Deep Shortcut Convolutional Encoder Networks for Multiple Sclerosis Lesion Segmentation

Tom Brosch and Roger Tam

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. Our model is a neural network that has both convolutional and deconvolutional layers, 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. 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 redundant calculations at the overlap of neighboring patches and thereby speeds up the training. Our network also uses a novel objective function that works well for segmenting underrepresented classes, such as MS lesions. 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. In addition, we have evaluated our method on the images of 500 subjects from an MS clinical trial and varied the number of training samples from 5 to 250 to show that the segmentation performance can be greatly improved by having a representative data set.**

Index Terms—**Segmentation, multiple sclerosis lesions, MRI, machine learning, unbalanced classification, deep learning, convolutional neural nets**

I. 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 the delineation of lesions, such as lesion load and lesion count, have established their importance for assessing disease progression and treatment effect. However, lesions vary greatly in size, 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 [7], which can be classified into unsupervised and supervised methods. Unsupervised methods do not require a labeled data set for training. Instead, lesions are identified as an outlier class using, e.g., clustering methods [20] or dictionary learning and sparse coding to model healthy tissue [21]. Current supervised approaches typically start with a large set of features, either predefined by the user [8] or gathered in a feature extraction step, which is followed by a separate training step with labeled data to determine which set of features are the most important for segmentation in the particular domain. For example, Yoo et al. [22] proposed performing unsupervised learning of domain-specific features from image patches from unlabelled data using deep learning. A major breakthrough for the fully automatic segmentation of medical images using deep learning comes from the domain of cell membrane segmentation, in which Ciresan et al. [3] proposed to classify the centers of image patches directly using a convolutional neural network (CNN) [15] without a dedicated feature extraction step. Instead, 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 deep network-based feature learning methods have shown great potential for image segmentation, the time required to train complex patch-based methods can make the approach infeasible when the size and number of patches are large.

More recently, different CNN architectures [2], [13], [17] have been proposed that are able to learn to segment entire images instead of patches, which removes the need to select representative patches, eliminates redundant calculations where patches overlap, and therefore scales up more efficiently with image resolution. Kang et al. used a fully convolutional neural network to segment crowds in surveillance videos [13]. However, fully convolutional neural networks predict segmented images of lower resolution than the input images due to the successive use of convolutional and pooling layers, which both reduce the dimensionality. To predict segmentations of the same resolution as the input images, we proposed to use a 3-layer convolutional encoder network (CEN) [2] for multiple sclerosis lesion segmentation. The combination of convolutional [15] and deconvolutional [24] layers allows our network to produce segmentations that are of the same resolution as the input images and are therefore potentially more accurate compared to segmentations predicted by a fully convolutional neural network. Ronneberger et al. [17] found that increasing the depth of the CNN, and thereby increasing the size of the receptive field of a segmented voxel, increases the abstraction from the input data, which makes the segmentation method more robust to anatomical and intensity variations.

T. Brosch is with the Multiple Sclerosis/Magnetic Resonance Imaging Research Group, Division of Neurology, The University of British Columbia, Vancouver, BC V6T 2B5, Canada, and also with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: tombr@msmri.medicine.ubc.ca)

R. Tam is with the Multiple Sclerosis/Magnetic Resonance Imaging Research Group, Division of Neurology, The University of British Columbia, Vancouver, BC V6T 2B5, Canada, and also with the Department of Radiology, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: roger@msmri.medicine.ubc.ca)

They proposed an 11-layer u-shaped network architecture called u-net, which leverages high-level and low-level features in order to predict segmentations that are both robust and accurate. However, the successive application of convolutional, pooling, and unpooling layers reduces the size of the predicted segmentation by the size of the receptive field. This requires special handling of the border regions and limits the maximum size of the receptive field.

We propose a new convolutional network architecture that combines the advantages of a 3-layer CEN and an 11-layer u-net. Our network is divided into two pathways, a convolutional pathway, which consists of alternating convolutional and pooling layers and learns increasingly more abstract and robust features, and a deconvolutional pathway, which consists of alternating deconvolutional and unpooling layers and predicts the final segmentation. In order for the segmentation to be driven by both low-level and high-level features, we introduce shortcut connections between layers of the two pathways, similar to the u-net architecture. In contrast to the u-net architecture, our network predicts segmentations that have the same resolution as the input images and therefore does not require special handling of the border of the image. We have evaluated the segmentation performance of our method on a large data set from an MS clinical trial and compared our results with Lesion-TOADS [19], a widely used freely available method for fully automatic lesion segmentation. Our results show that the CEN approach is able to segment MS lesions more accurately than Lesions-TOADS, and that our new CEN architecture further improves segmentation accuracy.

