

Brain Tissue Segmentation Using NeuroNet With Different Pre-processing Techniques

Fakrul Islam Tushar, Basel Alyafi, Md. Kamrul Hasan, Lavsén Dahal

Erasmus+ Joint Master In Medical Imaging and Applications, University of Girona, Spain

Email: f.i.tushar.eee@gmail.com, basel931991@gmail.com, kamruleekuet@gmail.com, er.lavsén@gmail.com

Abstract—Automatic segmentation of brain Magnetic Resonance Imaging (MRI) images is one of the vital steps for quantitative analysis of brain for further inspection. In this paper, NeuroNet has been adopted to segment the brain tissues (white matter (WM), grey matter (GM) and cerebrospinal fluid (CSF)) 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 a robust trained model. The model has been trained and tested on IBSR18 data-set. To validate the research outcome, performance was measured quantitatively using Dice Similarity Coefficient (DSC) and is reported on average as 0.84 for CSF, 0.94 for GM, and 0.94 for WM. The outcome of the research indicates that for the IBSR18 data-set, pre-processing and proper tuning of hyper-parameters for NeuroNet model have improvement in DSC for the brain tissue segmentation.

Index Terms—Magnetic resonance imaging (MRI), Brain tissue segmentation, NeuroNet, Residual Network (ResNet), Fully Convolution Network (FCN), IBSR18, Dice Similarity Coefficient (DSC).

I. INTRODUCTION

Brain image segmentation plays a crucial role in brain image analysis which extracts brain tissues, WM, GM, and CSF from a brain image by partitioning it into a set of disjoint regions. Pixels inside each of those regions should be homogeneous in space and intensity [1]. Segmentation of brain tissue helps to detect diseases like brain tumor, Alzheimers (AD), Parkinsons, dementia, schizophrenia, and traumatic injury [2]. The performance of brain tissue segmentation methods depends on several factors such as regions location, size, shape, texture, and contrast, especially when using multiple acquisition modalities with different properties [3].

Traditional brain tissue segmentation approaches can be classified into region-based, clustering-based, statistical based methods [1], like gaussian mixture models and atlas, and classification methods where features are finely engineered to be classified eventually using Support Vector Machines, for instance. In the first three algorithms, pipelines are usually used where a failure in one step might dramatically affect the final result. On the other hand, the fourth class suffers from hard feature selection and integration. Recently, Convolutional Neural Networks (CNNs) gained recognition in brain tissue segmentation [4]. Here, end-to-end segmentation is done by

extracting features from images in an objective-oriented manner.

A recent and effective work by [5] has shown that Neuronet is a powerful network that can be used to segment raw brain images and outperformed previous tools. The extra feature is that this model can reproduce the output of multiple state-of-the-art tools simultaneously.

Our contribution in this paper is the use of two different pre-processing strategies followed by training with NeuroNet model to segment brain tissues for IBSR18 dataset and compare the performance with Dice Coefficient Similarity as a metric. These two pre-processing strategies are put under the microscope with different complexity levels, a simple one with standardization only against a more complicated one with registration, histogram equalization, then histogram matching. Then, a quantitative evaluation for the outcomes shows which strategy has been the most effective.

The remaining parts of the paper is organized as follows: Section II is dedicated to describe the dataset used, while, section III is used to explain the pre-processing pipelines. We illustrate the proposed method in section IV. We describe the implementation in section V and, experiment and results in section VI. We finally conclude in section VII.

II. DATASET

The dataset used for this work, IBSR18, is a publicly-available dataset by the Center for Morphometric Analysis at Massachusetts General Hospital [6]. The dataset is composed of 18 T1-W volumes with different slice thicknesses. The image volumes used are skull stripped and bias field corrected. For this work, the dataset was divided randomly into three subsets: ten for training, five for validation, and three for testing. A brief description of the dataset is given in Table I. The training subset was used for training the model, while, the validation subset was used to tune the proposed model. While, the test subset was eventually to evaluate the model. Fig. 1 shows the volume IBSR_01 and corresponding labels.

III. PRE-PROCESSING

The task of segmenting brain tissues become more challenging as different scanners are used with different parameters during acquisition. That usually leads to intensity heterogeneity, contrast variations, and different types of noise [7]. Therefore, data homogenization is often necessary.

TABLE I
SUMMARY ON IBSR18 DATASET USED IN THIS PAPER.

