Magnetic Resonance Imaging 64 (2019) 21–27

Contents lists available at ScienceDirect

Magnetic Resonance Imaging

journal homepage: www.elsevier.com/locate/mri

Original contribution

# Open-source pipeline for multi-class segmentation of the spinal cord with deep learning

François Paugam[a,b], Jennifer Lefeuvre[c], Christian S. Perone[b], Charley Gros[b], Daniel S. Reich[c], Pascal Sati[c], Julien Cohen-Adad[b,d,*]

[a] *École Centrale de Lyon, Lyon, France*
[b] *NeuroPoly Lab, Institute of Biomedical Engineering, Polytechnique Montreal, Montreal, QC, Canada*
[c] *Translational Neuroradiology Section, National Institute of Neurological Disorders and Stroke, National Institutes of Health, Bethesda, MD, USA*
[d] *Functional Neuroimaging Unit, CRIUGM, Université de Montréal, Montreal, QC, Canada*

ARTICLE INFO

*Keywords:*
MRI
Segmentation
Deep learning
U-Net
cnn
Spinal cord
Marmoset

ABSTRACT

This paper presents an open-source pipeline to train neural networks to segment structures of interest from MRI data. The pipeline is tailored towards homogeneous datasets and requires relatively low amounts of manual segmentations (few dozen, or less depending on the homogeneity of the dataset). Two use-case scenarios for segmenting the spinal cord white and grey matter are presented: one in marmosets with variable numbers of lesions, and the other in the publicly available human grey matter segmentation challenge [1]. The pipeline is freely available at: https://github.com/neuropoly/multiclass-segmentation.

## 1. Introduction

Medical image segmentation consists in identifying which voxels belong to a specific structure. Such structures can be pathological (e.g., lesions) or normal (e.g., white or grey matter). Segmentations are useful to extract relevant quantitative information about the structure, such as its size, orientation, shape, and location. However, manual segmentation is a long and tedious process.

Fortunately, recent advances in deep learning techniques using convolutional neural networks have shown positive results in executing complex tasks automatically, including image segmentation [2]. Yet, while many papers applying such techniques to medical image segmentation have been published [2], most models presented as usable off-the-shelf have been validated in well-curated single-center datasets only. In the rare case where the code is publicly available, the algorithm usually fails when applied to other centers [3]. This happens because images across different centers have slightly different features than those used to train the algorithm (contrast, resolution, etc.), combined with the fact that low amounts of data and manual labels are available. Recent deep learning techniques, such as domain adaptation [4], have tackled this issue. Yet, these techniques are not well adapted to MRI segmentation, because in MRI, image features not only vary between centers, but also across a large number of acquisition parameters (e.g., repetition time, flip angle). Hence, there is a need for readily available frameworks that enable researchers to easily train new models for specific MRI segmentation applications and their specific data.

This paper presents an open-source pipeline to execute single-class or multi-class segmentation of MRI volumes using supervised neural networks. This pipeline is specifically designed for homogeneous datasets (i.e., same acquisition parameters). A typical use-case scenario consists in: segmenting a few slices, training the network and segmenting the rest of the images. This pipeline is in line with recent initiatives for machine learning frameworks applied to medical data [5–7]. Our implementation is simple and modular, allowing to modify the type of architecture and useful parameters such as data augmentation. This pipeline is particularly useful for training new models from scratch, where pre-trained models for fine tuning are not available.

The paper is structured as follows: first, the structure and algorithm of the pipeline are described. Then, two use-case scenarios are demonstrated using this pipeline for segmenting the spinal cord white and grey matter. In the first scenario, we used 7 T data from marmoset monkeys with experimental autoimmune encephalomyelitis (EAE), exhibiting highly contrasted lesions. In the second scenario, we used 3 T data from the publicly-available healthy human spinal cord grey matter challenge [1]. Results are then presented and discussed.

[*] Corresponding author at: École Centrale de Lyon, Lyon, France.
*E-mail addresses:* francois.paugam@laposte.net (F. Paugam), jcohen@polymtl.ca (J. Cohen-Adad).

https://doi.org/10.1016/j.mri.2019.04.009
Received 4 January 2019; Received in revised form 15 April 2019; Accepted 17 April 2019

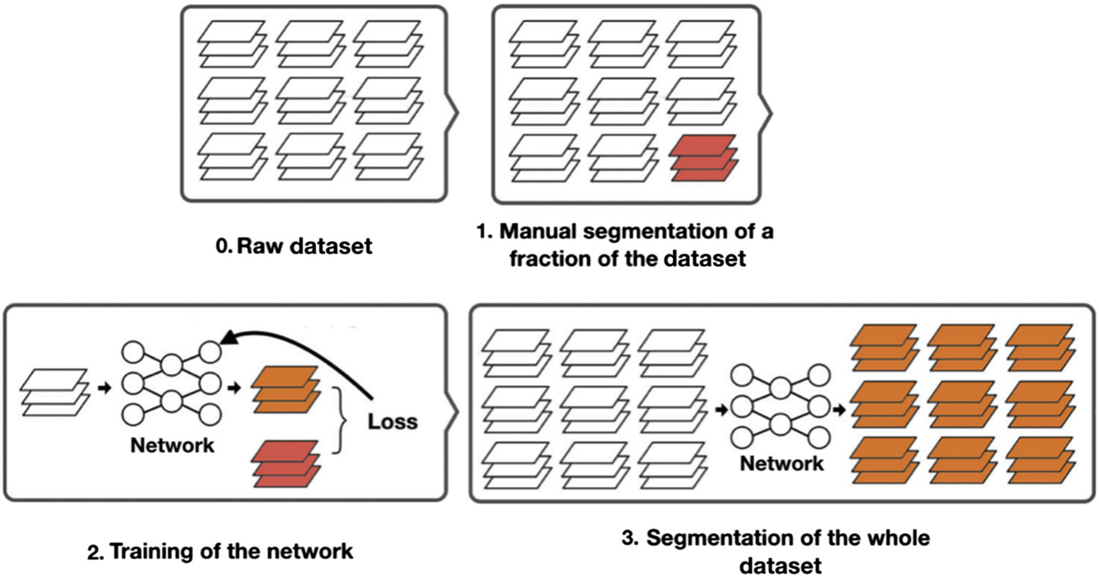
0. Raw dataset
1. Manual segmentation of a fraction of the dataset
Network
Loss
2. Training of the network
Network
3. Segmentation of the whole dataset