II. 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., $I = (I_{\text{FLAIR}}, I_{\text{T1}})$, 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

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

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.

A. Model Architecture

The set of possible segmentation functions is modeled by the convolutional encoder network (CEN) illustrated in Fig. 1. A CEN is a type of convolutional neural network (CNN) [15] that is divided into two pathways, the convolutional pathway and the deconvolutional [24] pathway. The convolutional pathway consists of alternating convolutional and pooling layers. The input layer of the convolutional pathway is composed of the image voxels $x_i^{(0)}(\vec{p})$, $i \in [1, C]$, where i indexes the modality or input channel, C is the number of modalities or channels, and $\vec{p} \in \mathbb{N}^3$ are the coordinates of a particular voxel. The convolutional layers automatically learn a feature hierarchy from the input images. A convolutional layer is a deterministic function of the following form

$$x_j^{(l)} = \max\left(0, \sum_{i=1} \tilde{w}_{\text{c},ij}^{(l)} * x_i^{(l-1)} + b_j^{(l)}\right), \tag{2}$$

where l is the index of a convolutional layer, $x_j^{(l)}$, $j \in [1, F]$, denotes the feature map corresponding to the trainable convolution filter $w_{\text{c},ij}^{(l)}$, F is the number of filters of the current layer, $b_j^{(l)}$ are trainable bias terms, $*$ denotes valid convolution, and $\tilde{w}$ denotes a flipped version of w. A convolutional layer is followed by an average pooling layer [18] that halves the number of units in each dimension by calculating the average of each block of $2 \times 2 \times 2$ units per channel.

The deconvolutional pathway consists of alternating unpooling layers and deconvolutional layers with optional shortcut connections to the corresponding convolutional layers. The first deconvolutional layer uses the extracted features of the convolutional pathway to calculate abstract segmentation features

$$y_i^{(L-1)} = \max\left(0, \sum_{j=1} w_{\text{d},ij}^{(L)} \circledast y_j^{(L)} + c_i^{(L-1)}\right), \tag{3}$$

where $y^{(L)} = x^{(L)}$, L denotes the number of layers of the convolutional pathway, $w_{\text{d},ij}^{(L)}$ and $c_i^{(L-1)}$ are trainable parameters of the deconvolutional layer, and $\circledast$ denotes full convolution. Subsequent deconvolutional layers use the activations of the previous layer and corresponding convolutional layer to calculate less abstract segmentation features

$$y_i^{(l)} = \max\left(0, \sum_{j=1} w_{\text{d},ij}^{(l+1)} \circledast y_j^{(l+1)} \right.$$
$$\left. + \sum_{j=1} w_{\text{s},ij}^{(l+1)} \circledast x_j^{(l+1)} + c_i^{(l)}\right), \tag{4}$$

where l is the index of a deconvolutional layer with shortcuts, and $w_{\text{s},ij}^{(l+1)}$ are the filter kernels connecting the activations of the convolutional pathway with the activations of the deconvolutional pathway. The last deconvolutional layer integrates the low-level features extracted by the first convolutional layer with the high-level features from the previous layer to calculate a probabilistic lesion mask

$$y_1^{(0)} = \operatorname{sigm}\left(\sum_{j=1}\left(w_{\text{d},1j}^{(1)} \circledast y_j^{(1)} + w_{\text{s},1j}^{(1)} \circledast x_j^{(1)}\right) + c_1^{(0)}\right), \tag{5}$$

where $\operatorname{sigm}(z)$ denotes the sigmoid function defined as $\operatorname{sigm}(z) = (1 + \exp(-z))^{-1}, z \in \mathbb{R}$. To obtain a binary lesion mask from the probabilistic output of our model, we chose a fixed threshold such that the mean Dice similarity coefficient [5] is maximized on the training set and use the same threshold for the evaluation on the test set.

