Deep Convolutional Encoder Networks for Multiple Sclerosis Lesion Segmentation

***

[1] ***
[2] ***
[3] ***

Abstract. We propose a novel segmentation approach based on deep convolutional encoder networks and apply it to the segmentation of multiple sclerosis (MS) lesions in magnetic resonance images (MRIs). Our model is a neural network that is both convolutional and deconvolutional, and combines feature extraction and segmentation prediction in a single model. The joint training of the feature extraction and prediction layers allows the model to automatically learn features that are optimized for accuracy for any given combination of image types and application. In contrast to existing automatic feature learning approaches, which are typically patch-based, our model learns features from entire images, which eliminates patch selection and reduces redundant calculations at the overlap of neighboring patches and thereby speeds up the training. This allows our model to be trained on larger data sets in order to learn features that cover the broad spectrum of lesion variability. We have evaluated our method on the publicly available labeled cases from the MS Lesion Segmentation Challenge 2008 data set, showing that our method performs comparably to the state-of-the-art even when a relatively small data set is used for training, which is typically not the strength of neural networks. In addition, we have evaluated our method on 500 images (split equally into training and test sets) from a data set from an MS clinical trial, showing that the segmentation performance can be greatly improved by having a representative training set.

Keywords: Multiple sclerosis lesions, segmentation, MRI, machine learning, unbalanced classification, deep learning, convolutional neural nets

1 Introduction

Multiple sclerosis (MS) is an inflammatory and demyelinating disease of the central nervous system with pathology that can be observed in vivo by magnetic resonance imaging (MRI). MS is characterized by the formation of lesions, primarily visible in the white matter on conventional MRI. Imaging biomarkers based on lesion segmentations, such as lesion load and lesion count, have established their importance ~~to assess~~ disease progression and treatment affect. However, lesions vary greatly in shape, intensity and location, which makes their automatic and accurate segmentation challenging.

Many automatic methods have been proposed for the segmentation of MS lesions over the last two decades. Most methods formalize lesion segmentation as a voxel classification problem, where each voxel of an image is assigned one of the two classes "lesion voxel" and "non-lesion voxel". The classification problem itself can then be solved ~~either in a supervised way using, e.g.,~~ artificial neural networks [1] or random forests [2], or unsupervised ~~using clustering methods with one outlier class [3] or by treating lesions as an outlier of a generative model [4].~~ A variety of features have been proposed to drive the segmentation. Early approaches have used the intensity values of different modalities at a particular voxel location as the input features [1]. However, simple intensity features can be sensitive to ~~intensity~~ variations between images. Geremia et. al [2] have shown that carefully chosen context-rich features are more robust to intensity variations, which improves segmentation accuracy. ~~Instead of hand-designing features, Youngjin~~ et. al [5] proposed ~~to learn~~ domain-specific features from image patches from ~~an~~ unlabelled data ~~set~~ using ~~unsupervised feature~~ learning. ~~For the automatic segmentation of cell membranes, Ciresan et. al~~ proposed to classify the center of image patches directly using a convolutional neural network without a dedicated feature extraction step [6]. ~~Features~~ are learned indirectly within the lower layers of the neural network during training, while the higher layers can be regarded as performing the classification. In contrast to unsupervised feature learning, this approach allows the learning of features that are specifically tuned to the segmentation task. Although supervised and unsupervised feature learning methods have shown great potential for image segmentation, the time required to train complex patch-based ~~feature extraction~~ methods can make the approach infeasible when the size and ~~the~~ number of patches ~~is~~ large. Ciresan et. al ~~have~~ reported a training time of more than a week ~~to train~~ their patch-based ~~segmentation~~ model using 4 GPUs on 2D images with a resolution of 512×512 pixels [6]. To scale patch-based classification to 3D images with a resolution of $256 \times 256 \times 50$, ~~Youngjin et. al~~ used only a small fraction (0.1 %) of the possible patches for training, which ~~might~~ limit the ability to learn features ~~that are representative of the entire image~~.

