Patch-Based Segmentation of Brain MRI Images Using Deep Learning

Jhon M. Gomez, Esteban Vaca, Sergio Tascon
Medical Image Registration and Applications - MAIA
University of Girona

January 24, 2020

Abstract

This document contains the information corresponding to the patch-based segmentation of MRI images using convolutional neural networks (CNN). A pipeline is proposed for the segmentation of Cerebro Spinal Fluid (CSF), Gray Matter (GM) and White Matter (WM). Results are measured by means of the Dice coefficient (DSC), Average Volumetric Difference and Hausdorff distance. Best average DSCs were 0.920 ± 0.015, 0.943 ± 0.008 and 0.937 ± 0.015 for CSF, GM and GM, respectively.

1 Introduction

The rising of the CAD systems and their demonstrated effectiveness in the duty of assisting the specialist in the detection of diseases creates a propitious climate for exploring and innovate every day.
In what Bran MRI images concern, the artificial intelligence along with the revolutionary concept Convolutional Neural Networks (CNN) have acquired relevance assisting the tedious task of manually segmenting tissues, structures or delimiting lesions and analyzing the brain evolution over the time, especially in persons with neurodegenerative diseases. Therefore, computerized methods for segmentation are, nowadays, fundamental tools in the task of assisting specialists during the diagnosis [1].
Having this context, this project aims to explore different CNN architectures and customizing their parameters to reach an accurate brain tissue segmentation over Brain MRI T1 sequences.
Such architectures are the current state-of-the-art in segmentation, and their main goal is to perform segmentation in volumes (3D), based on patches rather than slices.

2 Materials

In this work, we consider the publicly available dataset: Internet Brain Segmentation Repository 18 (IBSR18) [2] which contains 18 T1w-MRI scans of adults. The dataset has been processed by a bias-field correction and a skull stripping algorithm. However, the data present uncorrected intensities among the different scans and has three different anisotropic voxel spacing, namely (0.837, 1.5, 0.837), (0.938, 1.5, 0.938) and (1.0, 1.5, 1.0).