B. Gradient Calculation

The parameters of the model can be efficiently learned by minimizing the error E on the training set [15], which

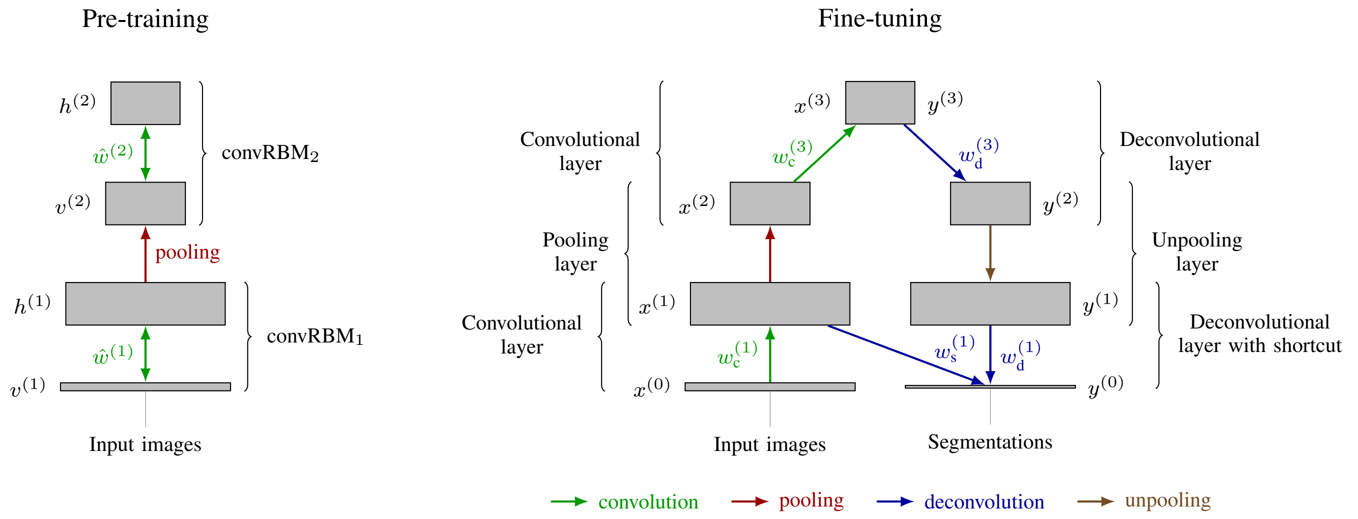


Fig. 1. Pre-training and fine-tuning of a 7-layer convolutional encoder network with shortcuts. Pre-training is performed on the input images using a stack of convolutional RBMs. The pre-trained weights and bias terms are used to initialize a convolutional encoder network, which is fine-tuned on pairs of input images, $x^{(0)}$, and segmentations, $y^{(0)}$.

requires the calculation of the gradient of E with respect to the model parameters. Typically, neural networks are trained by minimizing the sum of squared differences (SSD)

$$E = \frac{1}{2}\sum_{\vec{p}}\left(S(\vec{p}) - y^{(2)}(\vec{p})\right)^2. \tag{6}$$

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

$$\frac{\partial E}{\partial w^{(l)}_{\text{d},ij}} = \delta^{(l-1)}_{\text{d},i} * \tilde{y}^{(l)}_j, \qquad \frac{\partial E}{\partial c^{(l)}_i} = \sum_{\vec{p}} \delta^{(l)}_{\text{d},i}(\vec{p}), \tag{7}$$

$$\frac{\partial E}{\partial w^{(l)}_{\text{s},ij}} = \delta^{(l-1)}_{\text{d},i} * \tilde{x}^{(l)}_j, \tag{8}$$

$$\frac{\partial E}{\partial w^{(l)}_{\text{c},ij}} = x^{(l-1)}_i * \tilde{\delta}^{(l)}_{\text{c},j}, \text{ and } \quad \frac{\partial E}{\partial b^{(l)}_i} = \sum_{\vec{p}} \delta^{(l)}_{\text{c},i}(\vec{p}). \tag{9}$$

For the first layer, $\delta^{(0)}_{\text{d},1}$ can be calculated by

$$\delta^{(0)}_{\text{d},1} = \left(y^{(0)}_1 - S\right) y^{(0)}_1 \left(1 - y^{(0)}_1\right). \tag{10}$$