In this paper, we propose a ~~novel~~ method for segmenting MS lesions that can automatically learn features tuned for lesion segmentation ~~and scales better to large data sets of high-resolution 3D images than previous patch-based feature learning approaches, which~~ allows our model to take advantage of large data sets. Our ~~model is a~~ neural network ~~that~~ is composed of three layers: an input layer composed of the image voxels, a convolutional layer [7] that extracts features from multi-modal MRIs at each voxel location, and a deconvolutional layer [8] that uses the extracted features ~~from the first layer to~~ classify each voxel of the image in a single operation. ~~Both layers are~~ trained at the same time, which ~~facilitates the learning of features that are tuned for lesion segmentation. A key difference to the network of Ciresan et. al [6] is that our model is trained on the entire images instead of multiple patches from the same image, which eliminates~~ redundant calculations at the overlap of neighboring patches ~~and thereby~~ speeds up the training and ~~eliminates~~ the need to select representative patches. This

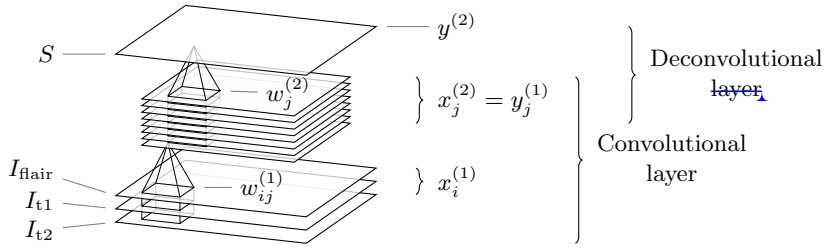

Fig. 1. Convolutional encoder network used to segment MS lesion of images from the MICCAI 2008 lesion segmentation challenge. The first two layers form a convolutional neural network, and the last two layers form a deconvolutional neural network.

allows our model to be trained on large data sets in order to learn features that cover the broad spectrum of lesion variability. The proposed network is similar in architecture to a convolutional auto-encoder [9] but instead of learning a lower dimensional representation of the input images themselves, the output of ~~your~~ network are the predicted lesion masks. Due to the structural similarity to convolutional auto-encoders, we will call our model a convolutional encoder network (CEN). Traditionally, neural networks are trained by back-propagating the sum of squared differences (SSD) of the predicted and ~~the~~ expected output. However, if one class is ~~much overrepresented~~, as is the case for ~~lesion segmentation,~~ the algorithm would learn to ignore the minority class completely. To overcome this problem, we propose ~~to use the weighted sum of sensitivity and specificity error as a new objective function, which is suitable to deal with very unbalanced classification problems, and we will derive the gradients of our proposed objective function in order train the model using stochastic gradient descent.~~

2 Methods

In this paper, the task of segmenting MS lesions is defined as finding a function s that maps multi-modal images I, e.g., $I = (I_{\text{flair}}, I_{\text{t1}}, I_{\text{t2}})$, to corresponding lesion masks S. Given a set of training images I_n, $n \in \mathbb{N}$, and corresponding segmentations S_n, we model finding an appropriate function for segmenting MS lesions as an optimization problem of the following form

need a better way to write this

$$\hat{s} = \arg\min_{s \in \mathcal{S}} \sum_n E(S_n, s(I_n)). \tag{1}$$

where $\mathcal{S}$ is the set of possible segmentation functions, and E is an error measure that calculates the dissimilarity between ground truth segmentations and predicted segmentations.

The set of possible segmentation functions is modeled by the convolutional encoder network illustrated in Figure 1. Our network consists of three layers: an input layer, a convolutional layer, and a deconvolutional layer. The input layer is composed of the image voxels $x_i^{(1)}(\boldsymbol{p})$, $i \in [1, C], C \in \mathbb{N}$, where i indexes the

modality, C is the number of modalities, and $\boldsymbol{p} \in \mathbb{R}^3$ are the coordinates of a particular voxel. The convolutional layer ~~extracts automatically learned~~ features from the input images. It is a deterministic function of the following form

$$y_j^{(1)} = \max\left(0, \sum_{i=1}^{C} \tilde{w}_{ij}^{(1)} * x_i^{(1)} + b_j^{(1)}\right) \tag{2}$$

where $y_j^{(1)}, j \in [1, F], F \in \mathbb{N}$, denotes the feature map corresponding to the jth feature, F is the number of features, w_{ij} and b_j are trainable parameters of the model, $*$ denotes valid convolution, and $\tilde{w}_{ij}$ denotes a flipped version of w_{ij}. The deconvolutional layer uses the extracted features to calculate a probabilistic lesion mask as follows