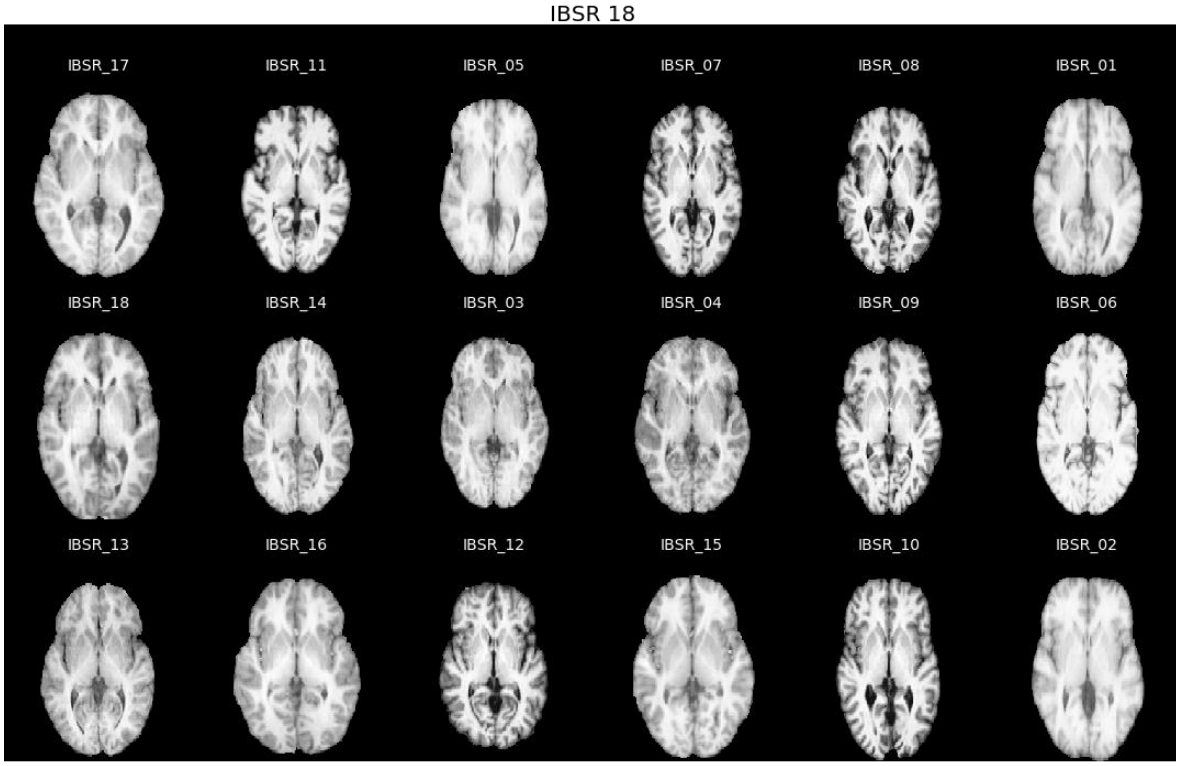


Figure 1: IBSR 18 Dataset

The architectures were deployed on python and tested in Google Colab notebooks, as they provide non-stop 12 hours processing virtual machines with dedicated graphics of NVidia Tesla, ideal for deep learning applications. We developed prototypes with the deep learning frameworks Pytorch and TF-Keras.

3 Methodology

To tackle the problem of brain tissue segmentation we used a deep learning approach based on 3D patches. Figure 3 shows the basic pipeline of our approach. Some preprocessing techniques were preliminarily explored. The literature points out that in-

tensity normalization is key for improving the network outcome, along with, CLAHE and histogram matching.
Once the prepossessing is done, 3D patches of a specific size are extracted, essentially from the foreground for which the ground truth is used as a segmentation mask. Patches are generated for both ground truth and training images so that the loss can be computed during training. The training set is manually divided between Training and Validation set, since the amount of available data.

Several architectures were tested with different configurations, and all of them with 3D patches. Particularly the 3D UNet, which was provided during one U-Net seminar, the VNet and the NeuroNet. Once the CNN has been trained for a proper amount of epochs, the performance is evaluated in the inference stage, in which images of the test set are passed through the model and a predicted segmentation is generated. Several metrics are computed to obtain quantitative measures. The following section explains the implementation of each stage in more detail.

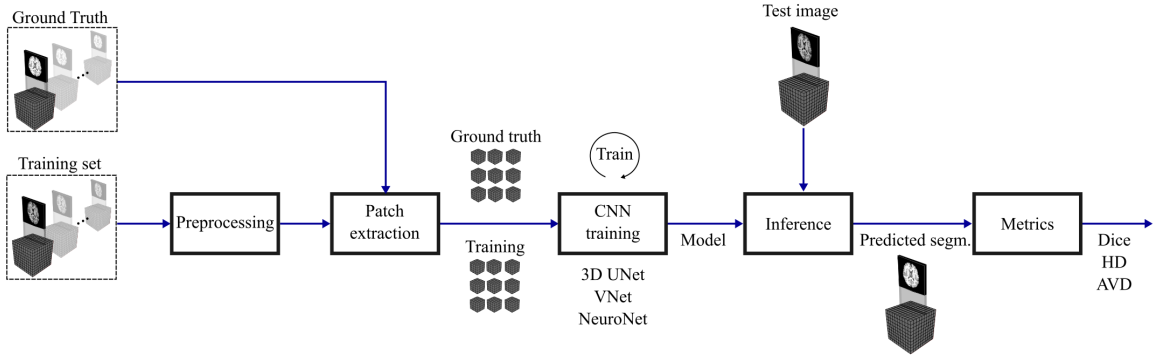


Figure 2: Pipeline for brain segmentation

4 Implementation

As mentioned before, we used 3 different CNN architectures for the segmentation of the brain tissues on MRI T1 images. All of them were used in a patch-based approach and implemented using Pytorch, except for NeuroNet which is based in TensorFlow.

