Medical Image Segmentation and Applications
Brain Tissue Segmentation

Amparo S. Betancourt Tarifa, Muhammad Roshan Mughees

Master degree in Medical Imaging and Applications
University of Girona

Supervisors:
Xavier Lladó, PhD
Arnau Oliver, PhD
Sandra Gonzalez, PhD

***Abstract*— Accurate segmentation of brain tissues from magnetic resonance imaging (MRI) is significantly important in clinical applications and research. Accurate automatic brain tissue segmentation on clinically acquired MRI is a challenging task due to the presence of intensity inhomogeneity, noise, and the complex anatomical structure of interest.**

In this project two proposed segmentation methods are presented. First one consists of a multi-atlas approach and the second is a deep learning approach using a 3D U-Net.

The performance is evaluated on the Internet Brain Segmentation Repository (IBSR) 18 dataset. The proposed deep learning method achieves a mean dice similarity coefficient (DSC) of 0.920, 0.941 and 0.938 for the segmentation of cerebrospinal fluid (CSF), gray matter (GM) and white matter (WM) respectively and a mean DSC of 0.372, 0.856 and 0.813 for the segmentation of CSF, GM and WM with the multi-atlas approach on the validation data provided from the IBSR18 dataset.

I. INTRODUCTION

Magnetic resonance brain imaging has numerous advantages, including a complete analysis of the brain in various dimensions, high spatial resolution, and high contrast for different soft tissues.

Tissue segmentation and classification becomes fundamental steps in clinical diagnosis [1]. Classification or segmentation of brain tissue is used to detect and diagnose normal and pathological tissues such as multiple sclerosis (MS) tissue defects and tumors [2]. These defects could be recognized by monitoring changes in brain tissue quantity, size, and regional distribution during the patient check-up. Several neurological and psychiatric diseases such as Parkinson's, depression, autism, Alzheimer's, and Huntington's disease can also be diagnosed with modifications in the morphology of subcortical nuclei and cerebellum.

Three major classes of brain volume are gray matter, white matter, and cerebrospinal fluid. To extract these brain tissues, the brain tissue segmentation method is adopted.

When it comes to medical image segmentation, U-Net [3] is one of the most commonly used and best-performing architectures, as it performs state of the art, or near, in most applications.

It was developed in 2015, with the goal of performing of biomedical (microscopy) image segmentation. U-Net is a convolutional neural network that has two parts which are an encoder and decoder. The **encoder** compresses and extracts features from the image, arriving at the last layer having a single vector representing the image.

The **decoder** reconstructs the segmented image through a series of up-scaling layers. It also has concatenation layers which help in this reconstruction process.

The main feature of this architecture is that the concatenations that happen are on the same level i.e., between the encoder and the decoder, as it can be seen on figure 1.

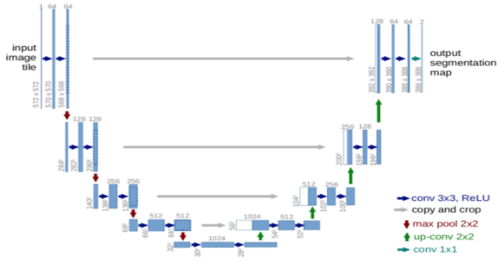

Fig. 1: U-Net architecture.

II. DATASET AND EVALUATION CRITERIA

A. *Dataset*

The MRI dataset used for this project, IBSR 18, consists of 18 real MRI images with different spatial resolutions, derived from healthy subjects [4].

For the development of our proposed segmentation methods, the original dataset was divided. Ten volumes were given for training, five for validation and three for testing.

All the images given for training and validation were provided with the ground truth tissue segmentation of WM, GM and CSF.

B. *Evaluation*

In this project, three metrics are used to evaluate the proposed method:

- DSC, which is defined by equation 1, for each tissue class (CSF, GM and WM).