This equation is almost identical to equation (1) in [8]. I've just added the sigmoid function because our output is binary.

$$y^{(2)} = \text{sigm}\left(\sum_{j=1}^{F} w_j^{(2)} \circledast x_j^{(2)} + b^{(2)}\right) \tag{3}$$

where $x_j^{(2)} = y_j^{(1)}$, $w_j^{(2)}$ and $b^{(2)}$ are trainable parameters, $\circledast$ denotes full convolution, and $\text{sigm}(x)$ denotes the sigmoid function defined as $\text{sigm}(z) = (1 + \exp(-z))^{-1}, z \in \mathbb{R}$. To obtain a binary lesion mask from the probabilistic output of our model, we chose a threshold such that the average dice similarity coefficient is maximized on the training set.

The parameters of the model can be efficiently learned by minimizing the error E on the training set using stochastic gradient descent (SGD) [7]. Typically, neural networks are trained by minimizing the sum of squared differences (SSD)

$$E = \text{SSD} = \frac{1}{2} \sum_{\boldsymbol{p}} \left(S(\boldsymbol{p}) - y^{(2)}(\boldsymbol{p})\right)^2. \tag{4}$$

The partial derivatives of the error with respect to the model parameters can be calculated using the delta rule and are given by

$$\frac{\partial E}{\partial w_j^{(2)}} = \delta^{(2)} * \tilde{x}_j^{(2)}, \qquad \frac{\partial E}{\partial b^{(2)}} = \frac{1}{N^3} \sum_{\boldsymbol{p}} \delta^{(2)}(\boldsymbol{p}) \tag{5}$$

with

$$\delta^{(2)} = \left(y^{(2)} - S\right) y^{(2)} \left(1 - y^{(2)}\right) \tag{6}$$

where N^3 is the number of voxels of a single input channel. The derivatives of the error with respect to the first layer parameters can be calculated by applying the chain rule of partial derivatives and is given by

$$\frac{\partial E}{\partial w_{ij}^{(1)}} = x_i^{(1)} * \tilde{\delta}_j^{(1)}, \qquad \frac{\partial E}{\partial b_j^{(1)}} = \frac{1}{M^3} \sum_{\boldsymbol{q}} \delta_j^{(1)}(\boldsymbol{q}) \tag{7}$$

with

$$\delta_j^{(1)} = \left(w_j^{(2)} \circledast \delta^{(2)}\right) \mathbb{I}\left(y_j^{(1)} > 0\right) \tag{8}$$

where M^3 is the number of voxels of a feature map and $\mathbb{I}(z)$ denotes the indicator function, which is defined as 1 if the predicate z is true and 0 otherwise.

The sum of squared differences is a good measure of classification accuracy, if the two classes are fairly balanced. However, if one class contains vastly more samples than the other class, the error measure is dominated by the majority class and consequently, the neural network would learn to completely ignore the minority class. To overcome this problem, we use a combination of sensitivity and specificity, which are two measures that are suitable to measure classification performance even for vastly unbalanced classification problems. More precisely, we calculate the mean squared difference for lesion and non-lesion voxels separately and then calculate the weighted sum of the two terms to form the final error measure

$$E = r_{\text{sen}} \frac{\sum_{\boldsymbol{p}} \left(S(\boldsymbol{p}) - y^{(2)}(\boldsymbol{p})\right)^2 S(\boldsymbol{p})}{\sum_{\boldsymbol{p}} S(\boldsymbol{p})} + (1 - r_{\text{sen}}) \frac{\sum_{\boldsymbol{p}} \left(S(\boldsymbol{p}) - y^{(2)}(\boldsymbol{p})\right)^2 \left(1 - S(\boldsymbol{p})\right)}{\sum_{\boldsymbol{p}} \left(1 - S(\boldsymbol{p})\right)} \tag{9}$$

where the first term captures the squared sensitivity error and the second term captures the squared specificity error. We formulate the sensitivity and specificity error as a squared error in order to yield smooth gradients, which makes the optimization more robust. The sensitivity ratio r_{sen} can be used to assign different weights to the two terms. Due to the large number of non-lesion voxels, weighting the specificity higher than the sensitivity is preferable. We found that the sensitivity ratio mostly affects the optimal lesion threshold, but has only a minor impact on the actual segmentation quality. On all our experiments, a sensitivity ratio between 0.1 and 0.01 yields very similar results.