Some of the steps of the approach were based on the scripts provided by Sergi Valverde during the seminar on the UNet. Particularly the patch generation algorithm and the image reconstruction from patches relied entirely on the mentioned scripts. Also, as it will be explained, the template of the UNet was also used and modified.

4.1 Pre-processing

We implemented several pre-processing methods aiming at improving the heterogeneity of the spatial and intensity distributions of the datasets.

4.1.1 Spacing normalization

We implemented spacing normalization using a resampler in SimpleITK, to have the same spacing for all images. This brought the need for also transforming the ground truth, which was performed with a nearest-neighbor interpolation to avoid new values.

4.1.2 Intensity normalization

Neural Networks trained on standardized data yield better results in general, as presented in [3]. In this work, we make use of Standardization for preprocessing all the images. However, when we analyzed the dataset carefully, the images have different intensities and proceed from different machines. Thus it is important to not only standardize the images but matches histograms of the images to make sure that all the samples belonging to a class are coherent among all of them.
We propose to perform histogram matching to all the images of the dataset with a reference image, to homogenize all the different available intensities of the dataset. We set the number of matching points of the histogram to seven.

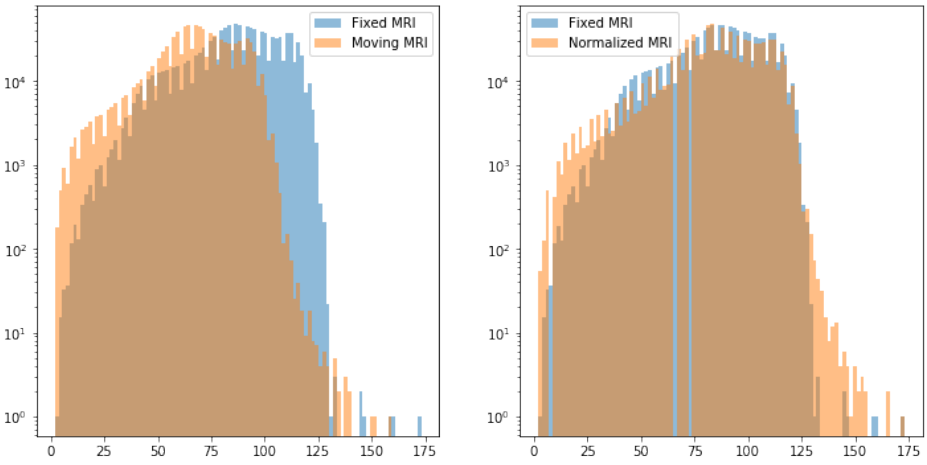


Figure 3: Effect of Histogram Matching

4.1.3 CLAHE

To improve the contrast of the images at a local level, we applied Clipped Local Adaptive Histogram Equalization (CLAHE) on the whole dataset using a tile grid size of

8x8 and a clip limit of 2.0. Because of the 2D property of the used implementation, we applied the CLAHE on every slice of the 3D volumes.

4.2 3D UNet

The provided 3D UNet was first adapted from the problem of lesion segmentation to the problem of brain segmentation, which consisted of modifying the number of output classes. For this problem, we needed to classify every voxel into one of the classes Background, CSF, Gray Matter, and White Matter. This architecture is a 3D extension of the original UNet [4], which was originally developed for processing 2D images. In this 3D version, there are several differences concerning the original version, such as the use of only one convolution in each level as opposed to the two convolutions used in the original version. Another difference is that the horizontal connections are implemented as summations in the 3D version, while in the original version concatenation is used.

After the architecture produced coherent results for the segmentation, we proceeded to modify it to improve the results. We tried to increase the depth of the network to have a higher number of feature maps. Another modification consisted of changing the initial number of feature maps from 32 to 64 so that a total of 512 feature maps could be reached in the latent space. This number of feature maps is twice the number of feature maps of the provided implementation, which would in principle allow the extraction of deeper features. Figure 4 shows this particular model.

