BRAIN TISSUE SEGMENTATION USING NEURONET WITH DIFFERENT PRE-PROCESSING TECHNIQUES

A PREPRINT

Fakrul Islam Tushar
Erasmus + Joint Master In Medical Imaging and Applications
University of Girona
Girona, Spain
Fakrul-Islam_Tushar@etu.u-bourgogne.fr

Basel Alyafi
Erasmus + Joint Master In Medical Imaging and Applications
University of Girona
Girona, Spain
u1951852@campus.udg.edu

Md. Kamrul Hasan
Erasmus + Joint Master In Medical Imaging and Applications
University of Girona
Girona, Spain
kamruleeekuet@gmail.com

Supervisors
Robert Martí, PhD
robert.marti@udg.edu
Xavier Lladó, PhD
xavier.llado@udg.edu

January 11, 2019

ABSTRACT

Automatic segmentation of MRI brain images is one of the vital steps for quantitative analysis of brain for further inspection. Since manual segmentation of brain tissues (white matter (WM), gray matter (GM) and cerebrospinal fluid (CSF)) is a time-consuming and tedious task that engages valuable human resources, hence, automatic brain tissue segmentation draws an enormous amount of attention in medical imaging. In this project, NeuroNet has been adopted to segment the brain which uses Residual Network (ResNet) in encoder and Fully Convolution Network (FCN) in the decoder. To achieve the best performance, various hyper-parameters have been tuned, while, network parameters (kernel and bias) were initialized using the NeuroNet pre-trained model. Different pre-processing pipelines have also been introduced to get best a robust trained model. The performance of the segmented validation images were measured quantitatively using Dice Similarity Co-efficient (DSC) and were reported in the best case as 0.8986 ± 0.0174 for CSF, 0.9412 ± 0.0086 for GM, and 0.9335 ± 0.0166 for WM. We worked out that keeping the original patch size and using histogram preprocessing with 4000 steps had the highest achievable performance.

Keywords Brain tissue segmentation · cerebrospinal fluid (CSF) · gray matter (GM) · white matter (WM) · NeuroNet · Residual Network (ResNet) · Fully Convolution Network (FCN) · Dice Similarity Co-efficient (DSC).

1 Introduction

Brain image segmentation plays a crucial role in brain image analysis which extracts brain tissue (white matter (WM), gray matter (GM) and cerebrospinal fluid (CSF)) from a brain image by partitioning it into a set of disjoint regions that have similar characteristics such as intensity homogeneity, texture etc [1]. Segmentation of brain tissue

helps to detect diseases like brain tumor, Alzheimer's disease (AD), Parkinson's disease, dementia, schizophrenia etc, also brain disorder identification and whole brain analysis of traumatic injury [2]. The performance of brain tissue segmentation methods depends on several factors such as location, size, shape, texture of tissues and unclear tissue boundary (due to inherent in the modalities used for image acquisition) [3]. Traditionally, brain tissue segmentation can be classified into 5 categories such as: 1) Manual, 2) Region-based, 3) Thresholding-based, 4) Clustering-based and 5) Feature extraction and classification-based segmentation [1].

In the above mentioned traditional methods, the training process of a classifier does not affect the nature of the extracted features and in addition most of the feature extraction methods require spatial and intensity information for accurate brain tissue segmentation. Recently, Convolutional Neural Networks (CNNs) and deep learning are gaining recognition in brain tissue segmentation [4].

In this project, we adopt NeuroNet which is a comprehensive brain image segmentation tool based on a novel multi-output CNN architecture which has been trained to reproduce simultaneously the output of multiple state-of-the-art neuroimaging tools [5].

2 Dataset

The dataset used for this work was IBSR18. IBSR is a publicly available dataset from the Center for Morphometric Analysis at Massachusetts General Hospital [6], which is one of the standard datasets for tissue segmentation evaluation.The dataset is composed of 18 T1-W volumes and with different slice thicknesses. The provided images are skull stripped and bias field corrected. For this work, the dataset was divided into three sets: ten for training, five for validation, and three for testing. A brief description of the dataset is given in Table 1. For the training and validation sets, corresponding tissue labels (CSF, GM and WM) were provided. The training set was used for training, while validation set was used to tune the proposed model. Figure 1 shows the volume IBSR_01 and corresponding labels.

Training Dataset		
Volume Name	Volume	Spacing (mm)
IBSR_01,IBSR_03,IBSR_04,IBSR_05,IBSR_06	$256 \times 128 \times 256$	$0.9375 \times 1.5 \times 0.9375$
IBSR_07,IBSR_08,IBSR_09,	$256 \times 128 \times 256$	$1 \times 1.5 \times 1$
IBSR_16,IBSR_18	$256 \times 128 \times 256$	$08371 \times 1.5 \times 0.8371$
Validation Dataset		
Volume Name	Volume	Spacing (mm)
IBSR_11,IBSR_12	$256 \times 128 \times 256$	$1 \times 1.5 \times 1$
IBSR_13,IBSR_14	$256 \times 128 \times 256$	$0.9375 \times 1.5 \times 0.9375$
IBSR_17	$256 \times 128 \times 256$	$08371 \times 1.5 \times 0.8371$
Test Dataset		
Volume Name	Volume	Spacing (mm)
IBSR_02	$256 \times 128 \times 256$	$0.9375 \times 1.5 \times 0.9375$
IBSR_10	$256 \times 128 \times 256$	$1 \times 1.5 \times 1$
IBSR_15	$256 \times 128 \times 256$	$08371 \times 1.5 \times 0.8371$

Table 1: Summary on IBSR18 dataset used in this project.

3 Pre-processing

Important tools in MRI Brain analysis include image registration, segmentation, then tissue volume measurement. However, those become challenging when different scanners were used with different parameters during acquisition, which usually lead to data heterogeneity, mismatched intensity distributions and contrast variations, and noise [7]. Therefore, prior to using those tools, data standardization steps are required.

Authors in [8] used intra-subject registration as the first step of pre-processing for Lupus segmentation. Dolz et al. in [9] used volume-wise intensity normalization, bias field correction and skull-stripping. Shakeri et al. [10] used registration and normalization for subcortical parcellation method. Nyul et al. proposed a method consist of a training stage to find standard parameters then matching the histograms to a standard histogram through a transformation stage [11].

In this work two different pre-processing pipelines were implemented. To see the effect on the performance of the deep CNN with different pre-processing scheme. Figure 2 shows the overview of the pre-processing pipelines.