To train our model, we have to derive the derivatives of the modified objective function with respect to the model parameters. Equations (5), (7), and (8) are a consequence of the chain rule of derivatives and independent of the chosen similarity measure. Hence, we only need to derive the update rule for $\delta^{(2)}$. With $\alpha = 2r_{\text{sen}}(\sum_{\boldsymbol{p}} S(\boldsymbol{p}))^{-1}$ and $\beta = 2(1 - r_{\text{sen}})(\sum_{\boldsymbol{p}}(1 - S(\boldsymbol{p})))^{-1}$ we can rewrite E as

$$E = \frac{1}{2} \sum_{\boldsymbol{p}} \left(S(\boldsymbol{p}) - y^{(2)}(\boldsymbol{p})\right)^2 \alpha S(\boldsymbol{p}) + \frac{1}{2} \sum_{\boldsymbol{p}} \left(S(\boldsymbol{p}) - y^{(2)}(\boldsymbol{p})\right)^2 \beta \big(1 - S(\boldsymbol{p})\big) \tag{10}$$

$$= \frac{1}{2} \sum_{\boldsymbol{p}} \big(\alpha S(\boldsymbol{p}) + \beta(1 - S(\boldsymbol{p}))\big) \left(S(\boldsymbol{p}) - y^{(2)}(\boldsymbol{p})\right)^2 \tag{11}$$

Our objective function is similar to the SSD, with the difference that an additional term is multiplied to the squared differences. The additional factor does not depend on $y^{(2)}$ and is therefore constant with respect to the model parameters. Consequently, $\delta^{(2)}$ can be derived analogously to the SSD case, where the

Table 1. Comparison of state of the art methods with our method.

Method	TPR	PPV	DSC
Souplet et al. [3]	20.65	30.00	—
Weiss et al. [4]	33.00	36.85	29.05
Geremia et al. [2]	39.85	40.35	—
Our method	39.71	41.38	35.52

new factor is carried over:

$$\delta^{(2)} = \big(\alpha S + \beta(1-S)\big)\big(y^{(2)} - S\big)y^{(2)}\big(1-y^{(2)}\big) \tag{12}$$

3 Experiments and Results

describe what I mean by lesion prior

To allow for a direct comparison to other state of the art lesion segmentation methods, we have evaluated our method on the 20 labeled cases from the MICCAI lesion segmentation challenge [10]. Data set contains FLAIR, T1, and T2. In addition, we have added a lesion prior as a forth channel. Our preprocessing pipeline includes downsampling to $1\,\text{mm} \times 1\,\text{mm} \times 1\,\text{mm}$ resolution, global contrast normalization, brain extraction. We have then cropped the images to a $159 \times 203 \times 153$ region of interest. To evaluate the performance, we have divided the data set into 5 splits with each split containing 16 images for training and 4 images for testing such that each image will be used exactly once for testing. For all experiments, we have used a filter size of $9 \times 9 \times 9$, 32 filters and trained the model for 4000 epochs. Training took approximately 36 hours on a single Geforce GTX 760 graphics card. Once the model is trained, segmentation of a single image can be performed in less than one second.

Statistics about lesion load (range, mean) for both data sets.

Figure 2 shows a comparison of segmentations predicted by our method with ground truth segmentation for three different cases. The first two rows illustrate cases were our method performs very well. Our method achieves a DSC of about 0.61 in both cases, despite the images varying greatly in contrast, which shows that our method is able to learn features that are robust to different contrast. The last row illustrates a case were our method produces more false positives than usually causing a DSC of only 9. This is caused by the inability of our method to distinguish between diffusely abnormal white matter and MS lesions.