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{1}$$

- Hausdorff distance (HD), defined by equation 2.

$$HD = max(h(A,B), h(B,A)) \tag{2}$$

Where h(A,B) is called the directed HD and it's given by equation 3.

$$h(A,B) = max_{a \in A} min_{b \in B} ||a - b|| \tag{3}$$

- Average volumetric difference (AVD), computed as shown in equation 4.

$$AVD(A,B) = \frac{100 \times ||A - B||}{\sum A} \tag{4}$$

III. ENVIRONMENT

The multi-atlas approach was developed using the dedicated IDE for Python, Spyder. For the registration process, ITKElastix was used, which provides an ITK Python interface to elastix. And for the pre-processing of the images simpleITK was used, along with OpenCV.
For the deep learning approach, we used python with the PyTorch framework. All the experiments were done on Google Colab with GPU and some testing was performed on our device with intel i5 and Nvidia GTX 1060 with (6 GB VRAM).
For visualization of the images, ITK SNAP was also used.

IV. PROPOSED METHODS

For this project we have proposed 3D U-Net and multi-atlas based approaches with pre-processing.

A. *Multi-atlas approach*

The proposed multi-atlas approach consists of a pre-processing step, a first affine registration to choose the K-best images using normalized cross correlation (NCC), presented on equation 5, as the selection criteria. Then, a second registration is performed using only the selected images (affine and B-spline) followed by a label propagation of the corresponding groundtruths on the image to segment and finally a label fusion of the previously obtained segmentations. Below a deeper explanation of each step is presented.

$$R(A,B) = \frac{\sum_{x=0}^{X}\sum_{y=0}^{Y}(A(x,y) - \overline{I_A})(B(x,y) - \overline{I_B})}{\sqrt{\sum_{x=0}^{X}\sum_{y=0}^{Y}(A(x,y) - \overline{I_A})^2(B(x,y) - \overline{I_B})^2}} \tag{5}$$

B. *Deep learning approach*

In this approach, the 3D volumes were used in the training and testing of a 3D U-Net.
A pre-processing step was performed doing a normalization of the images and before the training, the region of interest (ROI) in each image is found. This is done to remove the background.
The network was trained using patches of the volumes, that belong only to the ROI, with zero padding and using a multi loss approach that considers cross-entropy loss and Dice Coefficient loss.
Further details of this approach are explained below.

V. DESIGN AND IMPLEMENTATION OF THE PROPOSED METHODS

A. *Multi-atlas approach*

- **Pre-processing**.- The image to segment and all the training images go through this step, in which Open CV's bilateral filter is firstly applied to remove noise, followed by simple Itk's Adaptive Histogram equalization to enhance the image and finally the image is normalized in a range from 0 to 1. An example result of this step is shown in Figure 2.

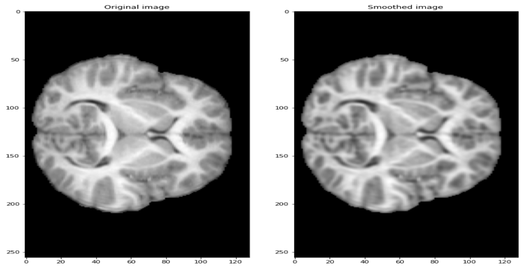

Fig. 2: Original image slice and Pre-processing result.

- **Affine registration**.- To put the volumes on the same voxel space, a basic registration is performed with the image to segment as the fixed image and the training images as moving.
- **K-best images selection**.- After the registration, the NCC between the image to segment and each training image is computed and stored. The selected best **K=3**, so the images with the three highest values of NCC are selected to eventually get the final segmentation, because this means they are the most similar to the image to segment.
- **Affine and B-spline registration**.-This second registration is performed with the selected K-best images and their corresponding groundtruths are also transformed, as shown in figure 3.

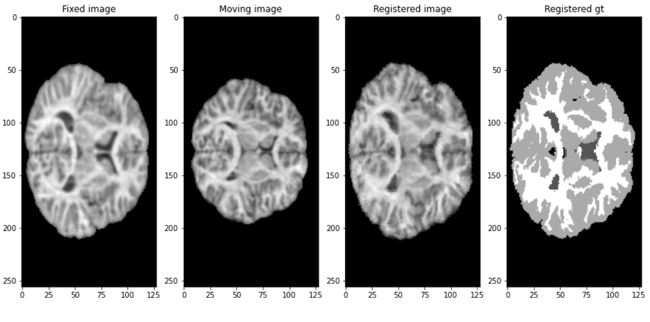

Fig. 3: IBSR 03 and respective groundtruth registered to IBSR 17.

- **Label propagation**.-Label propagation of the obtained transformed groundtruths is performed on the image to segment.
- **Label fusion**.- Finally the resulting segmentations are combined via majority voting obtaining the final segmentation result.