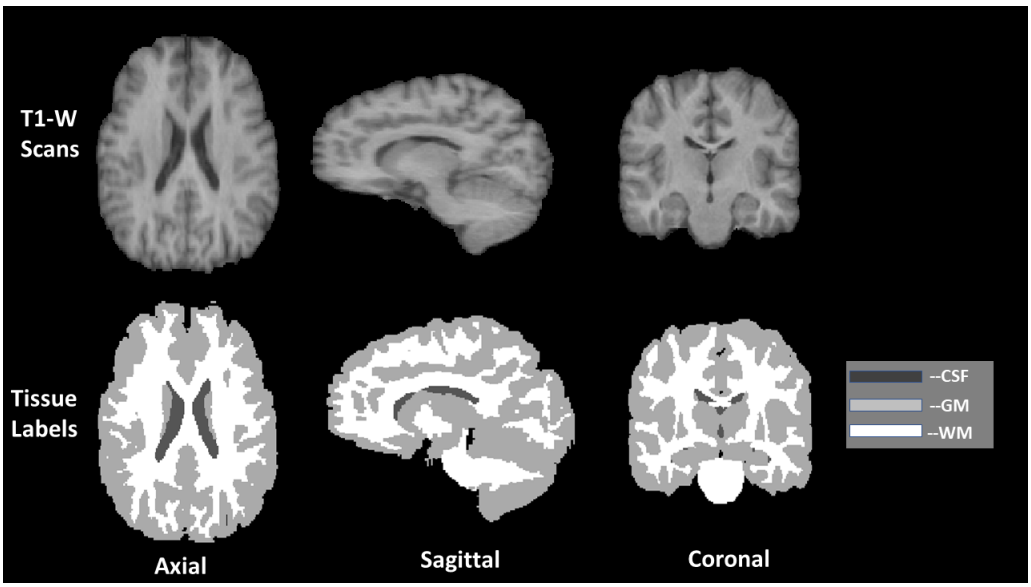


Figure 1: Graphical Description of IBSR_01 volume and Corresponding label in axial, sagittal and coronal view.

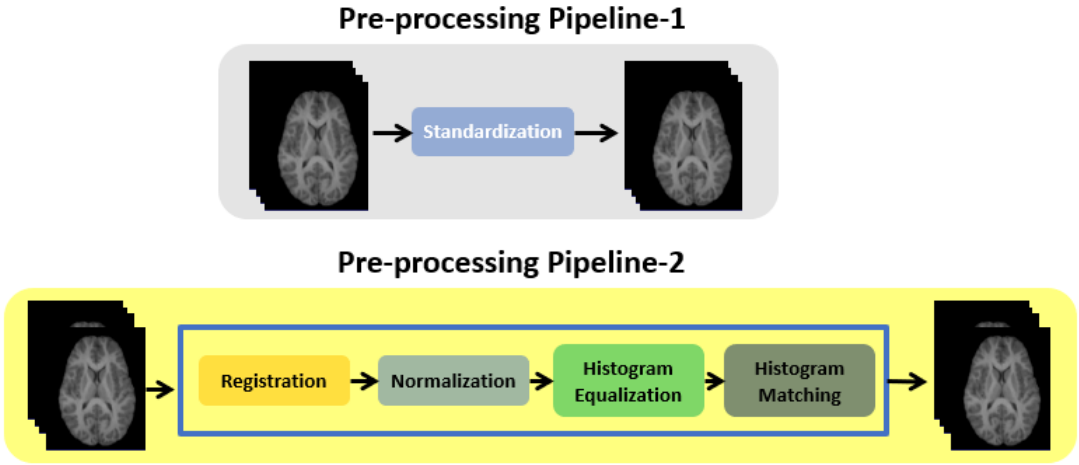


Figure 2: Pre-processing pipelines.

3.1 Pre-processing Pipeline-1

The input volumes were standardized to zero mean and unit standard deviation using the volume statistics as mentioned in the reference paper of Rajchl et al. [5]. It can be formulated as equation.1.

$$V_{new} = \frac{V_{old} - \mu}{\sigma} \tag{1}$$

Where μ, σ represents the mean and standard deviation of all pixels in the corresponding volume respectively.

3.2 Pre-processing Pipeline-2

The pre-processing pipeline-2 has been described briefly as follows in 3.2.1 and 3.2.2.

3.2.1 Registration

Registration is the process of spatial normalization/ alignment of two or more images in a common anatomical space [12], called the fixed space. As shown in table.1 training, validation and test datasets have different spacing. To standardize the dataset, we registered the images and transformed the labels correspondingly to Montreal Neurological Institute (MNI) template (n=152 subjects, 1x1x1mm T1w, skull stripped) [13]. Referred MNI template volume dimensions were 182 x 218 x 182. Figure. 3 shows the process of the registration. Simple-ITK framework in Python was used for the registration [14]. Registration was done in two steps:

- Registering the dataset to MNI template using rigid transform and saving the corresponding final transformation matrix. The available dataset was used as moving images, while, MNI template was used as a fixed image.
- Using the inverse of the transformation matrix, transform the corresponding predicted labels for training and validation datasets back to the original space.

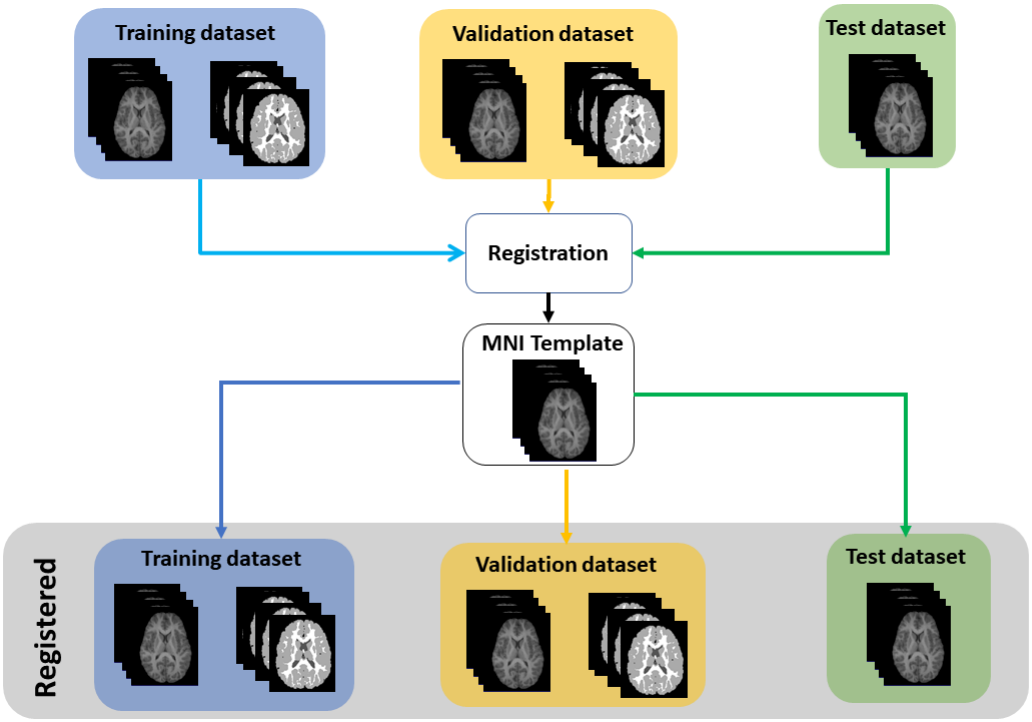


Figure 3: Registration to MNI template.

3.2.2 Normalization

In this step, intensity rescaling was applied to each volume. The intensity range of the volumes was rescaled to the range $[0, 1]$. It can be formulated as equation.2.

$$v_{new} = \frac{v - min(v)}{max(v) - min(v)} \tag{2}$$

3.2.3 Histogram Pre-processing

In this step, the procedure was as follows:

1. **Reference Selection**: by analyzing tissues distribution for all volumes, image IBSR_07 was nominated due to the following reasons:
   - It has a relatively wide spectrum of intensity values, see Figure 4 left part.
   - The overlapping between GM and WM is acceptable, see Figure 4 middle part.

- GM and WM had comparable shares, this will help in next step, histogram equalization ,for the reference image.