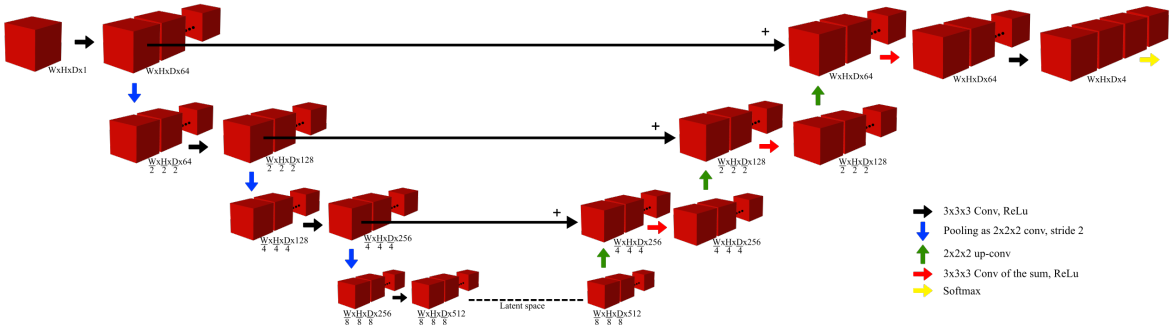


Figure 4: Implemented 3D UNet architecture with patches of size WxHxD

Another modification performed to the 3D U-net model was adding Instance-Layer Normalization to the convolutional layers, this normalization was implemented as presented in [5] extending it to a 3D model. Therefore, we considered a feature map of dimension (N, H, W, D, C), where N is the batch size, H is the feature height, W is the feature width, D is the feature deep and C is the feature channel, so we performed instance normalization to standardize the data along the dimensions (H, W, D), calculating the mean and the variance in the mentioned feature directions.

For training the architecture we implemented several strategies to overcome the data imbalance presented in the dataset. We re-sample 3 times the patches that contained CSF labels so that the model could specialize more in this label. Additionally, to overcome the overfitting problem presented when training a large number of epochs we included drop-out layers to the model, this allowed us to avoid overfitting and improve the quality of the training.

4.3 NeuroNet

NeuroNet [6] was conceived as a brain image segmentation tool based on a multi-output CNN architecture, trained to reproduce the output of multiple neuroimaging tools.
Authors point that one of its main characteristics is that the tool was specially designed to learn from raw images, avoiding the application of typical pre-processing techniques to get optimal results, thus reducing the need for adjusting the network's hyper-parameters.
As the challenge is limited to the segmentation of the brain structure in 3 different tissues (CSF, GM, and WM) the multi-output characteristic of the network is not exploded, in that way the single-output architecture is a proper 3D adaptation of the classical u-net [7] with a ResNet encoder, set-up by default with 2 residual units, each over 4 different resolution scales, downsampling through *Strided convolution* with strides of 1, 2, 2 and 2. Among the existing segmentation protocols, the indicated one for covering the project requirements is the `fsl_fast` which replicates a brain tissue-wise segmentation on structural MRI, done by the FMRIB Library [8].
Preliminary tests showed as non-suitable using the pre-trained weights. Better results were reached training the network from scratch.
The architecture is implemented in the DLTK (Deep Learning Toolkit) Model Zoo [9], a comprehensive Deep Learning Toolkit for Medical Image Analysis, over TensorFlow. It is important to remark that the framework was customized to add early stopping, change the optimizer parameters and adding a new loss function.

4.4 VNet

The VNet architecture was originally designed to deal directly with 3D volumes, hence we found it to be a convenient architecture to be used for this project. We adapted the code from [10] from the problem of lung nodule segmentation to the problem of brain segmentation.

Figure 5 shows the structure of the VNet architecture. This diagram shows an example for 128x18x64 images. In our case, the input is a patch of size WxHxD. The output is shown for a 2 class problem (foreground, background), but in our case we have 4 output classes. Some of the main differences with respect to the UNet architecture are the following [11]:

- At each stage of both the compression and expansion paths, a residual function is learned
- Kernels of the convolutions are 5x5x5 as compared to the ones of the UNet, which are 3x3x3.
- PreLU activation can be used instead of simple ReLU
- 1x1x1 convolution is used as the last stage of the expansion path