I would like to say where this issues appear but I'm also not to sure how to describe the location.

A comparison of our method with other state of the art methods is summarized in Table 1. Our method outperforms the winning method (Souplet et al. [3]) of the MICCAI 2008 challenge and a recently proposed method based on sparse coding (Weiss et al. [4]) and performs comparable with the current state of the art using a combination of carefully designed features for MS lesion segmentation and random forests (Geremia et al. [2]), despite not requiring any domain knowledge.

To evaluate the impact of the training set size on the segmentation performance, we have evaluated our model on different subsets of a data set from an MS clinical trail. With data set contained T2w and PDw images from 500

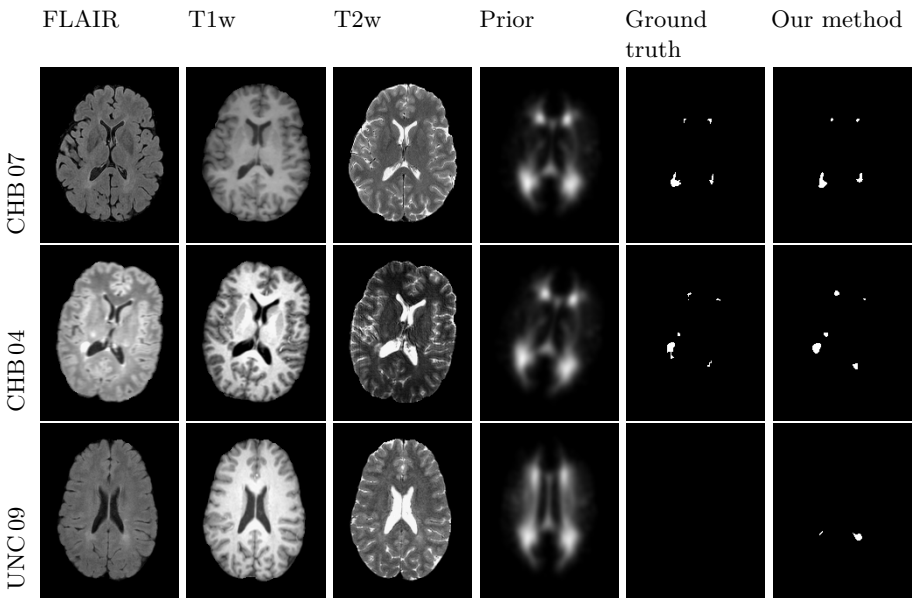

Fig. 2. Example segmentation using our method. Comment why the method performed poorly. Maybe it just was a very difficult case. CHB07, FLAIR, T1, T2, prior, ground truth, predicted segmentation. CHB04, UNC09. Robust to different contrast, but misclassifies diffusely abnormal white matter as MS lesions.

subjects at a single time point. We have divided the data set into a training and test set containing 250 images each. We have then trained our model on a varying number of image from the training set and evaluated the segmentation performance on the selected training samples and all images of the test set. The result of this test is illustrated in Table 3. For very small number of training samples, our model achieves a DSC of 71 in the training set and 49 in the test set. The large difference between the results on the training and test set indicates that the training sample size is not sufficient to prevent overfitting. As we increase the number of training samples, the difference between the results on the training and test set decreases. At 150 samples, the performance on the training set is almost as good as on the test set, which indicates that our model is not overfitting. We also do not see an improvement thereafter. For this study, if have trained a relatively simple model with only 2 layers and 32 filters per layer. We expect that models with more layers and filters are more prone to overfitting for smaller training set sizes, but might keep improving beyond 150 samples.

Table 2. This table is here just for reference. I'm not planning to keep it, but it shows the full picture of the MICCAI tests.