B. Deep learning approach:

- **Pre-processing**.-The dataset presented some difficulties since the image intensities varied significantly from image to image which caused bad results for CSF segmentation. Also, there was a lot of background in each volume and even some slices with just background. So we dealt with all these problems before feeding the data to our model.
  Firstly a **normalization** step is performed such that the images have 0 mean and 1 as standard deviation. If the image had negative values, they were summed up so that the minimum value becomes zero. The normalization was performed using equation 6.

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{6}$$

- **Region of Interest (General mask of the images)**.-Also before training, we computed the ROIs of the dataset (General mask of the images) so that we can use them to extract and remove the background and also the slices that just contain background. Here the tissue classes are not important, only that the image has a point of interest or not matters.
  Figure 4 presents an example of the obtained masks in volume IBSR 07 and IBSR 03 respectively.

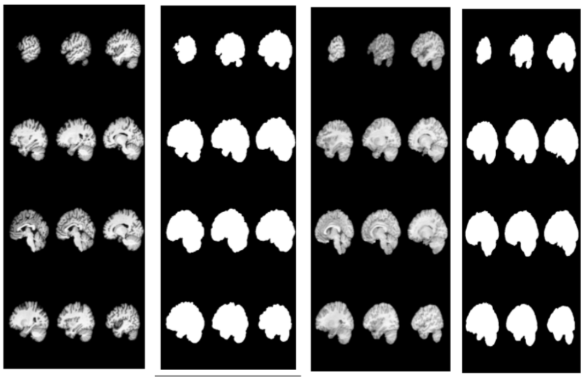

Fig. 4: ROI masks, IBSR 07 and IBSR 03

- **Useful patches extraction**.-In this step we basically divide the images into patches (*patch size*=32,32,32). The extra padding of zeros is added so that images become divisible into this patch size and all of them are loaded as input images and labels.
  A function was implemented to divide the images into patches.This function works in such a way that the returned patches are only those that contain some region of interest,so that we do not get an image with only background.
  This way we obtain the indices of the patches with useful information and at the end, we can apply some transformations to our new patches by loading them accordingly to their indices.
  On figure 5 we can see two sets of patches extracted from the training data, plotted with a batch size of 32.

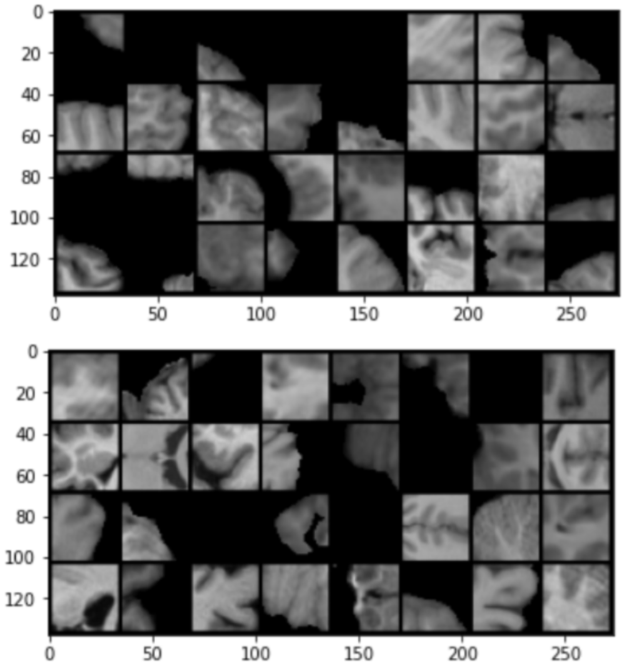

Fig. 5: Example of extracted useful patches from training data.

- **Loss function**.-As loss function, a new approach studied in the course was used. It is a multi-loss approach, that uses more than one loss function to compute the overall loss function. Cross-entropy loss and Dice Coefficient loss were used, and a threshold value (0.65) was set to give respective weightage to their values.

  Cross-Entropy Loss measures the performance of a classification model whose output is a probability value between 0 and 1. It increases as the predicted probability diverges from the actual label. The cross-entropy formula takes in two distributions, *y*, the true distribution, and *y'*, the estimated distribution, defined over the discrete variable and is given by the equation showned below (7).

$$L(\hat{y}, y) = -\sum_{k}^{K} y^{(k)} \log \hat{y}^{(k)} \tag{7}$$

  Dice Coefficient loss: also has a range from 0 to 1. It measures the overlap between predicted and actual. The equation corresponding to this metric is the one presented before (eq. 1).