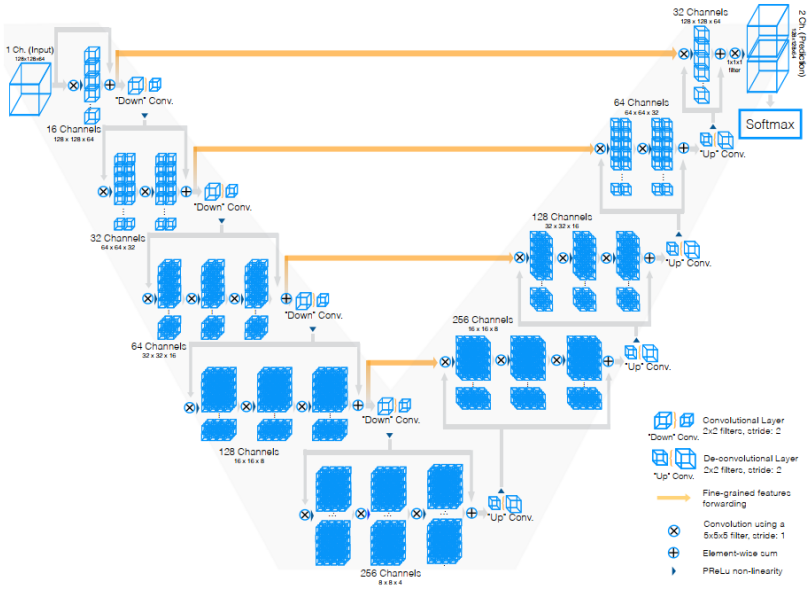


Figure 5: Used VNet architecture (from [12])

4.5 Loss Function

Although at the beginning we used cross-entropy (Eq. 1 as loss for the training process, the high-class imbalance caused by the small amount of CSF voxels caused the final results to be very poor for this tissue. To overcome this, we switched to dice loss (Eq. 2). Maximization of the dice loss, as proposed in [12], allows accounting for the class imbalance without the need of assigning weights to the samples of specific classes. In Eq. 2, p_i corresponds to the predicted segmentation volume and g_i corresponds to the ground truth. Because we deal with more than 2 classes, we compute the dice for every class (including the background), sum, and average by the mini-batch size.

$$L_{C.E.}(x, class) = -log\left(\frac{exp(x[class])]}{\sum_j exp(x[j])}\right) = -x[class] + log\left(\sum_j exp(x[j])\right) \quad (1)$$

$$L_{Dice} = \frac{2\sum_{i}^{N} p_i g_i}{\sum_{i}^{N} p_i^2 + \sum_{i}^{N} g_i^2} \tag{2}$$

5 Results and Discussion

To obtain quantitative results, we evaluated the performance of the models on the provided validation set, which was completely unknown to the trained models. We used two of the images from the training set as validation, and the remaining images as training. For validation we chose images *IBSR_07* and *IBSR_16* aiming at covering representative intensities and different spacing.

Testing Methodology The testing was performed over the explored networks in four different options to customize, leaving the architecture as it stands.

- Training and testing the available models with the provided dataset, and also with a pre-processed one where the intensities are normalized through Histogram Matching.
- Changing the loss function. In all the scenarios, at least testing with Dice loss implementation.
- Changing the optimizer from `AdaDelta` to `Adam` , since in some preliminary tests it showed a good performance. We trained the network with a learning rate scheduler, thus it reduced the learning rate when the validation loss is in a plateau.
- Increasing or decreasing the patch size. For the NeuroNet, using the freedom to limit the number of patches to generate during the training process.

Table 1 shows the obtained results for the most relevant models and configurations in terms of the mean Dice score, which is the common metric used for measuring the quality of segmentation. Table 2 shows the corresponding Absolute Volumetric Difference (AVD) and Hausdorff Distance (HD) values. Values of the segmentation metrics are shown also for the atlas-based segmentation using label propagation of the MNI atlas, which is a very common tool for brain image analysis.
From Tables 1 and 2 we can remark that the deep learning approaches outperform the label propagation approach with ATLAS. Although the atlas was implemented as a very simple approach, this comparison helps to build an idea about the performance that can be reached with traditional segmentation techniques.