The derivatives of the error with respect to the parameters of the other layers can be calculated by applying the chain rule of partial derivatives, which yields to

$$\delta^{(l)}_{\text{d},j} = \left(\tilde{w}^{(l)}_{\text{d},ij} * \delta^{(l-1)}_{\text{d},i}\right)\mathbb{I}\left(y^{(l)}_j > 0\right), \tag{11}$$

$$\delta^{(l)}_{\text{c},i} = \left(w^{(l+1)}_{\text{c},ij} \circledast \delta^{(l+1)}_{\text{c},j}\right)\mathbb{I}\left(x^{(l)}_i > 0\right), \tag{12}$$

where l is the index of a deconvolutional layer, and convolutional layer, respectively, $\delta^{(L)}_{\text{c},i} = \delta^{(L)}_{\text{d},j}$, and $\mathbb{I}(z)$ denotes the indicator function defined as 1 if the predicate z is true and 0 otherwise. If a layer participates in a shortcut connection, $\delta^{(l)}_{\text{c},j}$ needs to be adjusted by propagating the error back through the shortcut connection. In this case, $\delta^{(l)}_{\text{c},j}$ is calculated by

$$\delta^{(l)}_{\text{c},j} = \left(\delta^{(l)\prime}_{\text{c},j} + \tilde{w}^{(l)}_{\text{s},ij} * \delta^{(l-1)}_{\text{d},i}\right)\mathbb{I}\left(x^{(l)}_j > 0\right), \tag{13}$$

where $\delta^{(l)\prime}_{\text{c},j}$ denotes the activation of unit $\delta^{(l)}_{\text{c},j}$ before taking the shortcut connection into account.

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, as is the case for lesion segmentation, 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 can be used together to measure classification performance even for vastly unbalanced problems. More precisely, the final error measure is a weighted sum of the mean squared difference of the lesion voxels (sensitivity) and non-lesion voxels (specificity), reformulated to be error terms:

$$\begin{aligned} E = r&\frac{\sum_{\vec{p}}\left(S(\vec{p}) - y^{(2)}(\vec{p})\right)^2 S(\vec{p})}{\sum_{\vec{p}} S(\vec{p})} \\ &+ (1-r)\frac{\sum_{\vec{p}}\left(S(\vec{p}) - y^{(2)}(\vec{p})\right)^2 \left(1 - S(\vec{p})\right)}{\sum_{\vec{p}}\left(1 - S(\vec{p})\right)}. \end{aligned} \tag{14}$$

We formulate the sensitivity and specificity errors as squared errors in order to yield smooth gradients, which makes the optimization more robust. The sensitivity ratio r can be used to assign different weights to the two terms. Due to the large number of non-lesion voxels, weighting the specificity error higher is important, but the algorithm is stable with respect to changes in r, which largely affects the threshold used to binarize the probabilistic output. In all our experiments, a sensitivity ratio between 0.10 and 0.01 yields very similar results.

To train our model, we must compute the derivatives of the modified objective function with respect to the model parameters. Equations 7–9 and 11–13 are a consequence of the chain rule and independent of the chosen similarity measure. Hence, we only need to derive the update rule for $\delta^{(0)}_{\text{d},1}$. With

$\alpha = 2r(\sum_{\vec{p}} S(\vec{p}))^{-1}$ and $\beta = 2(1-r)(\sum_{\vec{p}}(1-S(\vec{p})))^{-1}$, we can rewrite E as

$$E = \frac{1}{2}\sum_{\vec{p}} \left(\alpha S(\vec{p}) + \beta(1-S(\vec{p}))\right)\left(S(\vec{p}) - y_1^{(0)}(\vec{p})\right)^2. \quad (15)$$

Our objective function is similar to the SSD, with an additional multiplicative term applied to the squared differences. The additional factor is constant with respect to the model parameters. Consequently, $\delta_{\mathrm{d},1}^{(0)}$ can be derived analogously to the SSD case, and the new factor is simply carried over:

$$\delta_{\mathrm{d},1}^{(0)} = \left(\alpha S + \beta(1-S)\right)\left(y_1^{(0)} - S\right)y_1^{(0)}\left(1 - y_1^{(0)}\right). \quad (16)$$