**Fig. 1.** Overview of the pipeline to segment a dataset. The data (represented as stacks) are volumes sliced into images along the vertical axis. Red data are manual segmentations, orange data are the network's segmentations. Step 0: the database is not segmented. Step 1: the user manually segments a fraction of the dataset. Step 2: the network is trained on the manually-segmented data with a gradient descent based algorithm. Step 3: Once the network is trained, it can be used to segment the entire dataset. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

## 2. Material and method

Fig. 1 illustrates the main processes of the pipeline: (i) manual segmentation of a fraction of the dataset, (ii) training the network, and (iii) automatic segmentation of the whole dataset. The pipeline was developed in python 2.7 and uses PyTorch [8].

The neural network learns to segment the structures through a supervised process. The neural network is presented with images and their corresponding manual segmentations, produces segmentations (inference), computes the error (loss) between the output segmentations and the input segmentations, and then corrects its parameters using the gradient backpropagation method [9] on the computed error.

### 2.1. Data pre-processing

The training process uses NIfTI files as inputs and returns a PyTorch network. The output network file can then be used for inference (i.e. to segment new data unseen by the network during training).

During the training, the network is presented with MRI volumes and associated manual segmentations (ground truth). The first step consists of splitting the NIfTI files dataset of the MRIs and their corresponding ground truth into three datasets: training, validation, and evaluation. The distribution of these datasets heavily depends on the initial database and its variability. Based on the literature [10] and preliminary experiments, we opted for 70/15/15% of the labelled data for training/validation/evaluation. The training dataset should contain a representative sample of the database (i.e. featuring a balanced variety of spinal levels, subjects, pathology load), whereas for the evaluation dataset, it can be preferable to use data from a specific group of the database, e.g. volumes from a specific subject who was not included in the training dataset.

For each dataset, NIfTI files are pre-processed and randomly dispatched into normalized batches. As shown in Fig. 2, the 3D volumes are sliced into 2D axial images along the superior-inferior direction. Although there are some neural network architectures that take advantage of 3D data and segment voxels in volumes instead of pixels on images, these networks require more computing power and thus more time for training to give results that are not necessarily better than when using 2D networks [11]. Therefore, to avoid high computational costs, we chose to work with 2D networks.

### 2.2. Training and evaluation

Several architectures can be chosen for the network: U-Net [12], SegNet [13] or NoPoolASPP [14]. These architectures have been chosen for their good performance on segmentation tasks, in particular the well-known U-Net and the NoPoolASPP, which demonstrated state-of-the-art results on the spinal cord grey matter segmentation challenge [14]. They have been optimized for small-data regime. In brief, each architecture uses the Adam optimiser [15] with the Dice loss generalised to multi-class [16] for loss function. The Dice loss has the dual advantages of describing surface similarity well and being minimally sensitive to intra-class unbalance [17]. Regularization is performed with batch-normalisation [18] and dropout [19]. Data augmentation includes rotation, elastic transformation, scaling, and vertical symmetry to simulate more anatomical and positional diversity, and channel shift to simulate more variability in the acquisition contrast. These transformations are illustrated in Fig. 3.

While the main criteria to evaluate the network is the loss value, other metrics are used to quantify other aspects of the network's performance, such as the Hausdorff surface distance (HSD) to evaluate if the predicted segmented surface is far from the ground truth, or the Jaccard index (JI) to evaluate the overlap between the prediction and the ground truth. Qualitative evaluations of the output segmentations are also important and systematically performed. They take the form of informal visual examinations of the predicted segmentation. These examinations focus on the shape and location of the segmentations, their connectedness, and the smoothness of their contours.