Table 1: Dice scores for different models, data conditions and configurations

N	Model	Data	Optim.	Loss	DSC		
					CSF	GM	WM
1	Atlas	Raw	-	-	0.821	0.898	0.806
2	VNet	Pre-processed	Adadelta	Dice	0.860	0.918	0.920
3	NeuroNet	Pre-processed	Adam	Cross entropy	0.822	0.923	0.903
4	NeuroNet	Raw	Adam	Dice	0.865	0.912	0.902
5	Basic 3D UNet	Raw	Adadelta	Dice	0.897	0.937	0.931
6	Basic 3D UNet	Pre-processed	Adadelta	Dice	0.872	0.904	0.916
7	Basic 3D UNet	Raw	Adadelta	Cross entropy	0.896	0.935	0.919
8	3D UNet + Resample	Pre-processed	Adam	Cross entropy	0.893	0.931	0.922
9	3D UNet + instance norm.	Pre-processed	Adam	Dice	**0.920**	**0.943**	**0.937**

Table 2: Continuation of Table 1. Average Volumetric Differences and Hausdorff Distances for different models, data conditions and configurations

AVD			HD		
CSF	GM	WM	CSF	GM	WM
15.96%	4.05%	8.11%	50.972	**9.025**	14.300
13.30%	6.99%	5.75%	30.320	11.063	12.808
10.34%	5.41%	10.15%	29.182	11.006	10.180
11.72%	10.23%	14.15%	22.672	10.794	10.246
9.87%	5.74%	7.29%	42.463	12.017	10.268
14.33%	12.54%	10.48%	43.916	11.960	10.982
5.67%	7.08%	11.12%	33.229	11.093	8.720
6.98%	7.39%	6.92%	26.246	10.229	9.389
8.05%	**2.92%**	**5.31%**	**16.645**	9.605	**7.521**

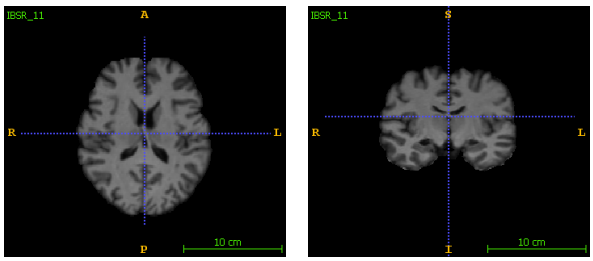
To have a graphical view of the performance of the models, we display in Figure 6 the seven best results obtained in this work. Based on this results the generation of the masks for the Testing images submitted for the challenge was extracted with models 5, 8 and 9.

Regarding the variety of architectures that were used for tackling the problem of brain segmentation, results show that the U-Net is a powerful architecture for this task. Especially when some more advanced strategies such as resampling and instance normalization are used, results can be improved notably. For the particular case of the VNet, although it has a more complex structure than the UNet, its performance was not what we expected. The increase in the training time that it implies (although speed increases when using normalized data) with respect to the UNet does not provide enough justification for using it for this task. The same can be said about the NeuroNet, which reached the poorest values in CSF and WM segmentation.

Other variations of the UNet were also explored, although not included in the tables. Particularly, we tried to modify the architecture to have a higher number of levels along with more horizontal convolutions, trying to make it more similar to the original architecture, as described in [4]. However, this modification produced poorer results, which allows understanding that more layers and more features do not necessarily imply a better quality of the segmentation results.

The Dice loss was found to be effective at providing a solution to the class imbalance problem that is caused by the low amount of CSF voxels compared to GM and WM.

5.1 Cases of Study



(a) Axial View (b) Coronal View

Figure 7: MRI sequence IBSR 11

This subsection presents the segmentation results of the scan ISBR 11 depicted in Figure 7. According to Table 1 the dice coefficients for all the cases are relatively similar and consistent. Thus, we will present only a case of study of the validation images respect to their ground truth masks. According to the metrics, most of the segmentation results were achieved with more than 80 % of DICE similarity. Therefore, the resulting masks show that there are not so easily noticeable mislabels of the segmentation results with respect to the ground truth image.