Patient	Souplet		Geremia		Weiss			Our method		
	TPR	PPV	TPR	PPV	TPR	PPV	DSC	TPR	PPV	DSC
CHB01	22	41	49	64	60	58	59	50	69	58
CHB02	18	29	44	63	27	45	34	42	52	46
CHB03	17	21	22	57	24	56	34	38	70	49
CHB04	12	55	31	78	27	66	38	60	63	61
CHB05	22	42	40	52	29	33	31	42	43	42
CHB06	13	46	32	52	10	36	16	24	63	35
CHB07	13	39	40	54	14	48	22	57	65	61
CHB08	13	55	46	65	21	73	32	47	75	58
CHB09	3	18	23	28	5	22	8	22	49	30
CHB10	5	18	23	39	15	12	13	11	64	19
UNC01	1	1	2	1	33	29	31	3	6	4
UNC02	37	39	48	36	54	51	53	54	38	44
UNC03	12	16	24	35	64	27	38	62	28	39
UNC04	38	54	54	38	40	51	45	59	35	44
UNC05	38	8	56	19	25	10	16	10	5	6
UNC06	57	9	15	8	13	55	20	24	49	32
UNC07	27	18	76	16	44	23	30	33	19	24
UNC08	27	20	52	32	43	13	20	51	13	20
UNC09	16	43	67	36	69	6	11	51	5	9
UNC10	22	28	53	34	43	23	30	56	19	28
Average	20.7	30.0	39.9	40.4	33.0	36.9	29.1	39.7	41.4	35.5

4 Conclusions

- Future work: use more layers to achieve a hierarchical segmentation method, but this paper, focus on the simplest possible network to evaluate the potential of such an approach.
- Used a simple model to reduce the risk of overfitting. In the future, we are planning to add more layers and use more filters and apply the model to larger data sets.
- We have demonstrated the potential of our approach for MS lesion segmentation, although the method is not inherently limited to this kind of segmentation. We are planning to apply this framework to other segmentation problems and we anticipate that other groups will adopt this approach to a variety of segmentation problems.

Acknowledgements ****
****

Table 3. Comparison of segmentation performance on the training and test set for varying number of training samples. The difference between training and test performance is reduces for increasing number of training samples. A training set size of 150 is sufficient to prevent overfitting.

Number of	Training set			Test set		
training samples	TPR	PPV	DSC	TPR	PPV	DSC
5	77.97	66.23	70.93	47.16	57.04	48.55
10	73.40	68.16	69.71	55.04	59.85	53.93
15	71.77	68.23	68.86	56.28	60.88	55.47
25	69.39	70.63	68.94	57.43	61.06	56.29
250	64.55	58.25	58.50	65.47	56.81	57.54

References

1. Zijdenbos, A.P., Dawant, B.M., Margolin, R.A., Palmer, A.C.: Morphometric analysis of white matter lesions in mr images: method and validation. Medical Imaging, IEEE Transactions on **13**(4) (1994) 716–724
2. Geremia, E., Menze, B.H., Clatz, O., Konukoglu, E., Criminisi, A., Ayache, N.: Spatial decision forests for ms lesion segmentation in multi-channel mr images. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010. Springer (2010) 111–118
3. Souplet, J.C., Lebrun, C., Ayache, N., Malandain, G.: An automatic segmentation of t2-flair multiple sclerosis lesions. In: The MIDAS Journal-MS Lesion Segmentation (MICCAI 2008 Workshop). (2008)
4. Weiss, N., Rueckert, D., Rao, A.: Multiple sclerosis lesion segmentation using dictionary learning and sparse coding. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013. Springer (2013) 735–742
5. Yoo, Y., Brosch, T., Traboulsee, A., Li, D.K., Tam, R.: Deep learning of image features from unlabeled data for multiple sclerosis lesion segmentation. In: Machine Learning in Medical Imaging. Springer (2014) 117–124
6. Ciresan, D., Giusti, A., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. Advances in Neural Information Processing Systems (2012) 1–9
7. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324
8. Zeiler, M.D., Taylor, G.W., Fergus, R.: Adaptive deconvolutional networks for mid and high level feature learning. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 2018–2025
9. Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: Artificial Neural Networks and Machine Learning–ICANN 2011. Springer (2011) 52–59
10. Styner, M., Lee, J., Chin, B., Chin, M., Commowick, O., Tran, H., Markovic-Plese, S., Jewells, V., Warfield, S.: 3d segmentation in the clinic: A grand challenge ii: Ms lesion segmentation. MIDAS Journal **2008** (2008) 1–6