2. **Histogram Equalization of Reference Volume**: Adaptive histogram equalization was applied on the reference normalized volume IBSR_07. It's a process of adaptive image-contrast enhancement based on a generalization of histogram equalization (HE) [15].
3. **Histogram Matching to the Reference One**: Apply histogram matching of all the dataset (minus IBSR_07) to the reference volume. As a last step of the Pre-processing pipeline-2, histogram matching was performed to map all volumes' histogram distributions to the reference volume's one. Figure 5 shows the raw volumes and the pre-processed volumes after applying pre-processing pipeline-2.

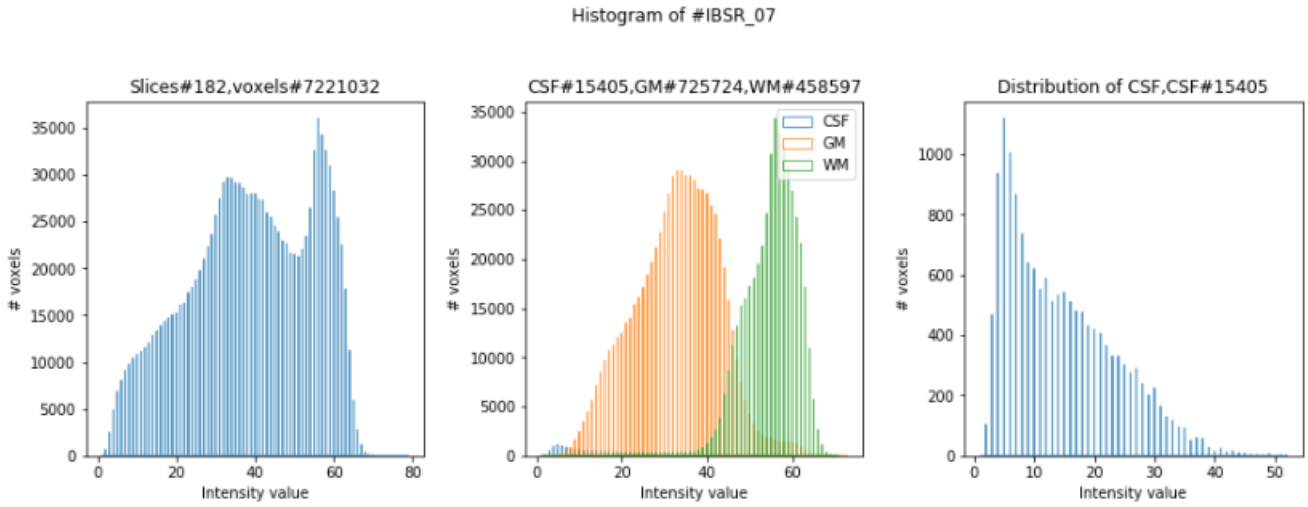


Figure 4: tissues intensity distribution, on the left, the complete distribution. In the middle, the white matter and grey matter distributions. On the right, CSF distribution.

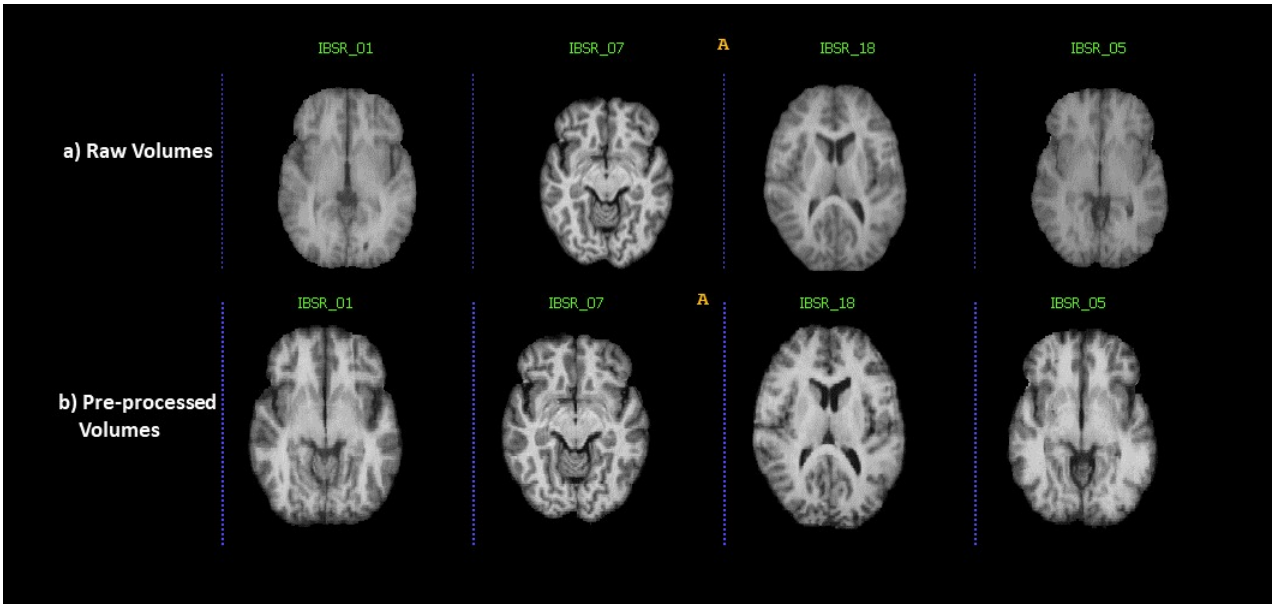


Figure 5: The effect of pre-processing the dataset, on the top, 4 cases before applying the pre-processing pipelines. At the bottom, the final pre-processed dataset.

4 Method

4.1 Network Architecture

For implementation of the project we adopt the NeuroNet [5] architecture and the code is available at Github repository of DLTK models [16]. NeuroNet is a deep convolutional neural network multi-output architecture, which is trained on 5,000 T1-weighted brain MRI scans from the UK Biobank Imaging Study that have been automatically segmented into brain tissue and cortical and sub-cortical structures using the standard neuroimaging pipelines [5]. For our work, as desired output is tissue segmentation only the , the architecture modified to an updated FCN architecture [17] with a ResNet encoder [18] as presented in [19]. Figure. 6 shown the original NeuroNet Architecture and the adoped architecture in this work.