The X intersection Y is actually element-wise multiplication between predicted and actual ground truth and then divide by the sum of both. We add a smoothing term in the denominator just for the case to prevent the term from getting divided by 0. Here, we calculate the loss for each class and add it to the total loss variable.
The loss function is to be minimized so we subtract it from 1 to get the maximum similarity between the actual and the predicted groundtruth.

- **Network**.-As mentioned before, we have very few examples for training a CNN (ten volumes), so batch normalization and dropout were added to avoid overfitting to the training data. The dropout rate was set to be 0.2.
The normalization technique used is instance normalization.
Overall in the encoder, every convolutional layer is followed by batch normalization, dropout, and then a pooling layer.
In the decoder, each layer is 3D transpose convolution and concatenations to respective layers from encoders.
- **Training process and implementation details**.-We set this model to train for 100 epochs but also implemented early stopping. Our criterion was if the validation loss did not decrease for 20 epochs then stop training. As it can be seen in figure 7 and 6, our model stopped training after 56 epochs. And also our learning rate was decreased if the validation loss didn't decrease (patience=5).

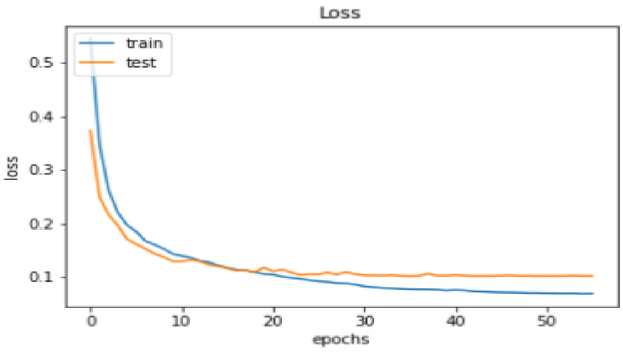

Fig. 6: Loss curve

We can see from the loss graph that overfitting started as the validation loss was stuck while the training loss continued to decrease. So our early stopping approach stopped it from training further.
Other implementation details are presented on table I.

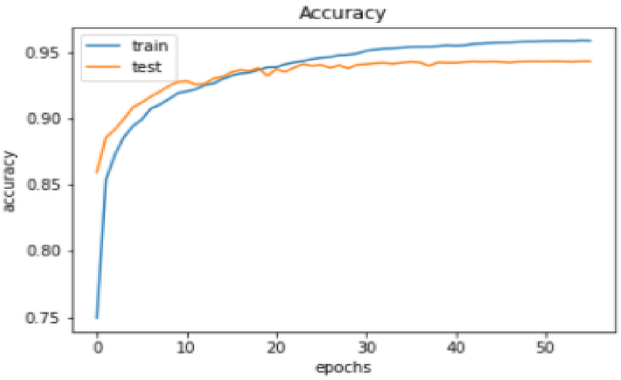

Fig. 7: Accuracy curve

TABLE I: Established parameters 3D U-Net

Parameter	Option
Optimizer	SGD, Adam
Patch size	16,32
Batch size	32,64,128
Dropout	None, 0.2, 0.5
Training Scheduler	None,10,20

VI. EXPERIMENTS AND RESULTS

To reach the final method used to obtain the presented segmentations on the test set, many trials were made to improve both approaches and finally choose the best one. Below we present a summary of the trials made.

A. *Multi-atlas approach*

On this approach, the modified aspects were mainly on the pre-processing, on the ways to perform the label fusion and on the K-best selection.

- Pre-processing.- On this step we tried applying different smoothing filters such as Median and Gaussian filter, but bilateral filter was chosen since it maintains the borders much better than the others.
Also to improve the contrast of the images, Open CV's Contrast Limited Adaptive Histogram Equalization (CLAHE) was used, but performance showed that simple Itk's worked better.
- Label fusion.- After our first implementation, we noticed the algorithm had problems with CSF segmentation, therefore we tried to think of ways to help the algorithm. Thinking that giving intensity information as weights on the label fussion step could help, we built the tissue models with the training images.This didn't help since, as it can be seen on figure 8, CSF intensity values are also shared with GM values, so the model couldn't solve this issue.
For label fusion, we also applied ITK's Simultaneous Truth and Performance Level Estimation (STAPLE), but overall majority voting gave better results after the K-best selection.