Training Subset		
Volume Name	Volume	Spacing (mm)
IBSR 01, 03, 04, 06	$256 \times 128 \times 256$	$0.94 \times 1.5 \times 0.94$
IBSR 07,08,09	$256 \times 128 \times 256$	$1 \times 1.5 \times 1$
IBSR 16,18	$256 \times 128 \times 256$	$0.84 \times 1.5 \times 0.84$
Validation Subset		
Volume Name	Volume	Spacing (mm)
IBSR 11, 12	$256 \times 128 \times 256$	$1 \times 1.5 \times 1$
IBSR 13,14	$256 \times 128 \times 256$	$0.94 \times 1.5 \times 0.94$
IBSR 17	$256 \times 128 \times 256$	$0.84 \times 1.5 \times 0.84$
Test Subset		
Volume Name	Volume	Spacing (mm)
IBSR 02	$256 \times 128 \times 256$	$0.94 \times 1.5 \times 0.9375$
IBSR 10	$256 \times 128 \times 256$	$1 \times 1.5 \times 1$
IBSR 15	$256 \times 128 \times 256$	$0.84 \times 1.5 \times 0.84$

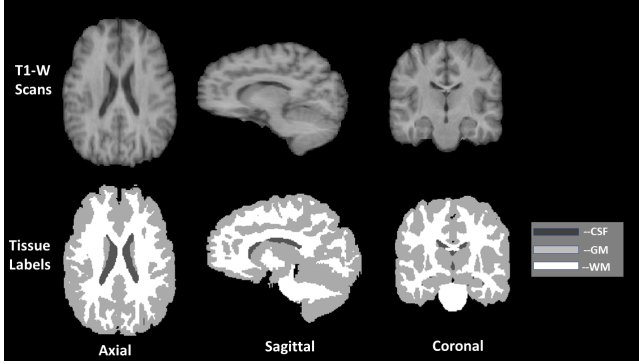


Fig. 1. Graphical description of IBSR_01 volume and corresponding ground truth labels in axial, sagittal and coronal view.

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 normalization for subcortical brain structure segmentation. 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. The idea is to apply a basic standardization pipeline first and compare the performance with a second registration-based pipeline. Fig. 2 shows an overview of both pipelines.

A. 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.

$$V_{new} = \frac{V_{old} - \mu}{\sigma} \quad (1)$$

where, μ, σ represent the mean and standard deviation of all voxels in the corresponding volume, respectively.

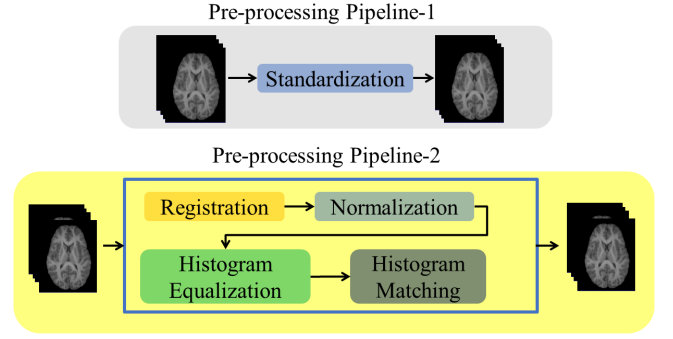


Fig. 2. Proposed pre-processing pipelines.

B. Pre-processing Pipeline-2

1) *Registration*: It is the alignment of two or more images in a common anatomical space [12], commonly called the fixed space. As shown in Table I, training, validation and test subsets have different spacing. To standardize the voxel spacing in the dataset, we registered the images and transformed the labels correspondingly to Montreal Neurological Institute (MNI) template (152 subjects, 1x1x1mm T1-w, dimensions: $182 \times 218 \times 182$, skull stripped) [13]. Fig. 3 shows the process of this registration. Simple-ITK framework in python was used for this purpose [14]. The idea is first to register the dataset to MNI template using rigid transformation, then save the corresponding final transformation matrix for transforming the labels. The available dataset was used as moving images, while, MNI template was used as the fixed image. After getting the final predicted labels, these should be brought back to the original space by applying the inverse of the corresponding saved transformation matrix.