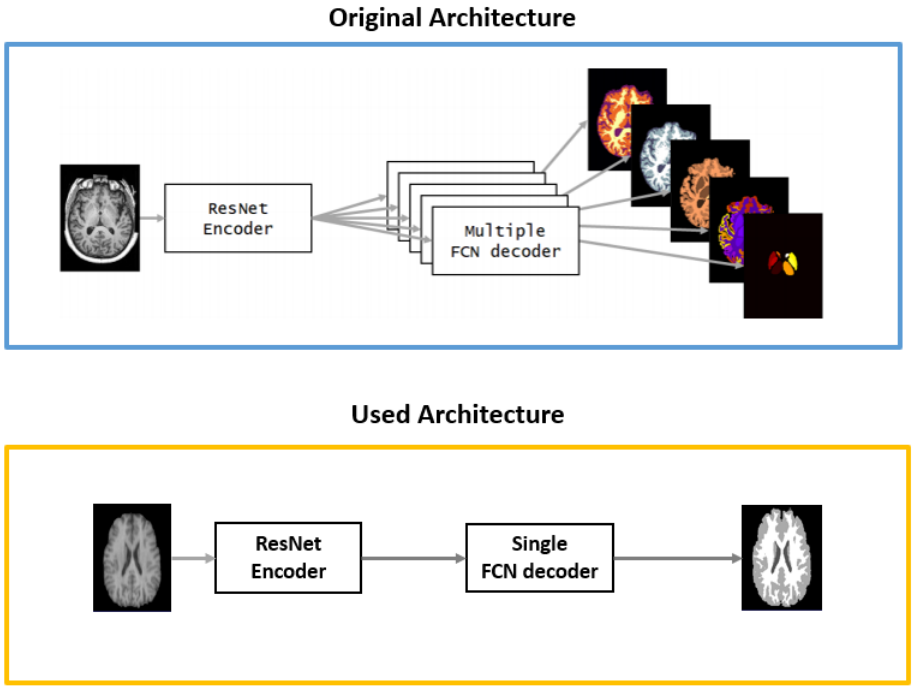


Figure 6: Network Architectures

One initial convolution was performed on input volumes,afterward features were extracted using the ResNet encoder [18] [19].Features were extracted in encoder part with two residual units ($N_{unit} = numberofresidualunitateachscale$), $U_i^{S_j} = \left\{U_1^{S_j}, U_2^{S_j}\right\}$ on each of the resolution scales $S_j = \{S_1, ..., S_{N_{scales}}\}$, where N_{scales} is the number of resolutions scales. In our work we used $N_{scales} = 4$,as used in default implementation [5].Leaky ReLu (leakiness = 0.1) was used as activation function [20] with preceding batch normalization. At each scale the down-sampling was performed using stride convolution [21] and where the $strides_j = \{1, 2, 2, 2\}$ operate in each spatial dimension.As defined in the reference paper [5] ,fixed number of filters for all convolutions in U^{S_j} was used doubling the number at each scale: 16, 32, 64 and 128.

In image segmentation fully convolutional networks (FCNs) is among widely used networks which is typically reconstruct the prediction as the same size of the input given. In original Neuronet architecture decoding part was based on multi-decoder architecture on FCN upscore operations [17].Prediction was reconstruct at each resolution scale S_{j-1} by up-sampling the prediction linearly at S_j scale and adding skip connection from the output of the last residual unit $U_2^{S_j}$.The output of the last residual unit at decoder serves as output of the network. A prediction was obtained applying $softmax$. Loss was calculated using categorical cross-entropy loss for all prediction outputs $\hat{y}$ at voxel locations v.

$$L(\hat{y}, y) = -\sum \hat{y}(v) \log y(v) \tag{3}$$

Where y is the true label volume.

4.2 Post-Processing

In case of segmentation post-processing can be beneficial for improve the performance [22]. As a post processing steps we implemented Conditional Random Fields as proposed in [23] using deep learning platform for medical imaging NiftyNet [24]. Applying post-processing wasn't make any significant improvement.

5 Environment

To train/validate/test the NeuroNet. The specification of the used GPU is mentioned in the following Table 5.

Specifications of the GPU	
Item Name	Specifications
NVIDIA-SMI	390.48
Driver Version	390.48
Graphics card	GeForce GTX 1080
Memory	2.7GB

6 Experiments

Dice Similarity Coefficient (DSC) was used for evaluating predicted labels and guiding the tuning process. Our main idea of the experiments was: firstly tune the model perfectly on the data with default parameters suggested in [5] ,then tune the parameters to improve the results. Here we discuss five main experiments that we have found to be meaningful. We carried out many other experiments which we discard here. The corresponding results are given in section 7. In Experiments 3 and 4, the proposed pre-processing pipeline-2 was applied on the datasets explained in Section 3.2. All the volumes were bring to same spatial coordinate by applying registration, then the volumes were normalized and afterward histogram matching was performed according the reference volume (IBSR_07).

6.1 Experiment_1

Firstly, we have trained our model with the processed data that went through only Standardization(0 mean and Unit standard deviation) mentioned in section 3.1. The model was trained from scratch but in the case of training deep models, enough care needs to be taken to initialize the parameters as shown in [22]. We used the pre-trained weights of Neuronet [5] as the initial ones to avoid variance vanishing problem. We trained the model for 1000 steps with 200 randomly extracted patches of size 128x128x128 mm. Reason for choosing such a big patch size (128x128x128 mm) was that authors in [5] used in original implementation. Performance of this model on the validation data is shown in Table 2 and Figure 7a.

In next attempt, the training steps were increased five times than before and trained the model for 5000 steps with double the number of patches (400 patches). This increased training and number of samples turns a huge improvement in the model's prediction in CSF from average 0.42 to 0.79 and slight increase and decrease in WM and GM prediction respectively. Results are shown on Table. 3. Afterward more training was performed but no significant improvement was achieved, which leads to looking for different pre-processing strategies.

6.2 Experiment_2

To improve the result, in this stage with the initial pre-processing pipeline-1 (Section 3.1) histogram matching was added in the pre-processing step. Histogram matching was performed randomly selecting one volume from the training dataset and it was IBSR_06. Table. 4 shown an overview of the trained models performance on the validation dataset.

Here again the pre-trained model's weight was used as the initial weight and trained the model with patches of size 128x128x128mm. We trained different model with different hyper-parameters such as changing the number of training steps and number of patches. from Table. 4 it can be clearly seen that how prediction or the segmentation by model is improving with respect the larger iteration of training. For explanation purpose we give different models name as 6.2.1 to 6.2.6 in this section, shown in Table. 4.

In case of model no 6.2.1 and 6.2.3 training with the same number of steps but with different number of patches makes a huge difference in CSF segmentation which was around 12 unit improvement due to reducing the number of patches to 1/4. Figure. 8a and 8b shown the DSC boxplot of the model no 6.2.1 and 6.2.2. Figure. 8b shown that CSF segmentation if Very stable in model 6.2.2 compared to model no 6.2.1. It can be said that with less number of patches the model is

learning well as the size of the patch is quite large, 50 patches from a 10 volume training is set is good enough in our observation.

6.3 Experiment_3

The proposed pre-processing pipeline-2 was applied on the datasets explained in Section 3.2. The pipeline consists of registration, normalization, adaptive histogram equalization and histogram matching steps. Training the model with the pre-processed data using pipeline-2 performed really well with 4000 training steps. Table.5 shows the performance of the model evaluated on the validation dataset. Overall, this experiment proved the best combination of parameters compared to all other experiments.

6.4 Experiment_4

As we achieved the so far the highest performance with respect to model no 6.3.2 and experiment 3. Next the analysis of the model performance under different hyper-parameter was performed. Sampling the number of patches contribute a lot in performance of the prediction as shown in Table. 4 earlier. In this stage we tried two different strategies for extracting the sample patches:

- Class balance Extraction: It will extracted the same number of patches from each of the classes in this case as we have four class CSF, GM, WM and Background.
- Random Extraction: It will randomly extract patches from the volumes.

Table. 6 shown the evaluation results on the validation dataset with the model training on class balance and random patch extraction strategy, and figure. 10 shown the corresponding DSC boxplot. All these models mentioned in Table. 6 were trained for 4000 steps and 50 patches of size 128x128x128mm. From Table. 6 can be seen random extraction of the patches is better than class balance patch extraction with both uniform distribution initialization and pre-trained weight initialization.