C. Training

At the beginning of the training procedure, the model parameters need to be initialized, where the choice of the initial parameters can have a big impact on the learned model. In our experiments, we found that initializing the model using pre-training [10] was required in order to be able to fine-tune the model without getting stuck in an early local minimum. Pre-training can be performed layer by layer [9] using a stack of convolutional restricted Boltzmann machines (convRBMs) [16] (see Fig. 1), thereby avoiding the potential problem of vanishing or exploding gradients [11]. The first convRBM is trained on the input images without ground truth segmentations. Subsequent convRBMs are trained on the hidden activations of the previous convRBM. After all convRBMs have been trained, the model parameters of, e.g., the first convolutional and last deconvolutional layer of the CEN can be initialized as follows

$$w_{\mathrm{c}}^{(1)} = \hat{w}^{(1)}, \quad w_{\mathrm{d}}^{(1)} = 0.5\hat{w}^{(1)}, \quad w_{\mathrm{s}}^{(1)} = 0.5\hat{w}^{(1)} \quad (17)$$

$$b^{(1)} = \hat{b}^{(1)}, \quad c^{(0)} = \hat{c}^{(1)}, \quad (18)$$

where $\hat{w}^{(1)}$ are the filter weights, $\hat{b}^{(1)}$ are the hidden bias terms, and $\hat{c}^{(1)}$ are the visible bias terms of the first convRBM, respectively.

A major challenge for gradient-based optimization methods is the choice of an appropriate learning rate. Classic stochastic gradient descent [15] uses a fixed or decaying learning rate, which is the same for all parameters of the model. However, the partial derivatives of parameters of different layers can vary substantially in magnitude, which might require the use of different learning rates for the parameters of different layers. In recent years, there has been an increased interest in developing methods for automatically choosing the learning rate. Most methods (e.g., AdaGrad [6], AdaDelta [23], RMSprop [4], and Adam [14]) collect different statistics of the partial derivatives over multiple iterations and use this information to set an adaptive learning rate for each parameter. This is especially important for the training of deep networks, were the optimal learning rates differ greatly for each layer. In our experiments, networks obtained by training with AdaDelta, RMSprop, and Adam performed comparably well. Because we found that AdaDelta was the most robust method to the choice of hyperparameters, we have used AdaDelta in all other experiments.

III. Experiments and Results

The method was evaluation on a large data set from a multi-center MS clinical trial that was acquired from 67 different scanning sites. The data set consists of 377 pairs of FLAIR and T1-weighted MRIs from 195 subjects with a resolution of $256 \times 256 \times 60$ voxels and a voxel size of $0.936\,\mathrm{mm} \times 0.936\,\mathrm{mm} \times 3.000\,\mathrm{mm}$. All images were skull-stripped using the brain extraction tool (BET) [12], followed by an intensity normalization to the interval $[0, 1]$, and a 6 degrees of freedom intra-subject registration. To speed-up the training, all images were cropped to a $164 \times 206 \times 52$ voxels region of interest with the brain roughly centered. We used 300 image pairs for pre-training and fine-tuning, and the remaining 77 image pairs for the evaluation. Pre-training and fine-tuning was performed using a highly optimized GPU-accelerated implementation of 3D convRBMs and CENs that was develop in-house [1]. Each model was trained using 500 epochs. Pre-training and fine-tuning of a 7-layer CEN with shortcut connections took approximately 27 hours and 37 hours, respectively, on a single GeForce GTX 780 graphics card. However, once the network is trained, new image pairs can be segmented in less than one second. We compared our results to the lesion masks produced by Lesion-TOADS [19], a widely used freely available tool for the fully automatic segmentation of MS lesions.

A. Measures of Segmentation Accuracy

We have used three different metrics to measure segmentation accuracy. The primary measure of accuracy that we employ is the Dice similarity coefficient (DSC) [5], which computes a normalized overlap value between the produced and ground truth segmentation. Is is defined as

$$\mathrm{DSC} = \frac{2 \times \mathrm{TP}}{2 \times \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \quad (19)$$

where TP, FP, and FN denote the number of true positives, false positives, and false negatives, respectively. A value of $100\,\%$ indicates a perfect overlap of the produced segmentation and the ground truth. The DSC incorporates measures of over- and undersegmentation into a single metric, which makes it a suitable measure to compare overall segmentation accuracy. In addition, we have used the true positive rate (TPR) and the positive predictive value (PPV) to provide further information on specific aspects of segmentation performance. The TPR is used to measure the fraction of the regions classified as lesion by the ground truth that were also identified as lesions by an automatic method. It is defined as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad (20)$$

where a value of $100\,\%$ indicates that all true lesion voxels were automatically identified. The PPV is used to determine the extent of the regions falsely classified as lesion by an automatic method. It is defined as the amount of true lesion voxels out of all identified lesion voxels

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \quad (21)$$

TABLE I
PARAMETERS OF THE 3-LAYER CEN USED TO EVALUATING DIFFERENT TRAINING METHODS.