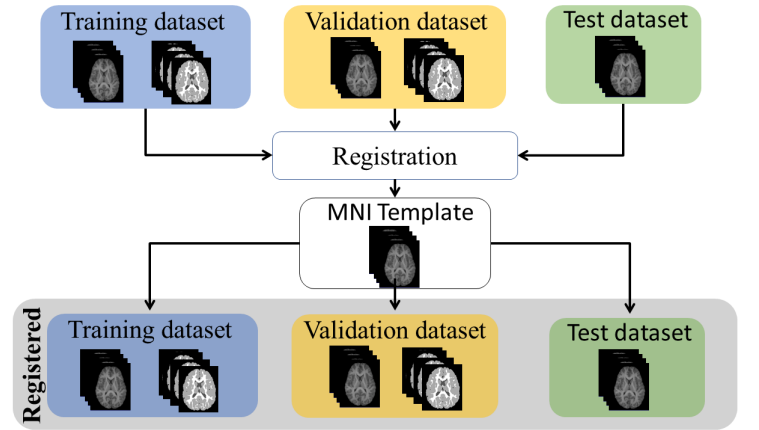


Fig. 3. Registering the whole dataset to MNI template to unify the voxel spacing.

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

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

3) *Histogram pre-processing*: In this step, the procedure was as follows:

- a) *Reference selection*: By analyzing tissues distribution for all volumes, image IBSR_07 was nominated due to the following reasons:
 - It has relatively wide spectrum of intensity values, see Fig. 4 left part.
 - The overlapping between GM and WM is acceptable, see Fig. 4 middle part.
 - GM and WM had comparable shares of voxels, this will help in the next step of histogram equalization of the reference image.
- b) *Histogram equalization of the 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].
- c) *Histogram matching to the reference*: apply histogram matching on 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. Fig. 5 shows 4 raw volumes and the corresponding pre-processed ones after applying pre-processing pipeline-2.

IV. METHOD

A. Network Architecture

In this paper, we adopted the NeuroNet [5] architecture for which the code is available at the Github repository of DLTK models [16]. NeuroNet is a deep convolutional neural network with multiple outputs, which was 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 the desired output is tissue segmentation only, the architecture was modified to an updated FCN architecture [17] with a ResNet encoder [18] as presented in [19]. Fig. 6 shows the original NeuroNet architecture and the adopted architecture in this work. One initial convolution was performed on input volumes, afterwards, features were extracted using the ResNet encoder [18] [19]. Features were extracted in encoder part with two residual units (N_{unit} =number of residual units at each scale), $U_i^{S_j} = \{U_1^{S_j}, U_2^{S_j}\}$ on each of the resolution scales $S_j = \{S_1, \dots, S_{N_{scales}}\}$, where N_{scales} is the number of resolutions scales. We used $N_{scales} = 4$, as in the default implementation [5]. Leaky ReLu (with leakiness = 0.1) was used as the activation function [20] with preceding batch normalization. At each scale, the down-sampling was performed using stride convolution [21] where the strides were $s_j = \{1, 2, 2, 2\}$ operating on each spatial dimension. As defined in the reference paper [5], the number of filters for

the convolutions in any U^{S_j} was double the number at the previous scale: 16, 32, 64 and 128.

In image segmentation, Fully Convolutional Networks (FCNs) are among the widely-used networks which typically reconstruct the prediction with the same size of the input given. In the original Neuronet architecture, the decoding part was based on multi-decoder architecture on FCN upscore operations [17]. The prediction was reconstructed 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 after a softmax layer. The loss was calculated using categorical cross-entropy loss for all prediction outputs \hat{y} at voxel locations v .

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

Where y, \hat{y} are the true, predicted label, respectively.

V. IMPLEMENTATION AND ENVIRONMENT

NeuroNet is implemented using Deep Learning Toolkit (DLTK) for Medical Image Analysis [19] on TensorFlow [22] with SimpleITK [23]. Pre-prcprocessing pipelines used in this work were implemented using DLTK and SimpleITK frameworks as well. The training was carried out on GeForce GTX 1080 GPU with a memory of 2.7GB.

VI. EXPERIMENTS AND RESULTS

Dice Similarity Coefficient (DSC) was used for evaluating the predicted labels and guiding the tuning process. Our main aim was to analyse the performance of the model with default parameters suggested in [5] with the proposed pre-processing pipelines and tune the parameters to improve the results. The corresponding results are shown in Table II.