6.5 Experiment_5

We tried three different cases using pipeline2 for pre-processing, those cases are as follows:

- Case 1 : patch size: 32^3, training steps 4000, 200 samples.
- Case 2 : patch size: 64^3, training steps 4000, 200 samples.
- Case 3 : patch size: 128^3, training steps 4000, 50 samples.

Table. 7 and fig. 11 show the results for all those cases.

6.6 Implementation

NeuroNet is implemented using Deep Learning Toolkit (DLTK) for Medical Image Analysis [19] on TensorFlow [25] with SimpleITK [26] for data I0 interface. Pre-prpcessing pipelines used in this work were implemented using DLTK and SimpleITK framework.

7 Results

In this section, subjective, using overlayed predicted validation labels and ground truth, and objective evaluations, using dice coefficient, are provided.

7.1 Experiment_1 Results

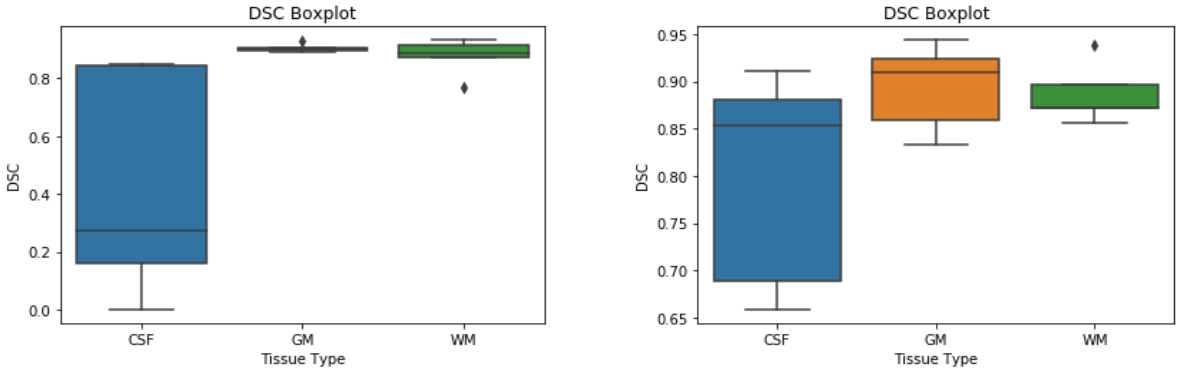
Here, we show the results using the parameters and model described in Experiment_1. With only 1000 training steps, the model is capable of extracting GM and WM with average dice of 0.90 and .87 respectively as shown in Table. 2 but a significantly lower performance on CSF segmentation with an average dice of 0.42. But, using 5000 training steps, the performance of the CSF segmentation increases much more although the performance of WM and GM reduces little bit.

Metric	DSC		
Tissue Type	CSF	GM	WM
IBSR_11	0.8497	0.9052	0.9154
IBSR_12	0.8428	0.8960	0.8879
IBSR_13	0.2755	0.9014	0.8737
IBSR_14	0.1610	0.9278	0.9327
IBSR_17	0.0	0.8913	0.7696
Mean	0.4258	0.9043	0.8758
Std	0.3961	0.0141	0.0637

Table 2: Validation results using 1000, iterations of training, 200 samples, pre-processing pipeline 1, $128 \times 128 \times 128$ patch size.

Metric	DSC		
Tissue Type	CSF	GM	WM
Mean	0.7981	0.8935	0.8871
Std	0.1160	0.0462	0.0318

Table 3: Results on Validation dataset training 5000 step, 400 samples on dataset pre-processed using pipeline-1, $128 \times 128 \times 128$ patch size



(a) 1000 training iterations

(b) 5000 training iterations

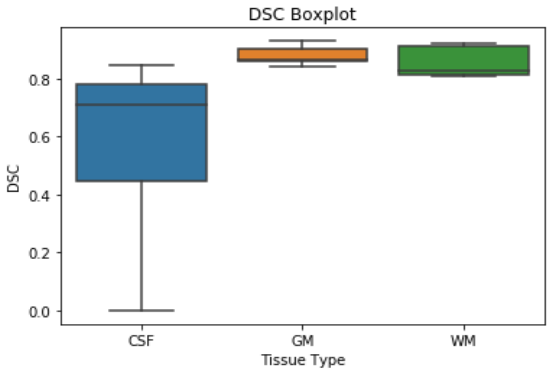
Figure 7: Box plot for validation results using Pipeline 1 pre-processing

7.2 Experiment_2 Results

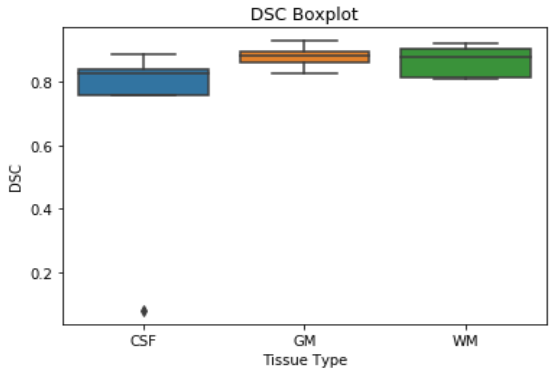
The Best result achieved in this stage was with 20000 steps of training with 50 samples the average DSC for CSF, GM and WM was 0.79, 0.90 and 0.87 respectively.

	Model Type	Model Taring Type	Validation Average DSC					
Model No	#Training Steps	#Samples (number of patches)	CSF	Std	GM	Std	WM	Std
6.2.1	1000	200	0.5553	0.3457	0.8787	0.0367	0.8558	0.0560
6.2.2	1000	50	0.6790	0.3382	0.8803	0.0380	0.8665	0.0520
6.2.3	3000	50	0.7544	0.1012	0.8932	0.0355	0.8706	0.0627
6.2.4	5000	50	0.7771	0.1300	0.9027	0.0367	0.8679	0.0851
6.2.5	5000+1000	50 +100	0.7102	0.3952	0.9048	0.0474	0.8575	0.1232
6.2.6	20000	50	0.7922	0.2155	0.9044	0.0344	0.8726	0.0667

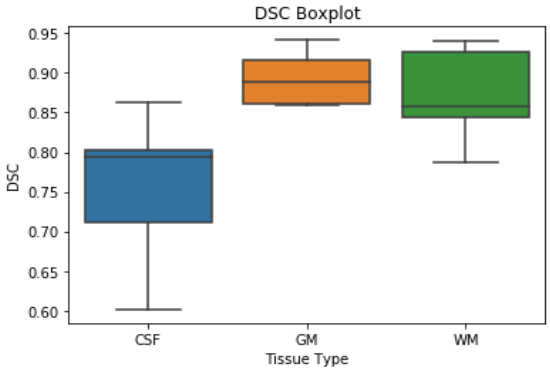
Table 4: Results on Validation dataset Histogram Matching and pipeline-1