## 3. Segmentation of white and grey matter in typical use-case scenarios

### 3.1. Marmoset spinal cord with EAE lesions

The framework presented in section 1 was used to segment the normal appearing white matter (NAWM) and normal appearing grey matter (NAGM) on MRIs of the spinal cord of marmosets with EAE. These represent the portions of the spinal cord that do not contain focal

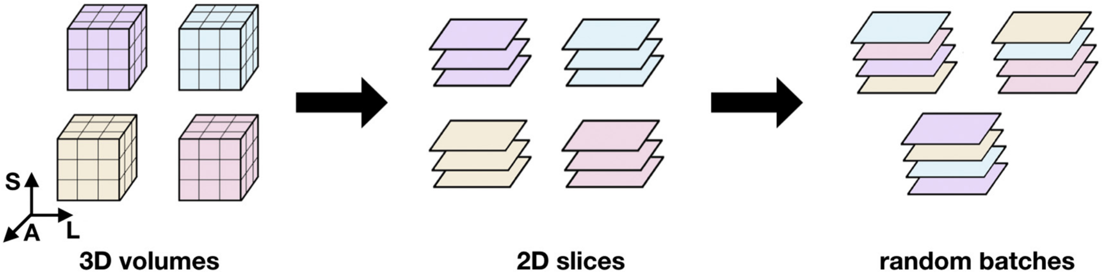

**Fig. 2.** Constitution of batches. S: superior, A: anterior, L: left.

inflammatory or post-inflammatory lesions.

### 3.1.1. Data

The entire spinal cords of three adult marmosets were scanned in vivo on a 7 T preclinical MRI system (horizontal magnet, BioSpec 7 T/30 cm USR, Bruker Biospin) using a birdcage volume coil as a transmitter and a custom-built 12-element phased-array receiver-only [20]. A 2D RARE (Rapid Acquisition with Relaxation Enhancement) PD-weighted sequence was acquired axially with the following MRI parameters: TR/TE = 3500/15 ms, RARE factor = 2, number of averages = 4, phase encoding = row direction. The resolution in plane was 0.15 × 0.15 $\text{mm}^2$ with a 0.8 mm slice thickness. A total of 20 slices was acquired with a 1.5 mm gap thickness between adjacent slices. A respiratory gating was applied (Bruker, trigger module) in order to synchronize the data acquisition with the breathing cycle. A faster sagittal PD-w RARE was acquired to guide the position of the axial slices. This step was important to maintain consistent slice position between MRI sessions. Examples of images are shown in Fig. 4. Due to the sensitivity profile of the birdcage transmit volume coil and the length of the marmoset spine, the imaging session was divided into three separate segments by repositioning the segment of interest at the isocenter of the transmit volume coil. For each animal, we always obtained one baseline MRI before disease induction (same EAE immunization protocol from [21]) and several follow-up MRIs until the end of the experiment based on the severity of the symptoms (based on the ACUC guidelines when the animals reached paraplegia). Three MRI sessions were performed on two animals over a four-week and eight-week period. 10 MRI sessions were performed for the third animal over a 15-week period. A total of 16 volumes of the entire spinal cord were obtained.

The database is composed of 16 volumes, or 946 axial slices. 140 slices were manually segmented by a trained specialist (JL) for focal (discrete, round/oval well-delineated white matter lesions) and subpial (abnormal WM signal along the cord edge) lesions, previously characterized in a MRI postmortem study [22]. Those slices were selected empirically, aiming to represent the variability of the dataset well in terms of vertebral level, lesion load and image quality (SNR, intensity bias). The number of slices (140) was empirically incremented: we first started with less slices, and increased the number until we obtained satisfactory results. Several binary masks were obtained by a semi-automatic intensity-based segmentation using the software Jim (Xinapse Systems) for grey matter, focal lesions, and the cross sectional area of the normal appearing cord.

A first model was trained to segment the NAWM and the NAGM on data coming mostly from the early time points ($t_0$: 38%, $t_1$: 19%, $t_2$: 13%, $t_9$: 30%), which are the least affected by the disease. For this model, the training dataset was composed of 79 images, the validation dataset of 17 images, and the evaluation dataset of 44. Images from marmosets 1 and 2 were used for the training and validation datasets, and from marmosets 2 and 3 for the evaluation dataset. A second model was trained to segment the same structures and with the same hyperparameters, validation dataset and evaluation dataset, but using all 140