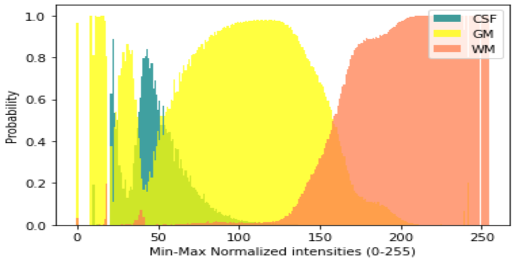

Fig. 8: Tissue models built with all training images.

- K-best selection.-After this attempt, we decided to implement the K-best selection with the sum of squared differences (SSD) and NCC. We observed the selected best images were the same with both metrics and decided to use NCC as selection criteria.
  Different values of K were tested but K=3 gave the best results with all validation images, but as mentioned before CSF segmentation results weren't good, therefore we chose to go further with the deep learning approach instead.

The final obtained results with this approach are presented in table II, and a slice of the segmentation result of IBSR 14 is presented on figure 9.

TABLE II: Multi-atlas results

Image	**DSC**			**HD**		
	CSF	GM	WM	CSF	GM	WM
IBSR 11	0.385	0.833	0.827	48.877	7.810	7.211
IBSR 12	0.422	0.835	0.819	48.847	7.000	6.481
IBSR 13	0.323	0.863	0.797	49.649	10.770	10.344
IBSR 14	0.292	0.874	0.824	50.100	7.483	8.602
IBSR 17	0.437	0.876	0.799	60.440	11.180	11.225
Mean	0.372	0.856	0.813	51.583	8.849	8.773

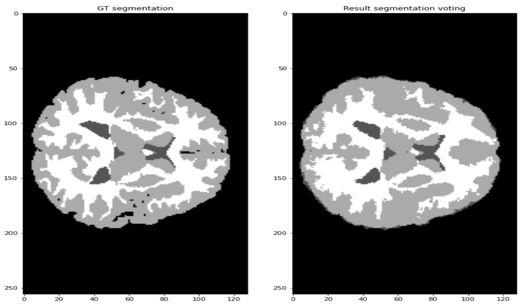

Fig. 9: **IBSR 14:**[left to right] Groundtruth segmentation and segmentation prediction by multiatlas approach.

B. Deep learning approach

- 2D approach.- For the 2D approach, we mainly tried a 2D U-Net on the complete data divided into slices.
  We split the dataset by the middle slice and normalized it.Then trained the network, but its results were not convincing so we decided to pursue the 3D approach.
- Transformations.-Different transformations were applied, such as rotation, scaling, ColorJitter, random crop, random affine, horizontal and vertical flips but it did not improve the performance of the model therefore were not used.
- Normalization technique.- Batch normalization was also applied but instance normalization achieved better results. Batch normalization normalizes for each channel across all the examples using mean and standard deviation. Whereas, in instance normalization, mean, and variance are calculated for each individual channel for each individual sample and was chosen for the final model.
- Patch size.- Three sizes were attempted. 32 proved to be better than 16 but we could not try the patch size of 64 (because of an error of CUDA running out of memory).

VII. Final Method and Results

As mentioned before, the deep learning approach was selected as the final method to provide the segmentation predictions of the test data, since the multi-atlas approach couldn't perform good on CSF segmentation.
Below we present the results obtained on the training and validation data, according to the three evaluation metrics previously explained.

Results on training data

On tables III, IV and V we can see the DSC, HD and AVD, respectively, for each tissue class on the volumes in the training data.
On figure 10, we present a slice of volume **IBSR 18** of the prediction obtained by the model and the groundtruth.

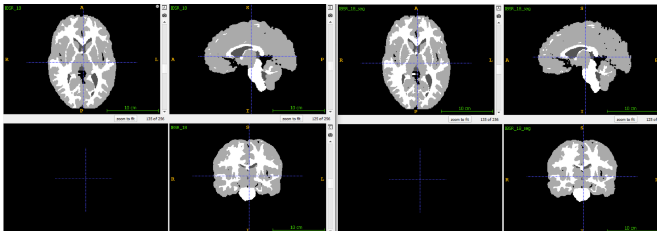

Fig. 10: **IBSR 18** [left to right] Segmentation prediction and groundtruth.