Layer type	Kernel Size	#Filters	Image Size
Input	—	—	$164 \times 206 \times 52 \times 2$
Convolutional	$9 \times 9 \times 5 \times 2$	32	$156 \times 198 \times 48 \times 32$
Deconvolutional	$9 \times 9 \times 5 \times 32$	1	$164 \times 206 \times 52 \times 1$

TABLE II
PARAMETERS OF THE 7-LAYER CEN USED TO EVALUATING DIFFERENT TRAINING METHODS.

Layer type	Kernel Size	#Filters	Image Size
Input	—	—	$164 \times 206 \times 52 \times 2$
Convolutional	$9 \times 9 \times 5 \times 2$	32	$156 \times 198 \times 48 \times 32$
Average Pooling	$2 \times 2 \times 2$	—	$78 \times 99 \times 24 \times 32$
Convolutional	$9 \times 10 \times 5 \times 32$	32	$70 \times 90 \times 20 \times 32$
Deconvolutional	$9 \times 10 \times 5 \times 32$	32	$78 \times 99 \times 24 \times 32$
Unpooling	$2 \times 2 \times 2$	—	$156 \times 198 \times 48 \times 32$
Deconvolutional	$9 \times 9 \times 5 \times 32$	1	$164 \times 206 \times 52 \times 1$

where a value of $100\,\%$ indicates that all voxels that were automatically classified as lesion voxels are indeed lesion voxels as defined by the ground truth.

B. Comparison of Network Architectures

In the first experiment, we have compared the segmentation performance of three different network architectures with each other and with Lesion-TOADS. Therefore, we trained a 3-layer CEN and two 7-layer CENs with and one without shortcut connections on the 300 training pairs. The parameters of the used networks are given in Table I and Table II. A comparison of the segmentation accuracy of the trained networks and Lesion-TOADS is summarized in Table III. All CEN architectures performed significantly better than Lesion-TOADS in terms of the overall segmentation accuracy, where the improvements of the mean DSC scores range from 9 pts for the 3-layer CEN to 14 pts for the 7-layer CEN with shortcut connections. The improved segmentation performance is mostly due to a reduction of the false positives. Lesion-TOADS achieved a mean PPV of only $39.86\,\%$, where the best performing CEN architecture achieved a mean PPV of $66.58\,\%$—an improvement of 27 pts. The mean TPRs were roughly the same (TPR $\approx 50\,\%$) for all methods except for the 7-layer CEN with shortcuts, which performed slightly better than the other methods with a mean TPR of $52.75\,\%$.

Furthermore, the first experiment showed that increasing the depth of the CEN and adding shortcut connections improves the segmentation accuracy. Increasing the depth of the CEN from three layers to seven layers improved the mean DSC by 2 pts. The improvement was confirmed to be statistically significant using a one-sided paired t-test (p-value $= 0.002$). Adding shortcut connections to the network further improved the segmentation accuracy as measured by the DSC by 3 pts. A second one-sided paired t-test was performed to confirm the statistical significance of the improvement with a p-value of 1.566×10^{-11}.

TABLE III
COMPARISON OF THE SEGMENTATION ACCURACY OF DIFFERENT CEN MODELS WITH LESION-TOADS

Method	TPR [%]	PPV [%]	DSC [%]
3-layer CEN	49.62 ± 20.32	59.87 ± 20.95	49.10 ± 15.78
7-layer CEN	49.94 ± 19.96	63.5 ± 20.0	51.04 ± 14.71
7-layer SCEN	52.75 ± 20.31	66.58 ± 20.71	54.02 ± 15.24
Lesion-TOADS	49.83 ± 14.79	39.86 ± 20.95	40.04 ± 14.86

Note: The table shows the mean and standard deviation of the true positive rate (TPR), positive predictive value (PPV), and Dice similarity coefficient (DSC).

TABLE IV
LESION LOAD CLASSES AS USED FOR THE DETAILED ANALYSIS.