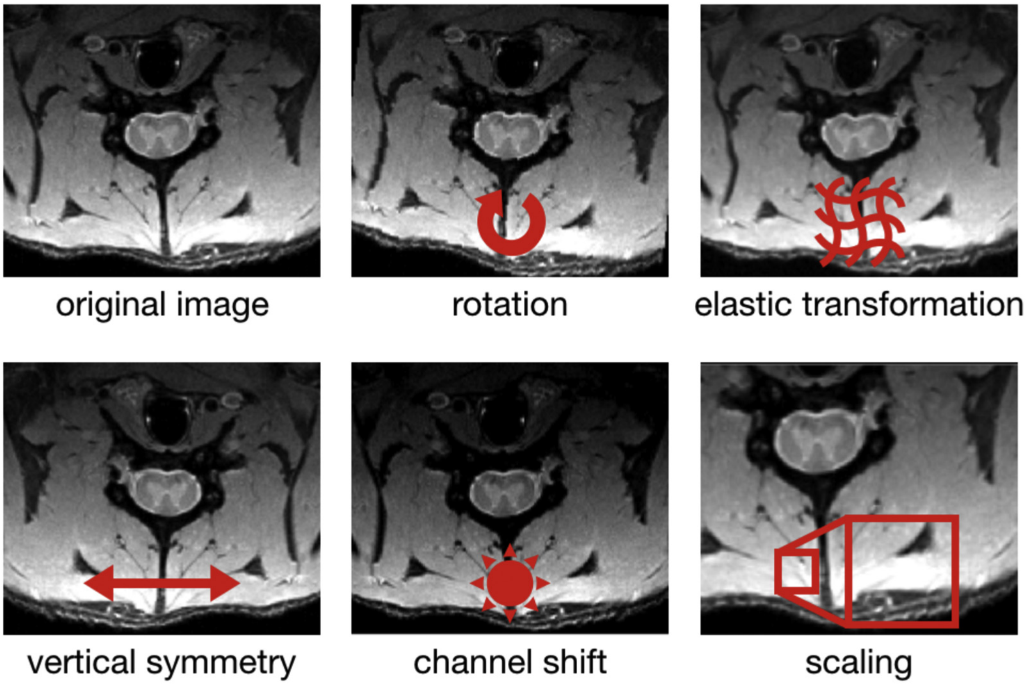

**Fig. 3.** Illustration of the possible transformations for data augmentation during training.

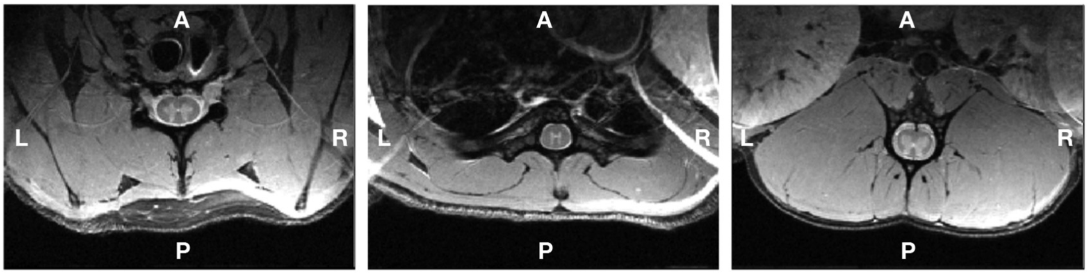

**Fig. 4.** Examples of axial PD-weighted MRI slices from healthy marmosets (before EAE). The spinal cord appears in the center. The grey matter exhibits an “H” pattern and is surrounded by the white matter (as opposed to the brain). From left to right: cervical, thoracic and lumbar levels.

images and ground truth for the training dataset. The first model was used to evaluate the method and hyper-parameters, and in particular to confirm there was no overfitting. The second model used the method and hyper-parameters validated by the first model, but since it was trained with more data, it produced better results.

### *3.1.2. Network and hyper-parameters*

The architecture used for the network is a variation of the original U-Net [12] with fewer layers and weights, as shown in Fig. 5. Using a smaller network makes the training faster.

The following hyper-parameters were used:

**Cropping**: All the slices were center-cropped to a rectangle of 160 × 160 pixels. This choice was a compromise between having a small image size for the patch (better computation performance) and having enough information to train the model. Hence, we made sure to at least include the cord, the spine and a reasonable portion of surrounding tissue. The size of the patch is illustrated in Fig. 3.

**Normalisation**: Mean centering and standard deviation normalisation of the pixel intensities. This is a common procedure for segmentation tasks;

**Batch size**: 11 samples. This parameter results in a compromise between the representativeness of the global dataset within each batch (large batches) versus the capability of exploring more suitable solutions by the network, at the detriment of converging slower (small batches). This parameter was chosen through a hyper-parameter optimization, where possible values ranged from 5 to 50.

**Optimization**: Adam optimizer [15] with a learning rate $\eta = 0.001$. This parameter impacts the speed and precision of convergence. It was chosen through hyper-parameter optimization, where possible values ranged from $10^{-9}$ to 0.1.

**Dropout**: Rate of 0.3. This parameter helps regularize the network and prevent overfitting. This value was based on the literature and our preliminary experience.

**Batch-normalisation**: Momentum of 0.1. This value was based on the literature and our preliminary experience.

**Iterations**: 10,000 epochs, with 8 batches at each epoch for the first and 13 for the second model. The number of epochs could be stopped automatically after reaching a Dice loss value, however we preferred to let the network continue training and regularly inspect the output segmentation through TensorBoard. For example, after about 3000 epochs, the Dice loss started to plateau, however we decided to continue the training, as we noted that the appearance of the segmentation was more regularized (based on qualitative evaluations, not captured by metrics).

**Data augmentation** (see Fig. 3): Right-left flipping (50% rate), scaling [0.5, 1], rotation [−20°, +20°], channel shift [−20%, +20%] of the max signal intensity (i.e., 1, after normalisation) and elastic deformation (30% rate, Gaussian displacement field of sigma = [3, 4.5] and amplitude= [8,17]). These values were based on our preliminary experience.

The complete list of the hyper-parameters and their value are available at https://github.com/neuropoly/multiclass-segmentation/blob/master/parameters_template.json.

Training on a single NVIDIA Tesla P100-SXM2 took approximately 8 h (10,000 epochs).