Results on validation data

On tables VI, VII and VIII also the DSC, HD and AVD, respectively, are presented for each tissue class on the volumes in the validation data.

On figure 11, we present a slice of volume **IBSR 14** of the prediction obtained by the model and the groundtruth.

Results on test data

On figures 12, 13 and 14 we present slices of the segmentation prediction corresponding to the three testing volumes.

TABLE III: Training DSC results

Image	DSC		
	CSF	GM	WM
IBSR 07	0.958	0.960	0.969
IBSR 06	0.966	0.950	0.957
IBSR 04	0.942	0.962	0.956
IBSR 05	0.954	0.954	0.955
IBSR 03	0.930	0.964	0.955
IBSR 01	0.963	0.966	0.965
IBSR 08	0.971	0.957	0.968
IBSR 16	0.947	0.971	0.966
IBSR 18	0.961	0.970	0.968
IBSR 09	0.968	0.954	0.962
Mean	0.956	0.960	0.962

TABLE IV: Training HD results

Image	HD		
	CSF	GM	WM
IBSR 07	11.090	6.0	7.071
IBSR 06	5.0	6.403	7.0
IBSR 04	6.633	5.831	8.246
IBSR 05	4.359	7.0	7.550
IBSR 03	5.477	8.062	8.602
IBSR 01	11.705	7.071	10.050
IBSR 08	5.744	8.307	5.477
IBSR 16	4.243	7.681	10.488
IBSR 18	14.353	7.280	10.488
IBSR 09	18.412	7.348	5.830
Mean	8.701	7.098	8.080

TABLE V: Training AVD results

Image	HD		
	CSF	GM	WM
IBSR 07	1.404	0.104	2.061
IBSR 06	0.291	2.564	1.355
IBSR 04	1.022	0.141	2.963
IBSR 05	0.907	3.686	3.357
IBSR 03	2.552	0.197	3.077
IBSR 01	2.652	1.093	1.622
IBSR 08	2.167	0.167	0.927
IBSR 16	1.154	0.725	1.825
IBSR 18	0.372	0.681	2.049
IBSR 09	1.477	0.769	2.290
Mean	1.400	1.013	2.153

TABLE VI: Validation DSC results

Image	DSC		
	CSF	GM	WM
IBSR 14	0.927	0.954	0.951
IBSR 11	0.897	0.937	0.955
IBSR 17	0.947	0.947	0.926
IBSR 13	0.899	0.938	0.918
IBSR 12	0.931	0.930	0.939
Mean	0.920	0.941	0.938

TABLE VII: Validation HD results

Image	HD		
	CSF	GM	WM
IBSR 14	4.242	7.550	6.708
IBSR 11	8.660	10.247	10.488
IBSR 17	20.396	12.884	9.695
IBSR 13	20.125	10.770	10.050
IBSR 12	7.810	6.708	8.307
Mean	12.247	9.632	9.050

TABLE VIII: Validation AVD results

Image	AVD		
	CSF	GM	WM
IBSR 14	8.426	1.298	0.857
IBSR 11	0.940	1.464	4.220
IBSR 17	1.525	1.772	6.452
IBSR 13	3.358	6.162	9.794
IBSR 12	3.489	1.664	1.531
Mean	3.548	2.472	4.571

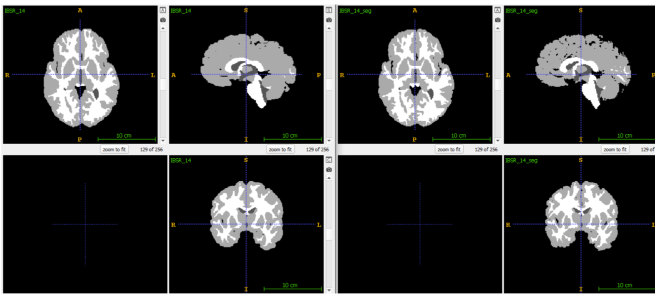

Fig. 11: **IBSR 14** [left to right] Segmentation prediction and groundtruth.