Firstly, we trained our model with the processed data that went through only standardization (zero mean and unit standard deviation) see section III-A. 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 [24]. We used the pre-trained weights of Neuronet [5] as the initial ones to avoid gradient vanishing problem. We trained the model for 1000 steps with 200 randomly extracted patches of size $128 \times 128 \times 128$. The reason beyond choosing such a big patch size was that authors in [5] used it in the original implementation. Performance on the validation data is shown in Table II corresponding to model 1.1.

Secondly, the training steps were increased five times and the model was trained with double the number of patches (400 patches). That caused a huge improvement in the model's performance in CSF and a slight increase and decrease in WM and GM, respectively. Results are shown in Table II corresponding to model 1.2. Afterwards, more training was performed but no significant improvement was achieved, which led to looking for different pre-processing strategies.

Thirdly, the proposed pre-processing pipeline-2 was applied on the datasets explained in section III-B. The pipeline consists

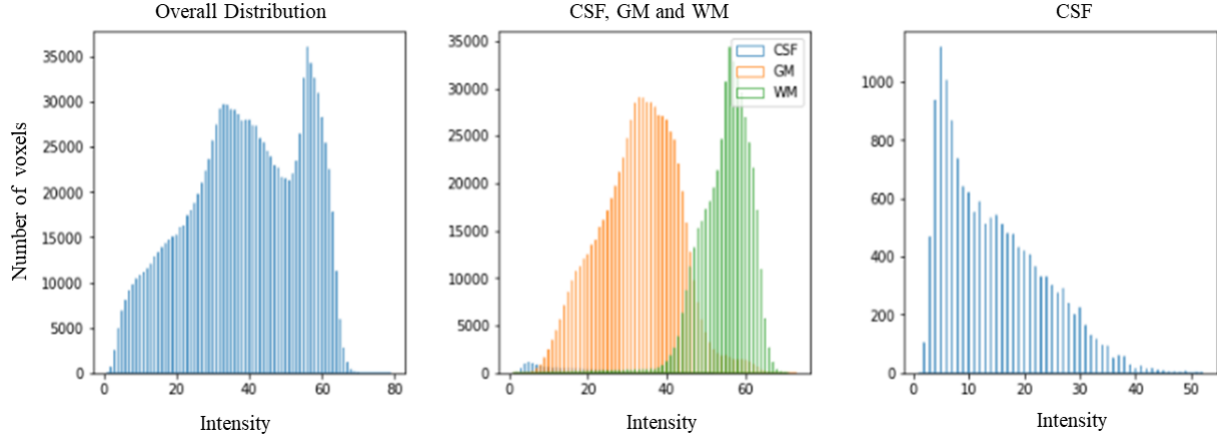


Fig. 4. IBSR_07 intensity distributions, where, on the left, the complete tissues intensity distribution, while, in the middle, CSF, white matter and GM distributions with different colours. The figure on the right is for CSF distribution only.

TABLE II
VALIDATION DSC OF DIFFERENT NETWORKS TRAINED WITH DIFFERENT HYPER-PARAMETERS. BOLD FONT HIGHLIGHTS THE HIGHEST DSC.

Model No.	#Training Steps	Model Configuration			Dice Coefficient Validation Set		
		Patch Size	Samples	Weights Initializations	CSF	GM	WM
1.1	1000	128x128x128	200	NeuroNet Pretrained	0.43±0.40	0.90±0.01	0.88±0.06
1.2	5000	128x128x128	400	NeuroNet Pretrained	0.80±0.12	0.89±0.05	0.89±0.03
2.1	4000	32x32x32	200	Uniform Distribution	0.07±0.03	0.71±0.04	0.72±0.06
2.2	4000	64x64x64	200	Uniform Distribution	0.80±0.05	0.90±0.02	0.89±0.03
2.3	4000	128x128x128	50	Uniform Distribution	0.89±0.02	0.93±0.01	0.93±0.01
2.4	2000	128x128x128	50	NeuroNet Pretrained	0.89±0.02	0.94±0.01	0.93±0.01
2.5	4000	128x128x128	50	NeuroNet Pretrained	0.90±0.02	0.94±0.01	0.93±0.02