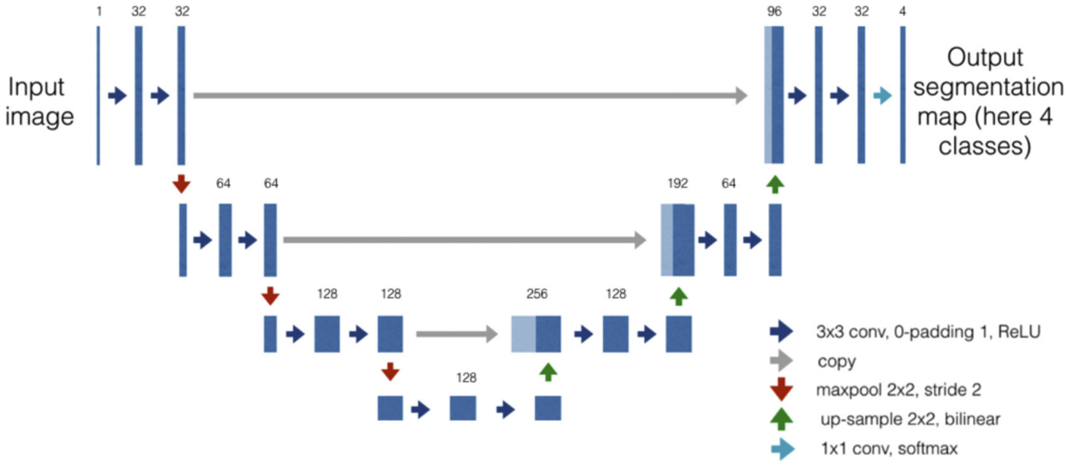

**Fig. 5.** Architecture used for the network, based on the U-Net architecture. Each dark blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. Light blue boxes represent copied feature maps. The arrows denote the different operations. The idea is to combine convolutions and down-sampling operations in the contracting path (left part) to extract high-level features (bottom). Then the high-level features are up-sampled and combined with the lower-level features (light blue boxes) through convolutions in the expanding path (right part). Combining high-level and low-level features corresponds to combining global and local information. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Table 1**
Comparison of the two models trained on the same evaluation dataset with the following metrics: precision, recall, Dice similarity coefficient (DSC), Jaccard index (JI), Hausdorff surface distance (HSD) and mean surface distance (MSD). The metrics are computed with respect to the manual segmentations. The best results are shown in bold font. Here HSD and MSD metrics are in pixels.

| | First model | | Second model | |
|---|---|---|---|---|
| | NAWM | NAGM | NAWM | NAGM |
| Precision (*Best*: *1*, *Worst*: *0*) | 0.848 | **0.956** | **0.910** | 0.952 |
| Recall (*Best*: *1*, *Worst*: *0*) | **0.957** | 0.852 | 0.955 | **0.934** |
| DSC (*Best*: *1*, *Worst*: *0*) | 0.892 | 0.896 | **0.923** | **0.940** |
| JI (*Best*: *1*, *Worst*: *0*) | 0.812 | 0.815 | **0.876** | **0.889** |
| HSD (*Best*: *0*) | 1.883 | 1.571 | **1.631** | **1.319** |
| MSD (*Best*: *0*) | 0.119 | 0.110 | **0.106** | **0.067** |

### *3.1.3. Results*

Table 1 lists resulting scores for the first and second models. The first model already gives satisfactory results on the evaluation dataset (i.e., unseen during training), suggesting a good likelihood of successful generalization with new data. Yet, we can see in Fig. 6 that the first model produces false positives of NAWM segmentation. This first model's results are good enough to validate the chosen hyper-parameters. However, the presence of false positives motivated us to increase the generalization of the network by training it with more data, which is done with the second model.

The second model presents even better results for almost all metrics. However, since the images of the evaluation dataset were also used for training, these scores are somewhat less meaningful. Hence, we resorted to a visual evaluation of the segmentations' aspect on data that was not used during both trainings (see Fig. 7). Since these data were not manually segmented, computing metrics was not possible. Overall, the second model presents fewer false positives, suggesting that it performs better than the first model.

While the Best/Worst results are listed in the table for easier interpretation, it should be noted that the relevance of the overlap (DSC, JI) and distance (HSD, MSD) metrics are relative to the size/shape of the object and the native resolution of the image. For example, if objects are small (i.e. few voxels) a difference of one voxel between them will have disproportionate effect on the DSC, as also pointed out in [23,24]. Therefore, those metrics are best interpreted when compared between different conditions, as opposed to being considered in their absolute sense.

## *3.2. Human spinal cord grey matter challenge*

To demonstrate the applicability of the method in another scenario, we used data from the publicly-available spinal cord grey matter segmentation challenge [1].

### *3.2.1. Data and training*

The dataset consists of 4 sites with 20 healthy subjects each (80 subjects in total). The MRI pulse sequence was a multi-echo gradient echo, with voxel size ranging from 0.25 × 0.25 × 2.5 mm to 0.5 × 0.5 × 5.0 mm. The dataset was split between training (40, 10 per site) and test (40, 10 per site). The test set's manual segmentations were hidden from us.

The challenge data includes ground truth segmentations of GM and WM generated by four independent raters. Here, we used the labeling obtained with the majority vote of these four labelings. As our framework is intended to be used on homogeneous datasets, we trained independently four models, one for each site. The models were trained on all the training volumes, which represents 30, 113, 274 and 134 images for the sites 1, 2, 3 and 4 respectively. For training, we used the same hyper-parameters as the ones used for training the marmoset model, except for the cropping size which depends on the original size and resolution of the images. The cropping size was [80 × 80] for sites 1 and 2, and [160 × 160] for sites 3 and 4.

Trainings on a GeForce GTX 1070 took about 1, 3, 10 and 6 h respectively for sites 1, 2, 3 and 4.

We also trained one model using the data from all sites to compare the results. This model used the same hyper-parameters, with a [80 × 80] cropping size. The training of this model took about 6 h on a GeForce GTX 1070.