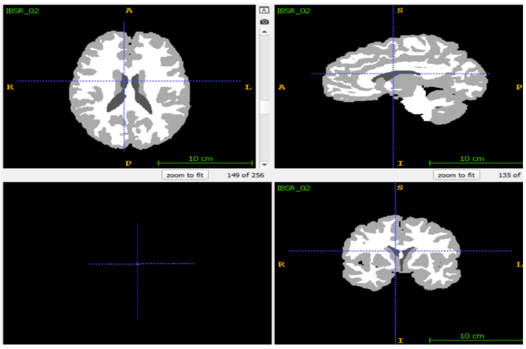

Fig. 12: **IBSR 2:** Segmentation prediction

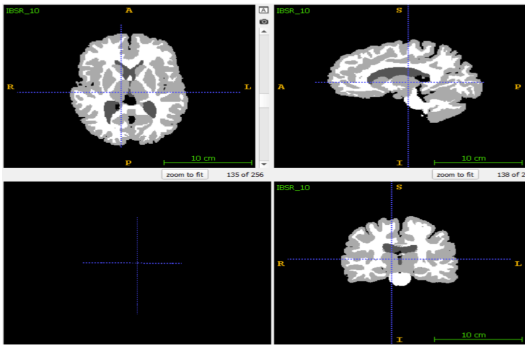

Fig. 13: **IBSR 10:** Segmentation prediction

The complete training of the proposed network took 65 minutes and 26 seconds on google Colab's GPU.

All the prediction volumes can be found on [5].

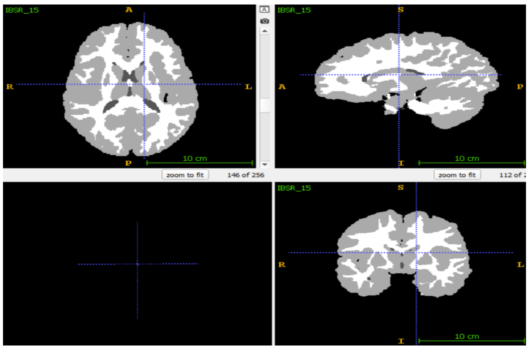

Fig. 14: **IBSR 15:** Segmentation prediction

VIII. PROJECT MANAGEMENT

We developed this project in a group of two, scheduling different meetings and giving different tasks both before and during winter holidays. We mainly used our devices to develop code and Google Colab to test our algorithms and shared our progress on Google Drive.
First we had a discussion with our lab supervisor and decided to try deep learning and non-deep learning approaches. As we progressed further, based on our results we decided to perform different experiments with both approaches and present our results in this report, which was jointly written using the online LaTeX editor overleaf.
On table IX we present an estimate of the time spent on each task performed to develop this project.

TABLE IX: Project management

Task	**Estimated time [hours]**
Literature review on multi-atlas and deep learning approaches for brain tissue segmentation.	4
Building of multi-atlas approach.	14
Design, training and experiments on deep learning approach.	36
Report writing	6
Approximate total time	60

IX. CONCLUSIONS

During this Project we learned how to deal with a segmentation problem through deep and non-deep learning approaches.
We saw the potential of non deep learning approaches, and understood there could be ways to add information to the model in order to improve the segmentation results, such as joint label fusion which we couldn't implement due to time constraints. On this case, we were also able to verify that the segmentation groundtruths of the dataset limited the achievable performance of this method.
We were able to understand, adapt and improve a given architecture to solve our task in a better way. In our work, the deep learning model works significantly better than the non deep-learning approach. Even with little pre-processing we were able to achieve good results, which shows the power of the U-Net architecture and deep learning in general if we provide the data in an efficient form (in our case as non-background patches).

REFERENCES

[1] Dora, L., Agrawal, S., Panda, R., Abraham, A. (2017). State-of-the-art methods for brain tissue segmentation: A review. IEEE reviews in biomedical engineering, 10, 235-249.

[2] Adhikari, S. K., Sing, J. K., Basu, D. K., Nasipuri, M. (2015). Conditional spatial fuzzy C-means clustering algorithm for segmentation of MRI images. Applied soft computing, 34, 758-769.

[3] Ronneberger, O., Fischer, P., Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[4] NITRC NeuroImaging Tools and Resources Collaboratory https://www.nitrc.org/projects/ibsr

[5] https://drive.google.com/drive/folders/1jIPOvH-7qtsHLBnFeSn5xJ3lSQUGtFJ7?usp=sharing