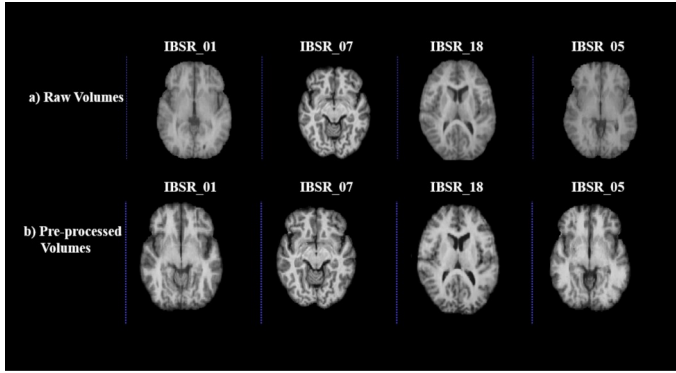


Fig. 5. The effect of pre-processing the dataset. At the top, 4 cases before applying the pre-processing pipelines. At the bottom, the final pre-processed dataset.

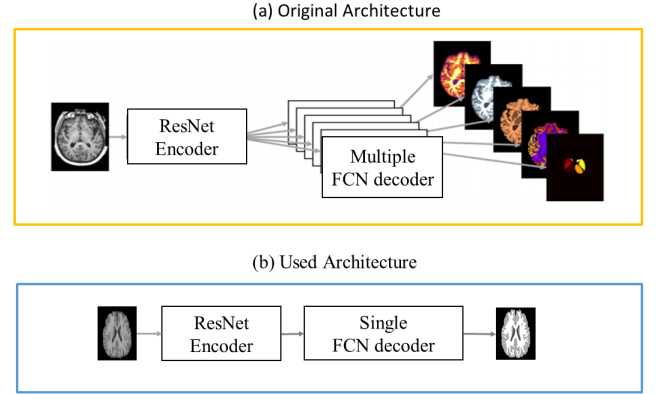


Fig. 6. Network Architectures [5].

of registration, normalization, adaptive histogram equalization and histogram matching steps. We tried three different cases using pipeline-2 for pre-processing, these cases were as follows:

- Case 1 : patch size: 32x32x32, 200 samples.
- Case 2 : patch size: 64x64x64, 200 samples.
- Case 3 : patch size: 128x128x128, 50 samples.

Table II, model 2.1, 2.2 and 2.3 shows Case 1, Case 2 and Case 3 results respectively. Model 2.1, 2.2 and 2.3 were trained

from end to end using the uniform distribution as weight initialization. Model 2.1, which was trained on patches of size 32x32x32, performed poorly and failed to segment CSF region. Training the model with bigger patch sizes turned out to be better as used in the original implementation. Overall, this model, 2.5, proved to have the best combination of parameters compared to all other experiments shown in Table II. Finally, we performed the test on the test set using the best model (2.5) and achieved average DSC 0.84 for CSF, 0.94 for GM and 0.94 for WM.

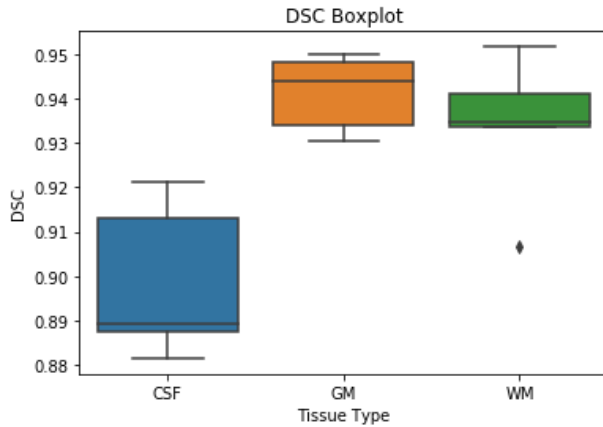


Fig. 7. DSC of best performing model 2.5 on the validation dataset.

VII. CONCLUSION

In this work, we proposed using NeuroNet architecture for 3D brain tissue segmentation and investigated the performance of the CNN under different hyper-parameters and preprocessing techniques. To the best of our knowledge, this study is the first effort to use Neuronet with IBSR 2018 dataset by the time of this work. Our results show that histogram preprocessing was clearly effective in boosting the performance. Registration assisted mainly in unifying the voxel spacing to make all the inputs have similar spatial spaces. The use of NeuroNet pre-trained weights helped significantly 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 in [5]). 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. Future work includes extending the analysis with 3D Unet and modified ResNet-Unet architectures. We also plan to ensemble different models to improve the DSC for CSF especially. We also plan to extend the use of the same CNN in disease segmentation.