The ATLAS based segmentation is a label propagation result from the generation of an atlas by using the available Training images. This method has shown the lowest

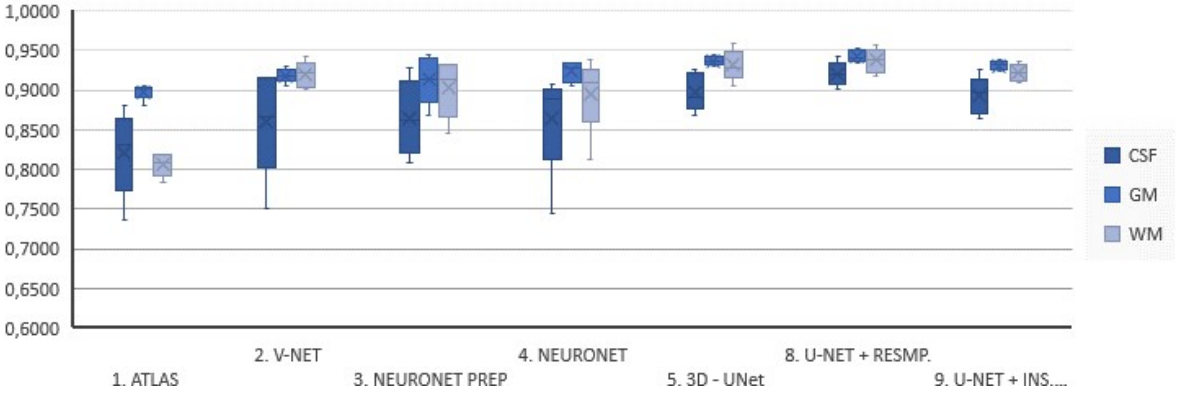


Figure 6: Box Plots of the implemented approaches

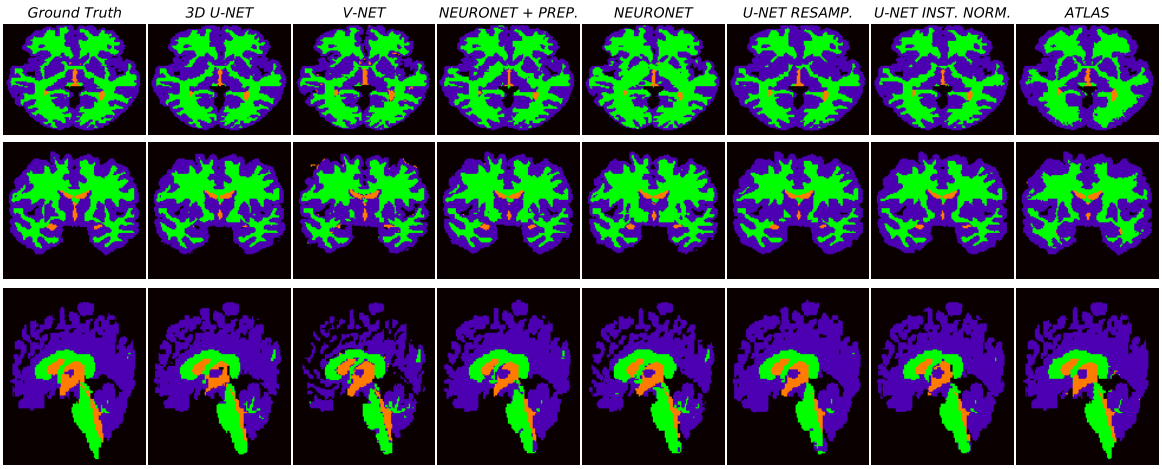


Figure 8: Results of Scan IBSR 11, The colors of orange, blue and green represents the labels of segmentation for cerebrospinal fluid (CSF), grey matter (GM) and white matter (WM). The three rows are the two dimensional axial, coronal and sagittal views of three dimensional segmentation results, respectively. From left to right column (a) the ground truth; Segmentation Results: (b) 3D U-NET, (c) V-NET, (d)Neuronet + Preprocessing, (e) Neuronet, (f) U-Net + CSF Resample, (g) U-Net with instance normalization, (h) Label Propagation with Atlas

performance metrics. It is noticeable that the structures of the brain are smoothed, this is mainly due to the naturalizing of the atlas used to perform the label propagation. We can also see that the results of the different architectures adapt to the different structures of the brain. However, it is worth mentioning that in the result of the V-Net removes some of the GM in the sagittal plane of the brain. However, this occurs because the selected slice corresponds to the division of the cerebral hemispheres.