### *3.2.2. Results*

Although we trained our models to segment both the WM and GM, the spinal cord grey matter segmentation challenge platform only accepts submission for the GM segmentations. Table 2 shows a comparison of existing segmentation methods and our method (last column) and demonstrate best performance of our method in 6 out of 10 metrics. It should be stressed however that our best performing method uses one model per site while other methods use single training from all aggregated data across sites.

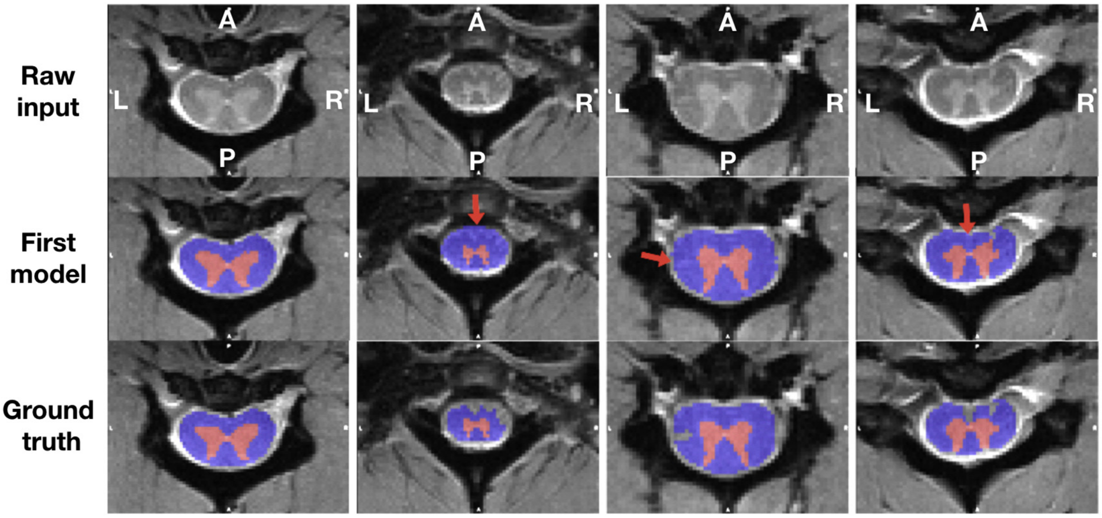

**Fig. 6.** Comparison between the first model and the ground truth. Blue: NAWM, red: NAGM. False positives of NAWM are pointed by red arrows. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

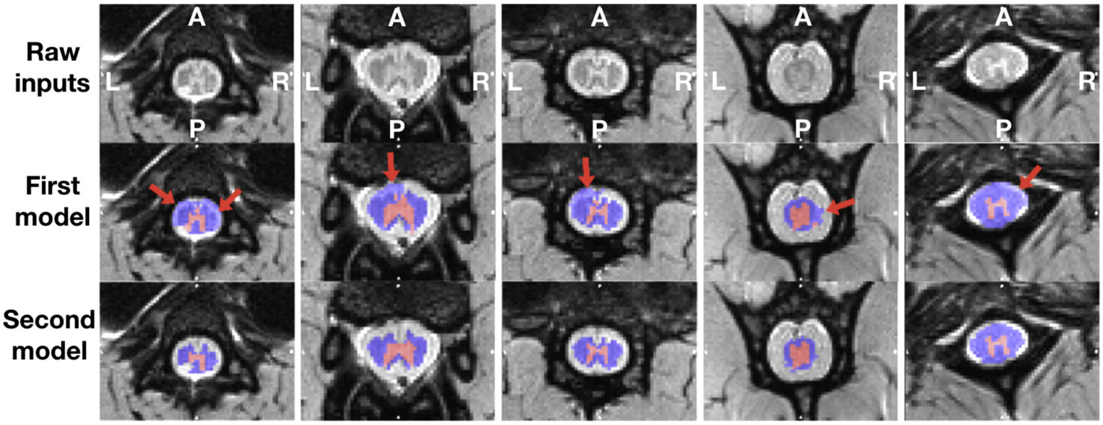

**Fig. 7.** Comparison between the two models trained. Blue: NAWM, red: NAGM. False positives of NAWM are pointed by red arrows. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

## 4. Discussion and conclusion

In this study, we presented an open-source pipeline to perform single-class or multi-class segmentation of MRI volumes using supervised neural networks. This pipeline is specifically designed for small and homogeneous datasets, hence it is particularly adapted to single-site datasets acquired with the same acquisition protocol. Two typical use-case scenarios were presented for segmentation of the grey and white matter in the spinal cord.

The first scenario was based on 7 T MRI data from marmoset model of EAE, exhibiting variable numbers of highly-contrasted lesions in the spinal cord. Here, the presented framework enabled us to segment almost 1000 images based on 140 manual segmentations. While the effort is largely minimized thanks to this automatic framework (see Table 1), the exploitation of the output predictions in a medical context still requires the overview and manual corrections of an expert. This case-scenario was quite conservative because the varying appearance of lesions across time points made it very difficult for the network to generalize well, hence a relatively high number of ground truth slices (140) was required. In similar data without lesions, we estimate that less than 5% of ground truth would be necessary [14].