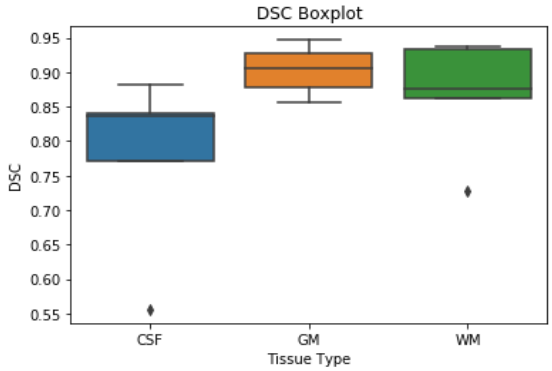
(a) Validation DSC of model 6.2.1



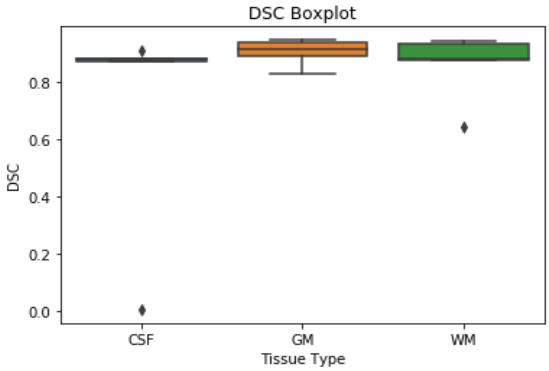
(b) Validation DSC of model 6.2.3.2



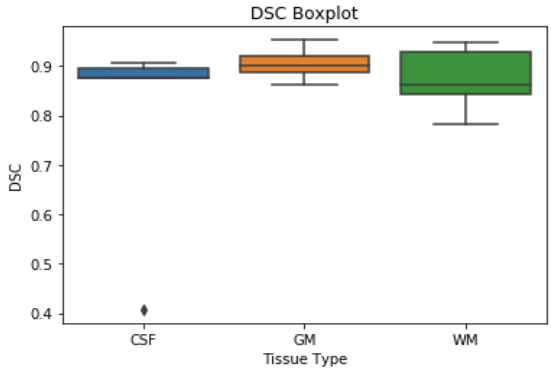
(c) Validation DSC of model 6.2.3



(d) Validation DSC of model 6.2.4



(e) Validation DSC of model 6.2.5



(f) Validation DSC of model 6.2.6

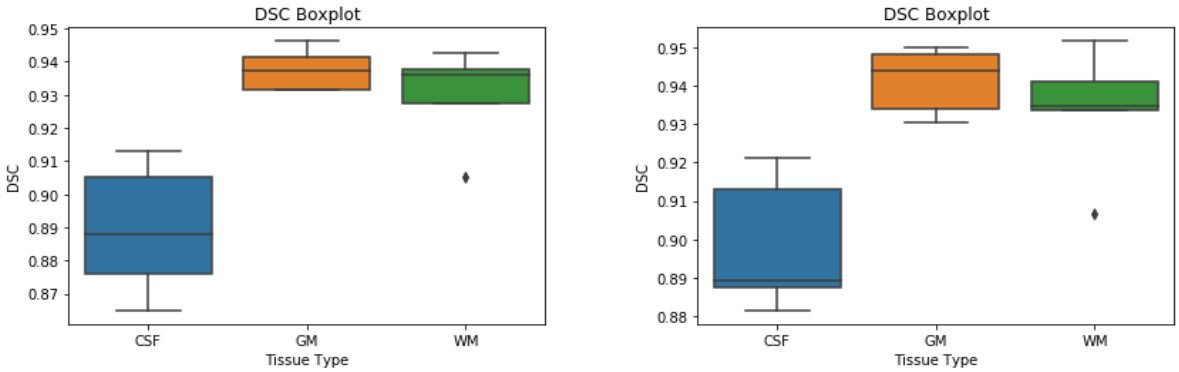
Figure 8: Box plots for Experiment 2, 6 cases

7.3 Experiment_3 Results

Proposed pre-processing pipeline-2 shown an remarkable improvement in the segmentation prediction in all three tissue cases especially in CSF segmentation. Table. 5 shown the evaluation done of validation data in average DSC.

	Model Type		Validation Average DSC					
Model No	#Training Steps	#Samples (number of patches)	CSF	Std	GM	Std	WM	Std
6.3.1	2000	50	0.8893	0.0199	0.9376	0.0064	0.9298	0.0147
6.3.2	4000	50	0.8986	0.0174	0.9412	0.0086	0.9335	0.0166

Table 5: validation DSC using the models trained on data pre-processed by pipeline-2



(a) Validation DSC of model 6.3.1

(b) Validation DSC of model 6.3.2

Figure 9: validation DSC by Models trained on data pre-processed by Proposed pre-prpcessing pipeline-2

7.3.1 Experiment_4 Results

	Model Type		Validation Average DSC					
Model No	Extract Patch	Initialization	CSF	Std	GM	Std	WM	Std
6.3.1.1	Class Balance	Uniform Distribution	0.8796	0.0267	0.9375	0.0044	0.9255	0.0129
6.3.1.2	Random	Uniform Distribution	0.8867	0.0198	0.9309	0.0098	0.9275	0.0103
6.3.1.3	Class Balnce	Pre-trained Weight	0.8916	0.0197	0.9348	0.0112	0.9265	0.0249
6.3.1.4	Random	Pre-trained Weight	0.8949	0.0194	0.9416	0.0072	0.9330	0.9330

Table 6: Balanced vs Random Sampling of training patches

7.3.2 Experiment_5 Results

	Model Type		Validation Average DSC					
Patch Size	Training Steps	#Samples	CSF	Std	GM	Std	WM	Std
32x32x32	4000	200	0.06917	0.02826	0.7079	0.0438	0.7166	0.0630
64x64x64	4000	200	0.8015	0.0463	0.90343	0.0185	0.8863	0.0328
128x128x128	4000	50	0.8867	0.0198	0.9309	0.0098	0.9275	0.0103

Table 7: Performance of the model with different patch sizes

7.4 Qualitatively Speaking

Here, 5 representative near-middle slices were selected from the 5 validation volumes. Figures 12 and 13 show subjective results. It can be seen that the algorithm is working very well. Mismatchings are hardly seen.

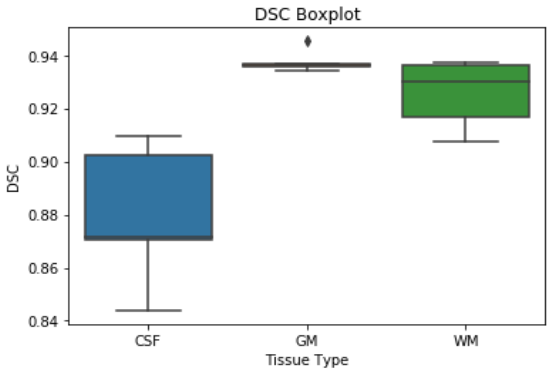
8 Acknowledgement

we would like to convey grateful thanks to University of Girona (UdG) for providing the GPU server to do our experiments.

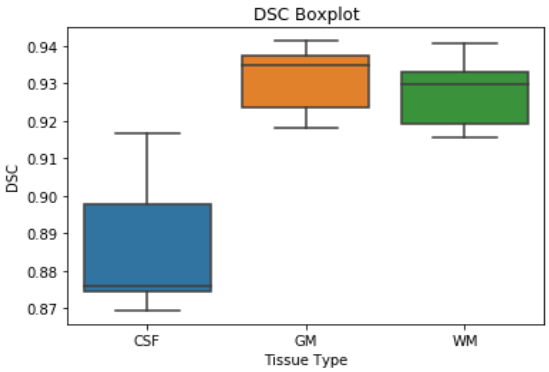
9 Discussion

So far, we have seen the following main points:

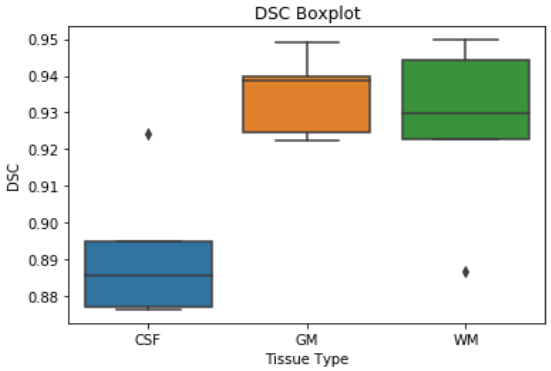
- Preprocessing was clearly effective in boosting the performance. This was shown by using pipeline 2 instead of pipline 1. Histogram equalization and matching were playing a great role. Registration or spatial normalization helped as well in unifying the voxel spacing mainly so all inputs belong to similar intensity and spatial spaces.



