBSR\_11 was chosen as a candidate for analysis. From top to bottom:

- Image 1, presenting the ground truth labels for each tissue for patient IBSR\_11,

- Image 2, presenting the majority voting assigned labels for each tissue for patient IBSR\_11,
- Image 3, presenting the weighted majority voting assigned labels for each tissue for patient IBSR\_11,
- Image 4, presenting the non-local patch-based model assigned labels for each tissue for patient IBSR\_11,

As we know that, for patient IBSR\_11, the best-performing model was the non-local method, we can analyze each tissue for that. It can be clearly seen, that for majority and weighted-majority models, there are some clear discontinuation in the labeled images, which was more continued and smoother for non-local. This is also true for Grey matter (GM) and White Matter (WM) as the label images for non-local was more closer to the ground truth than the other two methods. So, the tabular analysis discussed in the section above agrees with the image visualization analysis. As, this is just for one single patient, it can be similarly shown for all the rest of the images. For Patient IBSR\_12 and IBSR\_13, it will be seen that the labels assigned by majority voting and weighted majority voting will be more accurate than non-local method.

2) *Comparison taking into account all methods*:: When taking into account all methods (multi-atlas based one and Deep Learning one), it is worth analyzing figure 7. In figure 7a, it can be seen, for instance, that the CSF dice scores are distributed between 0.78-0.87, 0.79-0.85, 0.88-0.92 for majority voting, weighted voting, patch-based and 2-D U-NET, respectively. This leads to the conclusion that patch-based method is the least robust one, since it is not as consistent on the predictions along all validation patients as the other methods, and therefore the distribution of the dice score values is broader. This behavior repeats for grey matter and white matter, and it can be seen in images 7b and 7c. Now, when analyzing images 7d and 7e it can be concluded once again that 2D-UNET was the best segmentation method, as it provides the highest mean score rate and smallest standard deviation for all of the tissues. Also, as discussed earlier, patch-based method is the not as robust since it has the highest standard deviation.

## V. CONCLUSIONS

Overall, Multi-atlas based segmentation is a method for automating the process of segmenting medical images, such as MRIs or CT scans. In this method, multiple “atlas” images, which are pre-segmented by experts, are registered to the target image and used to generate a prediction for the segmentation of the target image. The final segmentation is then obtained by fusing the predictions from all the atlases using a label fusion technique, such as majority voting or weighted majority voting.

- Majority voting involves simply taking the most common prediction among the classifiers. This method is simple to implement and can be effective in some cases, but it can also be prone to errors if the classifiers make mistakes that cancel each other out.
- Weighted majority voting involves assigning weights to each classifier based on their performance and taking the

weighted average of their predictions. This method can be more accurate than majority voting, but it requires a way to measure the performance of each classifier.

- Non-local patch-based methods involve comparing the predictions of each classifier to the predictions of their neighbors and using this information to make a final prediction. This method can be more accurate than either majority voting or weighted majority voting, but it can also be more computationally expensive.

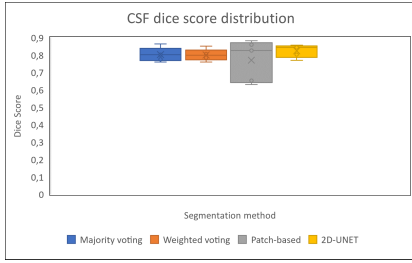
Overall, the choice of label fusion method will depend on the specific requirements and constraints of the problem at hand. In some cases, a simple method like majority voting may be sufficient, while in others, a more sophisticated approach like non-local patch based methods may be necessary.

When comparing the performance of multi-atlas based methods, it could be noted that no single method was the best for all five cases. However, when looking at average results, majority voting performed better. Patch-based method was the less consistent one as it was seen that the distribution of dice scores had a high standard deviation. This problem might be dependent on the patch of the size. Therefore, that should be a finely-tuned parameter. However, this method is too time consuming and computationally expensive, and in this particular task its efficacy was not seen when compared with simpler methods.

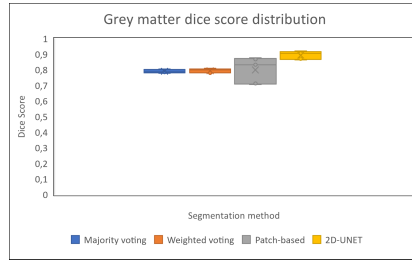
Finally, when taking into account all the methods (both multi-atlas based and deep learning), it was clear that 2D UNET was the best segmentation approach, even though it was a very basic network the improvement on the segmentation for tissues such as white matter and grey matter was noticeable. This once again confirms the promising performance of Deep Learning methods in medical imaging segmentation. If the network was more specialized and a more in-depth hyperparameter grid search had been done, the results would have even been much more better.