REFERENCES

- [1] L. Dora, S. Agrawal, R. Panda, and A. Abraham. State-of-the-art methods for brain tissue segmentation: A review. *IEEE Reviews in Biomedical Engineering*, 10:235–249, 2017.
- [2] S. K. Adhikari, J. K. Sing, D. K. Basu, and M. Nasipuri. Conditional spatial fuzzy c-means clustering algorithm for segmentation of mri images. *Appl. Soft Comput.*, 34:758–769, 2015.
- [3] L. Wen, X. Wang, Z. Wu, M. Zhou, and J. S. Jin. A novel statistical cerebrovascular segmentation algorithm with particle swarm optimization. *Neurocomputing*, 148:569 – 577, 2015.
- [4] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. C. Courville, Y. Bengio, C. J. Pal, P. Jodoin, and H. Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.
- [5] M. Rajchl, N. Pawłowski, D. Rueckert, P. M. Matthews, and B. Glocker. NeuroNet: Fast and Robust Reproduction of Multiple Brain Image Segmentation Pipelines. *arXiv e-prints*, 2018.
- [6] IBSR-18 Dataset. Available at: https://www.nitrc.org/frs/?group_id=48. Accessed: 2019-01-05.
- [7] X. Sun, L. Shi, Y. Luo, W. Yang, H. Li, P. Liang, K. Li, V. C. T. Mok, W. C. W. Chu, and D. Wang. Histogram-based normalization technique on human brain magnetic resonance images from different acquisitions. *BioMedical Engineering OnLine*, 14(1):73, 2015.
- [8] E. Roura, N. Sarbu, A. Oliver, S. Valverde, S. Gonzalez-Vill, R. Cervera, N. Bargall, and X. Llado. Automated detection of lupus white matter lesions in mri. *Frontiers in Neuroinformatics*, 10:33, 2016.
- [9] J. Dolz, C. Desrosiers, and I. B. Ayed. 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study. *NeuroImage*, 170:456 – 470, 2018.
- [10] M. Shakeri, S. Tsogkas, E. Ferrante, S. Lippe, S. Kadoury, N. Paragios, and I. 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, 2000.
- [12] A. Klein, J. Andersson, B. A. Ardekani, J. Ashburner, B. Avants, M. Chiang, G. E. Christensen, D. L. Collins, J. Gee, P. Hellier, J. H. Song, M. Jenkinson, C. Lepage, D. Rueckert, P. Thompson, T. Vercauteren, R. P. Woods, J. J. Mann, and R. V. Parsey. Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration. *NeuroImage*, 46(3):786 – 802, 2009.
- [13] V. Fonov, A. Evans, R. McKinstry, C. R. Almli, and D. L. Collins. Unbiased nonlinear average age-appropriate brain templates from birth to adulthood. *NeuroImage*, 47:S102, 2009.
- [14] Z. Yaniv, B. C. Lowekamp, H. J. Johnson, and R. Beare. Simpleitk image-analysis notebooks: a collaborative environment for education and reproducible research. *Journal of Digital Imaging*, 31(3):290–303, 2018.
- [15] J. A. Stark. Adaptive image contrast enhancement using generalizations of histogram equalization. *IEEE Transactions on Image Processing*, 9(5):889–896, 2000.
- [16] DLTK models. Available at: https://github.com/DLTK/models/tree/master/ukbb_neuro_net_brain_segmentation.htm. Accessed: 2019-01-05.
- [17] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.
- [18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [19] N. Pawłowski, S. I. Ktena, M. C. H. Lee, B. Kainz, D. Rueckert, B. Glocker, and M. Rajchl. DLTK: state of the art reference implementations for deep learning on medical images. *CoRR*, abs/1711.06853, 2017.
- [20] A. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models. *Computer Science Department, Stanford University, CA 94305 USA*.
- [21] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.
- [22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Zefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467, 2016.
- [23] B. Lowekamp, D. Chen, L. Ibanez, and D. Blezek. The design of simpleitk. *Frontiers in Neuroinformatics*, 7:45, 2013.
- [24] K. Kamnitsas, C. Ledig, F. J. V. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical Image Analysis*, 36:61 – 78, 2017.