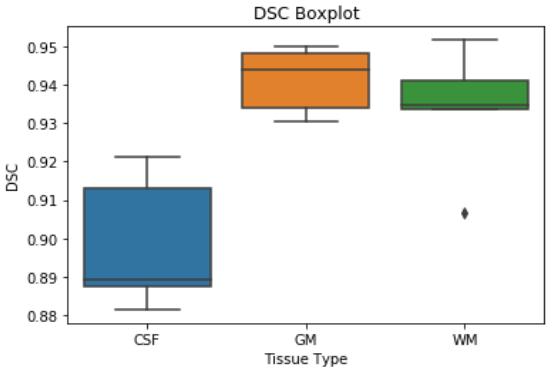
(a) Validation DSC of model 6.3.1.1



(b) Validation DSC of model 6.3.1.2

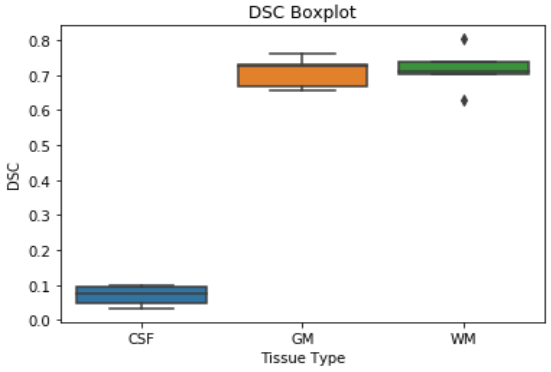


(c) Validation DSC of model 6.3.1.3

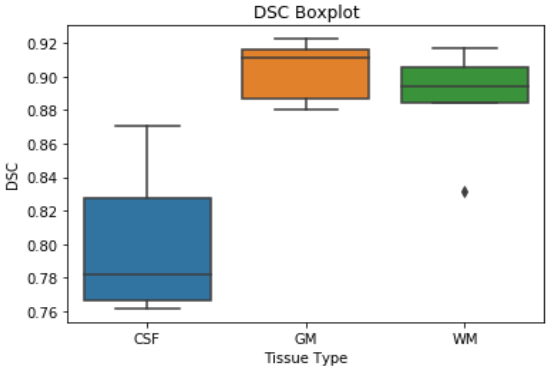


(d) Validation DSC of model 6.3.1.4

Figure 10: Balance vs Random Sampling of Training Patches dice



(a) Performance of the model with patch sizes 32x32x32



(b) Performance of the model with patch sizes 64x64x64

Figure 11: Performance of the model with different patch sizes.

- The use of NeuroNet pretrained weights was significantly helping the network to start from a good initial point. As compared to starting with random initialization, the pretrained weights were performing enormously better even though the size of the used dataset was relatively small (10 volumes compared to 5000 original size).
- Patch sizes had important effect on the performance as well. The original patch size was the best fit, while, smaller sizes just did not work as good.

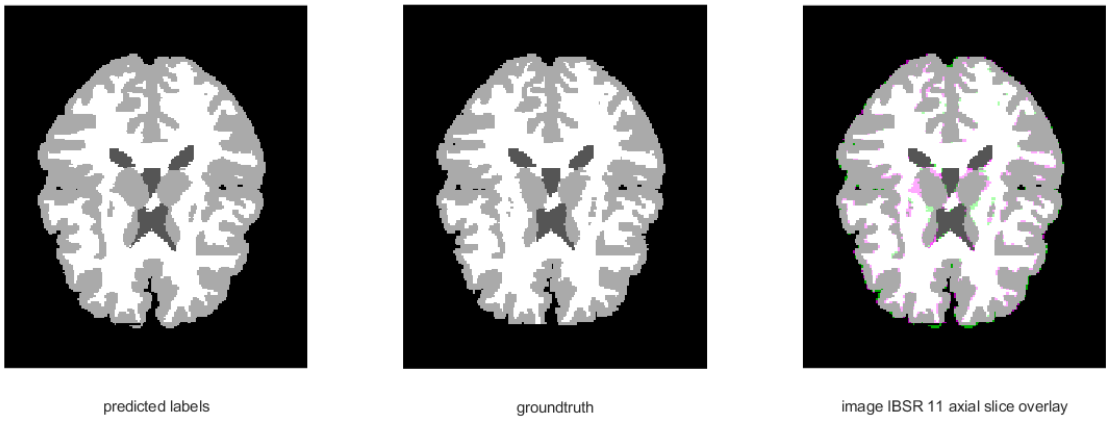


Figure 12: Qualitative results for validation volumes. Slices were selected near the middle to show all three tissues. volumes IBSR 11, 12, and 13. Pink pixels are coming from predicted labels, while, green pixels belong to the ground truth.