## REFERENCES

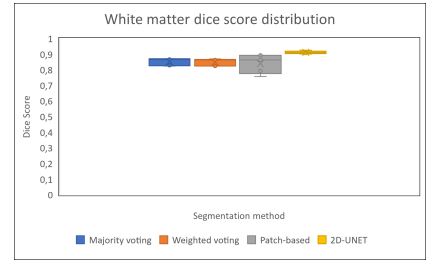
- [1] M. K. V. M. P. J. Klein S, Staring M, “elastix: a toolbox for intensity-based medical image registration.” 2010.
- [2] D. S. P. J. C. C. Y. P. Wang H, Suh JW, “Multi-atlas segmentation with joint label fusion.” in *IEEE Trans Pattern Anal Mach Intell.* 2013. IEEE, 2007, pp. 1–8.
- [3] V. S. W. L. Z. R. L. X. González-Villà S, Oliver A, “A review on brain structures segmentation in magnetic resonance imaging.” in *Artif Intell Med.* 2016, 2016, pp. 45–69.
- [4] V. S. Fonov, P. Coupé, S. F. Eskildsen, J. V. Manjón, and D. L. Collins, “Multi-atlas labeling with population-specific template and non-local patch-based label fusion,” 2012.
- [5] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention.* Springer, 2015, pp. 234–241.



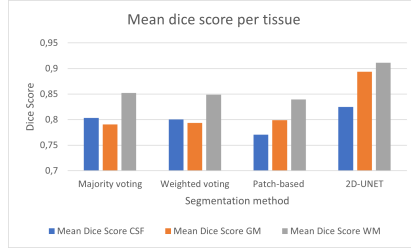
(a) Dice Scores Box Plots (CSF).



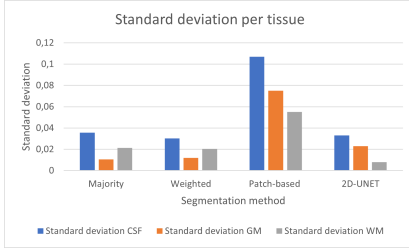
(b) Dice Scores Box Plots (GM).



(c) Dice Scores Box Plots (WM).



(d) Mean Dice Scores.



(e) Standard Deviation of Dice Scores.

Fig. 7: Result Analysis Plots

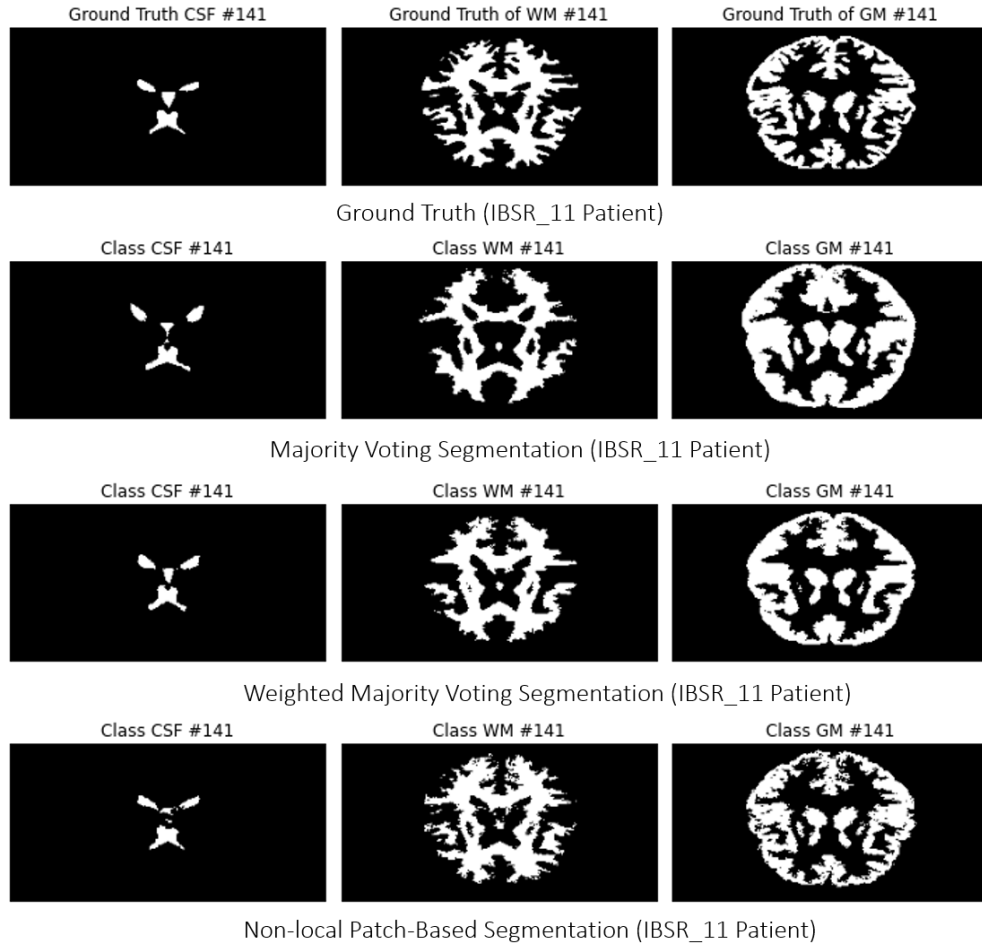


Fig. 8: Segmented Tissue Visualization.