Group	Lesion load in mm^3	Number of samples
Very low	$[0, 3250]$	17
Low	$(3250, 6500]$	16
Medium	$(6500, 10000]$	18
High	$(10000, 25000]$	18
Very high	> 25000	8

C. Comparison for Different Lesion Loads

To further examine the results, we have stratified the test set into five groups based on their reference lesion loads as summarized in Table IV. A comparison of segmentation accuracy of a 3-layer CEN and a 7-layer CEN with shortcut connections for different lesion loads is illustrated in Fig. 2. Adding four layers and shortcut connections improves the segmentation accuracy for all lesion load groups, where the improvements are largest for very high lesion loads. In our data set, lesion load is strongly correlated with lesion size, which means that the group with the highest lesion load also contains scans with the largest lesions. The receptive field of the 3-layer CEN has a size of only $17 \times 17 \times 9$ voxels, which reduces its ability to identify very large lesions. In contrast, the 7-layer CEN has a receptive field size of $49 \times 53 \times 26$ voxels, which allows it to learn features that can capture much larger lesions than the 3-layer CEN. Consequently, the 7-layer CEN is able to learn a feature set that captures a wider range of lesion sizes, which in turn improves the segmentation accuracy especially for very high lesion loads, where larger lesions are also more prevalent.

Fig. 3 shows a comparison of the 7-layer CEN with shortcut connections and Lesion-TOADS. The CEN approach performs more consistently throughout all lesion load groups than Lesion-TOADS, where the improvements are largest for very low to medium lesion loads. For very high lesion loads, Lesion-TOADS achieves a higher mean DSC than the CEN approach. However, a two-sided paired t-test yielded that the differences are not statistically significant (p-value $= 0.2566$). Table V shows a more detailed comparison. While the PPV increases consistently with higher lesion loads for both methods, the TPR is highest for low to medium lesion loads and decreases again for high to very high lesion loads. This shows the difficulty for both methods to correctly identify very large lesions that can extend far into the white matter.

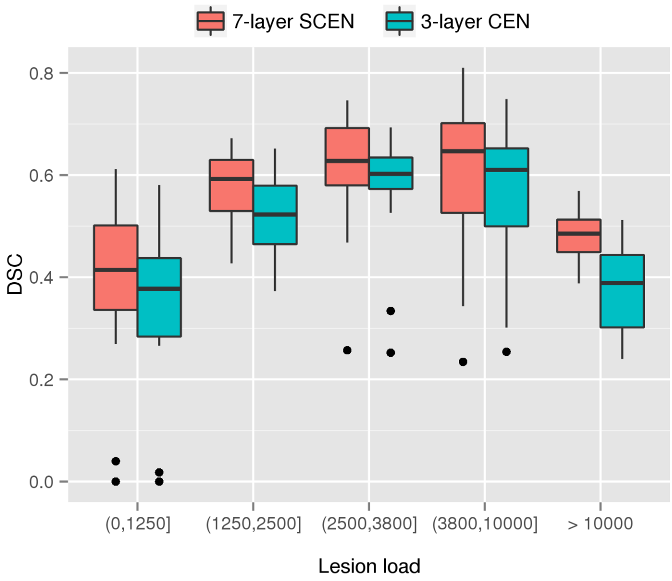


Fig. 2. Comparison of the segmentation accuracy measured by the DSC of a 3-layer CEN and a 7-layer CEN for different lesion load groups. Adding four layers and shortcut connections improves the performance across all lesion loads, where the improvements are especially large for scan with large lesion loads, which are also correlated with an increase in lesion size.

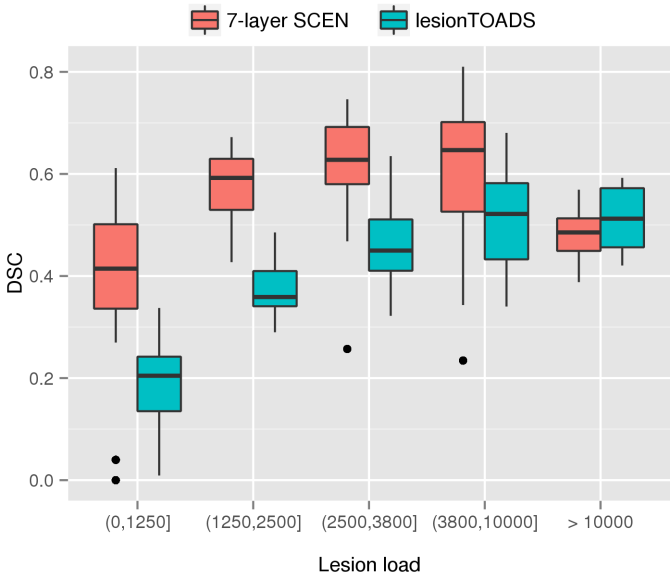


Fig. 3. Comparison of the segmentation accuracy measured by the DSC of Lesion-TOADS and a 7-layer CEN with shortcut connections for different lesion loads. The CEN approach is much more sensitive in detecting small lesions, while still being able to detect large lesions.