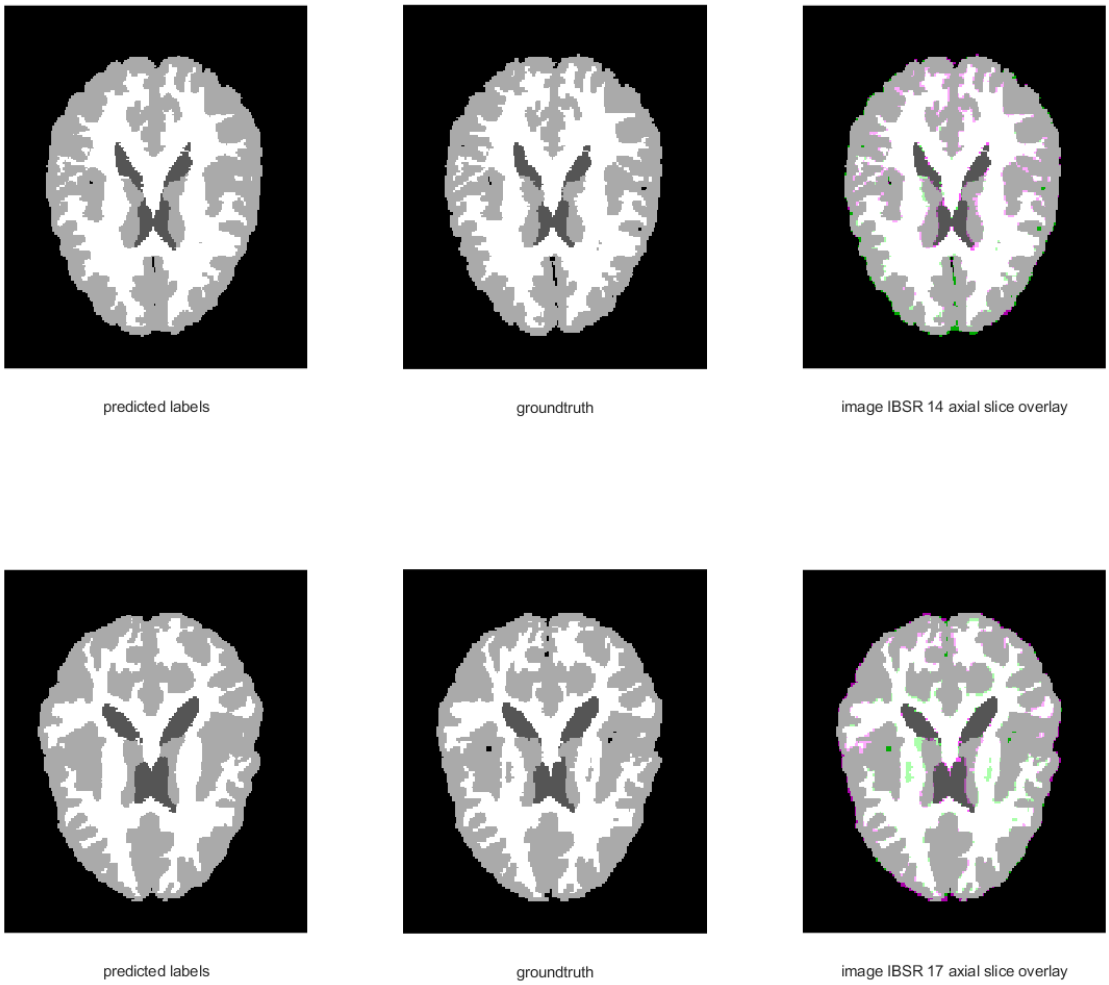


Figure 13: Qualitative results for validation volumes. Slices were selected near the middle to show all three tissues. volumes IBSR 14 and 17. Pink pixels are coming from predicted labels, while, green pixels belong to the ground truth.

References

[1] Lingraj Dora, Sanjay Agrawal, Rutuparna Panda, and Ajith Abraham. State-of-the-art methods for brain tissue segmentation: A review. *IEEE Reviews in Biomedical Engineering*, 10:235–249, 2017.

[2] Sudip Kumar Adhikari, Jamuna Kanta Sing, Dipak Kumar Basu, and Mita Nasipuri. Conditional spatial fuzzy c-means clustering algorithm for segmentation of mri images. *Appl. Soft Comput.*, 34:758–769, 2015.

[3] Lei Wen, Xingce Wang, Zhongke Wu, Mingquan Zhou, and Jesse S. Jin. A novel statistical cerebrovascular segmentation algorithm with particle swarm optimization. *Neurocomputing*, 148:569 – 577, 2015.

[4] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron C. Courville, Yoshua Bengio, Christopher Joseph Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.

[5] Martin Rajchl, Nick Pawlowski, Daniel Rueckert, Paul M. Matthews, and Ben Glocker. NeuroNet: Fast and Robust Reproduction of Multiple Brain Image Segmentation Pipelines. *arXiv e-prints*, page arXiv:1806.04224, June 2018.

[6] Ibsr dataset. `https://www.nitrc.org/forum/message.php?msg_id=12067.htm`. Accessed: 2019-01-05.

[7] Xiaofei Sun, Lin Shi, Yishan Luo, Wei Yang, Hongpeng Li, Peipeng Liang, Kuncheng Li, Vincent C. T. Mok, Winnie C. W. Chu, and Defeng Wang. Histogram-based normalization technique on human brain magnetic resonance images from different acquisitions. *BioMedical Engineering OnLine*, 14(1):73, Jul 2015.

[8] Eloy Roura, Nicolae Sarbu, Arnau Oliver, Sergi Valverde, Sandra González-Villà, Ricard Cervera, Núria Bargalló, and Xavier Lladó. Automated detection of lupus white matter lesions in mri. *Frontiers in Neuroinformatics*, 10:33, 2016.

[9] Jose Dolz, Christian Desrosiers, and Ismail Ben Ayed. 3d fully convolutional networks for subcortical segmentation in mri: A large-scale study. *NeuroImage*, 170:456 – 470, 2018. Segmenting the Brain.

[10] Mahsa Shakeri, Stavros Tsogkas, Enzo Ferrante, Sarah Lippe, Samuel Kadoury, Nikos Paragios, and Iasonas Kokkinos. Sub-cortical brain structure segmentation using F-CNN's. In *ISBI 2016: International Symposium on Biomedical Imaging*, Prague, Czech Republic, 2016.

[11] L. G. Nyul, J. K. Udupa, and Xuan Zhang. New variants of a method of mri scale standardization. *IEEE Transactions on Medical Imaging*, 19(2):143–150, Feb 2000.

[12] Arno Klein, Jesper Andersson, Babak A. Ardekani, John Ashburner, Brian Avants, Ming-Chang Chiang, Gary E. Christensen, D. Louis Collins, James Gee, Pierre Hellier, Joo Hyun Song, Mark Jenkinson, Claude Lepage, Daniel Rueckert, Paul Thompson, Tom Vercauteren, Roger P. Woods, J. John Mann, and Ramin V. Parsey. Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration. *NeuroImage*, 46(3):786 – 802, 2009.

[13] VS Fonov, AC Evans, RC McKinstry, CR Almli, and DL Collins. Unbiased nonlinear average age-appropriate brain templates from birth to adulthood. *NeuroImage*, 47:S102, 2009. Organization for Human Brain Mapping 2009 Annual Meeting.

[14] Ziv Yaniv, Bradley C. Lowekamp, Hans J. Johnson, and Richard Beare. Simpleitk image-analysis notebooks: a collaborative environment for education and reproducible research. *Journal of Digital Imaging*, 31(3):290–303, Jun 2018.

[15] J. A. Stark. Adaptive image contrast enhancement using generalizations of histogram equalization. *IEEE Transactions on Image Processing*, 9(5):889–896, May 2000.

[16] Dltk models. `https://github.com/DLTK/models/tree/master/ukbb_neuronet_brain_segmentation.htm`. Accessed: 2019-01-05.

[17] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[19] Nick Pawlowski, Sofia Ira Ktena, Matthew C. H. Lee, Bernhard Kainz, Daniel Rueckert, Ben Glocker, and Martin Rajchl. DLTK: state of the art reference implementations for deep learning on medical images. *CoRR*, abs/1711.06853, 2017.

[20] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models.

[21] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.

[22] Konstantinos Kamnitsas, Christian Ledig, Virginia F.J. Newcombe, Joanna P. Simpson, Andrew D. Kane, David K. Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical Image Analysis*, 36:61 – 78, 2017.

[23] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. *CoRR*, abs/1502.03240, 2015.

[24] Eli Gibson, Wenqi Li, Carole H. Sudre, Lucas Fidon, Dzoshkun Shakir, Guotai Wang, Zach Eaton-Rosen, Robert Gray, Tom Doel, Yipeng Hu, Tom Whyntie, Parashkev Nachev, Dean C. Barratt, Sébastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. Niftynet: a deep-learning platform for medical imaging. *CoRR*, abs/1709.03485, 2017.

[25] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467, 2016.

[26] Bradley Lowekamp, David Chen, Luis Ibanez, and Daniel Blezek. The design of simpleitk. *Frontiers in Neuroinformatics*, 7:45, 2013.