Another discrepancy present in the Figure 8 can be noticed in the coronal and axial view of the Neuronet segmentation, where the white matter is over segmented and removes some of the subcortical structures present in the ground-truth mask.

6 Conclusions

In this project, we implemented a patch-based deep learning approach for brain MRI image segmentation into the 3 basic tissue types: CSF, GM, and WM. Instead of dealing with only the provided architecture of the U-Net, we decided to explore also other architectures that could be also applied to this problem, such as the V-Net and the NeuroNet. Change the problem domain from brain nodule segmentation to brain segmentation, for example, was found to be a very useful task to understand better the architecture and the function of the different building blocks that constitute the model.

One consideration that allowed us to improve the segmentation result and to deal with the imbalance was resampling the CSF patches, this reduced the heavy ratio among the classes and helped to make the model more specialized in segmenting of the CSF.

In this project, we explored not only the available techniques to perform medical image segmentation, but we also made a comparison between a classical ATLAS approach and the new-trend deep-learning methodologies, as we expected the deep learning approaches that outperform the classical method. However, these methods require of the exhaustive training process and careful selection of parameters, that could be a drawback to put one of this algorithm in production. In the other hand the classical approach contains a well defined pipeline that can be easily put in production.

The obtained results are consistent with what was expected from a deep learning approach. We can highlight the importance of choosing the hyper-parameters properly for performing successfully training and also to explain possible sources of errors. It is also important to understand the limitations of the implemented approach. One of these limitations is the training time, which took several hours for some cases. This of course also implies the need for an environment with GPU.

It is important to remark that the training cycles for the NeuroNet were significantly reduced, compared with the other tested architectures, mainly since the network was not trained on every step with all the available patches of a volume. For NeuroNet, a limited set of random patches are chosen as a dataset augmentation strategy, for training the network, and, although the reached results were not optimal, it would be an interesting strategy to apply on the other explored architectures, to reduce the training time.

References

[1] Ivana Despotović, Bart Goossens, and Wilfried Philips. Mri segmentation of the human brain: challenges, methods, and applications. *Computational and mathematical methods in medicine*, 2015, 2015.

[2] Nitrc: Ibsr: Internet brain segmentation repository. https://www.nitrc.org/projects/ibsr. , Accessed: 2020-10-01.

[3] Murali Shanker, Michael Y Hu, and Ming S Hung. Effect of data standardization on neural network training. *Omega*, 24(4):385–397, 1996.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[5] Xiao-Yun Zhou, Peichao Li, Zhao-Yang Wang, and Guang-Zhong Yang. U-net training with instance-layer normalization. In *International Workshop on Multiscale Multimodal Medical Imaging*, pages 101–108. Springer, 2019.

[6] Martin Rajchl, Nick Pawlowski, Daniel Rueckert, Paul M. Matthews, and Ben Glocker. Neuronet: Fast and robust reproduction of multiple brain image segmentation pipelines, 2018.

[7] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation, June 2015.

[8] Fmrib software library v6.0. https://fsl.fmrib.ox.ac.uk/fsl/fslwiki. , Accessed: 2020-03-01.

[9] Dltk model zoo - github. https://github.com/DLTK/models. , Accessed: 2020-03-01.

[10] Mattmacy. A pytorch implementation for v-net: Fully convolutional neural networks for volumetric medical image segmentation, github repository, Mar 2017.

[11] Sik-Ho Tsang. Review: V-net - volumetric convolution (biomedical image segmentation), Mar 2019.

[12] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016.