The second scenario was based on the publicly-available dataset of the human spinal cord grey matter segmentation challenge [1]. Here, we demonstrated that the underlying strategy of the framework, consisting of training with less data but from homogeneous datasets, can produce better results than models trained on larger and more heterogeneous datasets. It should be stressed, however, that our architecture might not be the best in absolute terms.

Indeed when we use data from all sites in a single training our model show worse results, with sensibly higher standard deviations of the scores, which indicates far less consistent performances.

Not explored in this paper but worth mentioning, is that transfer learning techniques from traditional image classification/segmentation tasks are expected to greatly improve the performance of the network. This was not investigated here because the purpose of this study was to offer a possibility to train a model from scratch.

Another approach that takes advantage of large databases with little ground truth data is the use of semi-supervised learning methods [25,26], that can yield improvements even in a small data regime. Another way to minimize the burden of creating manual annotations is to use active learning, whereby the most uncertain predictions are selected for manual correction before re-training the model [7].

The proposed open-source framework is freely available at https://github.com/neuropoly/multiclass-segmentation. It provides a tool to partially automate the segmentation of MRIs. It is an end to end, self-sufficient solution that shows good results on homogeneous datasets, even with small amounts of data.

We hope this framework will result in the creation and sharing of trained models that will facilitate research in medical studies.

## Acknowledgements

Funded by the Canada Research Chair in Quantitative Magnetic Resonance Imaging [950-230815], the Canadian Institute of Health Research [CIHR FDN-143263], the Canada Foundation for Innovation [32454, 34824], the Fonds de Recherche du Québec - Santé [28826], the Fonds de Recherche du Québec - Nature et Technologies [2015-PR-182754], the Natural Sciences and Engineering Research Council of

**Table 2**

Results of the segmentation methods that participated in the spinal cord grey matter segmentation challenge [1] and our method (last two columns). Values for each metric are in the format mean (std). The metrics are: Dice similarity coefficient (DSC), mean surface distance (MSD), Hausdorff surface distance (HSD), skeletonized Hausdorff distance (SHD), skeletonized median distance (SMD), true positive rate (TPR), true negative rate (TNR), positive predictive value (PPV), Jaccard index (JI) and conformity coefficient (CC). MSD, HSD, SHD and SMD metrics are in millimeter. The best results are shown in bold font.

| | JCSCS | DEEPSEG | MGCAC | GSBME | SCT | VBEM | NoPoolASPP | Our models (one per site) | Our model (trained on all sites) |
|---|---|---|---|---|---|---|---|---|---|
| DSC | 0.79 (0.04) | 0.80 (0.06) | 0.75 (0.07) | 0.76 (0.06) | 0.69 (0.07) | 0.61 (0.13) | 0.85 (0.04) | **0.86** (0.03) | 0.84 (0.14) |
| MSD | 0.39 (0.44) | 0.46 (0.48) | 0.70 (0.79) | 0.62 (0.64) | 0.69 (0.76) | 1.04 (1.14) | 0.36 (0.34) | **0.32** (0.31) | 0.51 (1.03) |
| HD | 2.65 (3.40) | 4.07 (3.27) | 3.56 (1.34) | 4.92 (3.30) | 3.26 (1.35) | 5.34 (15.35) | 2.61 (2.15) | **2.12** (1.10) | 4.20 (12.45) |
| SHD | 1.00 (0.35) | 1.26 (0.65) | 1.07 (0.37) | 1.86 (0.85) | 1.12 (0.41) | 2.77 (8.10) | 0.85 (0.32) | **0.84** (0.33) | 0.96 (0.39) |
| SMD | 0.37 (0.18) | 0.45 (0.20) | 0.39 (0.17) | 0.61 (0.35) | 0.39 (0.16) | 0.54 (0.25) | **0.36** (0.17) | 0.37 (0.17) | 0.44 (0.29) |
| TPR | 77.98 (4.88) | 78.89 (10.33) | 87.51 (6.65) | 75.69 (8.08) | 70.29 (6.76) | 65.66 (14.39) | **94.97** (3.50) | 93.40 (3.61) | 88.93 (15.56) |
| TNR | **99.98** (0.03) | 99.97 (0.04) | 99.94 (0.08) | 99.97 (0.05) | 99.95 (0.06) | 99.93 (0.09) | 99.95 (0.06) | 99.96 (0.05) | 99.97 (0.03) |
| PPV | **81.06** (5.97) | 82.78 (5.19) | 65.60 (9.01) | 76.26 (7.41) | 67.87 (8.62) | 59.07 (13.69) | 77.29 (6.46) | 80.76 (6.18) | 80.32 (13.59) |
| JI | 0.66 (0.05) | 0.68 (0.08) | 0.60 (0.08) | 0.61 (0.08) | 0.53 (0.08) | 0.45 (0.13) | 0.74 (0.06) | **0.76** (0.05) | 0.74 (0.13) |
| CC | 47.17 (11.87) | 49.52 (20.29) | 29.36 (29.53) | 33.69 (24.23) | 6.46 (30.59) | −44.25 (90.61) | 64.24 (10.83) | **68.18** (10.01) | −1654.79 (11,953.99) |

Canada [RGPIN-2013-435897, RGPIN-2019-07244], the Canada First Research Excellence Fund (IVADO and TransMedTech) and the Quebec BioImaging Network [5886, 35450].

## References

[1] Prados F, Ashburner J, Blaiotta C, Brosch T, Carballido-Gamio J, Cardoso MJ, et al. Spinal cord grey matter segmentation challenge. Neuroimage 2017;152:312–29.

[2] Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. Med Image Anal 2017;42:60–88.

[3] Perone CS, Cohen-Adad J. Promises and limitations of deep learning for medical image segmentation. Journal of Medical Artificial Intelligence 2019:2 Available: http://jmai.amegroups.com/article/view/4659/html.

[4] Csurka G. A comprehensive survey on domain adaptation for visual applications. In: Csurka G, editor. Domain adaptation in computer vision applications. Cham: Springer International Publishing; 2017. p. 1–35.

[5] Gibson E, Li W, Sudre C, Fidon L, Shakir DI, Wang G, et al. NiftyNet: a deep-learning platform for medical imaging. Comput Methods Programs Biomed 2018;158:113–22.

[6] Rajchl M. An introduction to biomedical image analysis with TensorFlow and DLTK. Medium [Internet]. TensorFlow; 3 Jul 2018 [cited 3 Jan 2019]. Available:. https://medium.com/tensorflow/an-introduction-to-biomedical-image-analysis-with-tensorflow-and-dltk-2c25304e7c13.

[7] Gorriz M, Carlier A, Faure E, Giro-i-Nieto X. Cost-effective active learning for melanoma segmentation [internet]. arXiv [cs.CV] Available: http://arxiv.org/abs/1711.09168; 2017.

[8] Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, et al. Automatic differentiation in PyTorch [internet] Available: https://openreview.net/pdf?id=BJJsrmfCZ; 2017.

[9] LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, et al. Backpropagation applied to handwritten zip code recognition. Neural Comput 1989;1:541–51. MIT Press.

[10] Guyon I. A scaling law for the validation-set training-set size ratio. AT&T Bell Laboratories. Citeseer; 1997. p. 1–11.

[11] Baumgartner CF, Koch LM, Pollefeys M, Konukoglu E. An exploration of 2D and 3D deep learning techniques for cardiac MR image segmentation. In: Pop M, Sermesant M, Jodoin P-M, Lalande A, Zhuang X, Yang G, editors. Statistical atlases and computational models of the heart ACDC and MMWHS challenges. Cham: Springer International Publishing; 2018. p. 111–9.

[12] Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation [internet]. arXiv [cs.CV] Available: http://arxiv.org/abs/1505.04597; 2015.

[13] Badrinarayanan V, Handa A, Cipolla R. SegNet: a deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling [internet]. arXiv [cs.CV] Available: http://arxiv.org/abs/1505.07293; 2015.

[14] Perone CS, Calabrese E, Cohen-Adad J. Spinal cord gray matter segmentation using deep dilated convolutions. Sci Rep 2018;8:5966.

[15] Kingma DP, Ba J Adam. A method for stochastic optimization [internet]. arXiv [cs.LG] Available: http://arxiv.org/abs/1412.6980; 2014.

[16] Milletari F. Hough voting strategies for segmentation, detection and tracking [internet]. Universität München 2018:72–4 Available: https://mediatum.ub.tum.de/doc/1395260/1395260.pdf.

[17] Sudre CH, Li W, Vercauteren T, Ourselin S, Jorge Cardoso M. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations [internet]. arXiv [cs.CV] Available: http://arxiv.org/abs/1707.03237; 2017.

[18] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift [internet]. arXiv [cs.LG] Available: http://arxiv.org/abs/1502.03167; 2015.

[19] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15:1929–58.

[20] Lefeuvre J. In vivo magnetic resonance imaging of the marmoset spinal cord at 7T. 25th annual meeting of ISMRM. 2017. [Hawai].

[21] Lee NJ, Ha S-K, Sati P, Absinta M, Luciano NJ, Lefeuvre JA, et al. Spatiotemporal distribution of fibrinogen in marmoset and human inflammatory demyelination. Brain 2018;141:1637–49.

[22] Lefeuvre JA, Guy JR, Luciano NJ, Ha S-K, Leibovitch E, Santin MD, et al. The spectrum of spinal cord lesions in a primate model of multiple sclerosis. Mult Scler 2019:1352458518822408https://doi.org/10.1177/1352458518822408.

[23] Dupont SM, De Leener B, Taso M, Le Troter A, Stikov N, Callot V, et al. Fully-integrated framework for the segmentation and registration of the spinal cord white and gray matter. Neuroimage 2016. https://doi.org/10.1016/j.neuroimage.2016.09.026.

[24] Gros C, De Leener B, Badji A, Maranzano J, Eden D, Dupont SM, et al. Automatic segmentation of the spinal cord and intramedullary multiple sclerosis lesions with convolutional neural networks. Neuroimage. 2018; doi: https://doi.org/10.1016/j.neuroimage.2018.09.081.

[25] Perone CS, Cohen-Adad J. Deep semi-supervised segmentation with weight-averaged consistency targets. In: Stoyanov D, editor. Deep learning in medical image analysis and multimodal learning for clinical decision support. DLMIA 2018, ML-CDS 2018. Lecture notes in computer science. 11045. Cham: Springer; 2018.

[26] Perone Christian S, Ballester Pedro, Barros Rodrigo C, Cohen-Adad Julien. Unsupervised domain adaptation for medical imaging segmentation with self-ensembling. NeuroImage 2019;194:1–11. https://doi.org/10.1016/j.neuroimage.2019.03.026.