D. Qualitative Results

A qualitative comparison of segmentation performance for four characteristic cases is shown in Fig. 4. Our method produces segmentations that are spatially consistent (see Fig. 4a), while still being able to detect small isolated lesions (green circle). Furthermore, our method is able to learn a wide spectrum of lesion shapes and appearances from training data. This allows our method to correctly identify multiple different types of MS lesions (e.g., T1 black holes), which can be challenging to detect for Lesion-TOADS (see Fig. 4b). Our method uses a combination of automatically learned intensity and appearance features, which makes it inherently robust to noise as shown in Fig. 4c. Figure 4d shows one of the most challenging cases for our method. Very large lesions might extend beyond the size of the receptive field of the CEN, which reduces its ability to extract characteristic lesion features. Consequently, our method might underestimate the size of very large lesions for some cases.

TABLE V
COMPARISON OF SEGMENTATION ACCURACY FOR DIFFERENT LESION LOAD CATEGORIES.

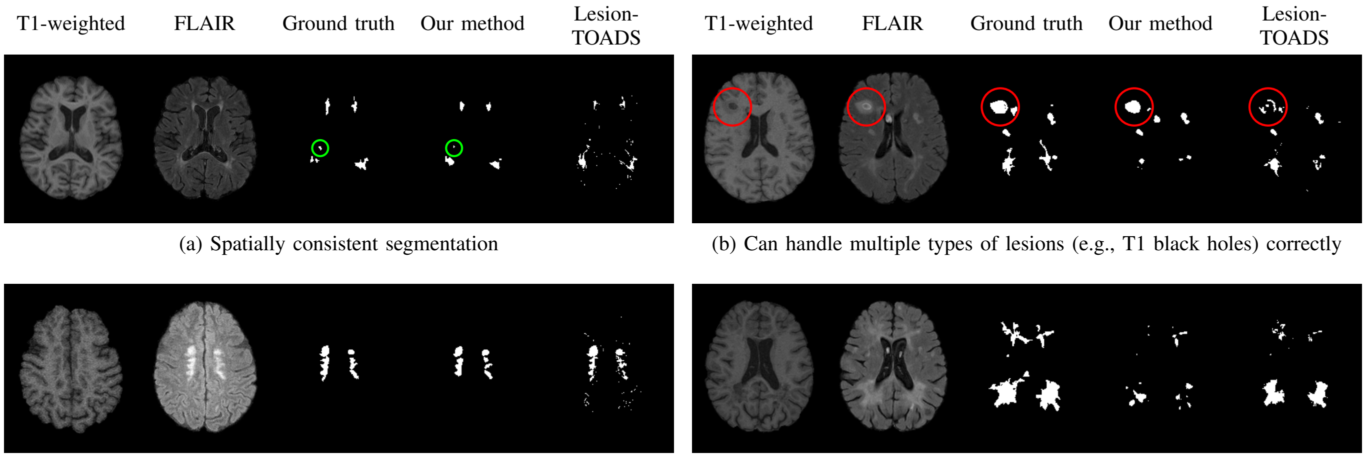
Lesion load	7-layer SCEN			Lesion-TOADS		
	TPR	PPV	DSC	TPR	PPV	DSC
$(0, 1250]$	50.00	41.15	39.34	49.96	13.09	18.86
$(1250, 2500]$	61.92	59.01	57.45	52.39	29.95	37.74
$(2500, 3800]$	57.64	71.54	61.31	54.17	41.83	46.53
$(3800, 10000]$	51.14	81.11	60.13	47.97	56.56	50.76
> 10000	32.82	91.95	48.19	38.88	74.6	50.93

IV. Discussion

We have presented a new method for the automatic segmentation of MS lesions based on multi-layer convolutional encoder networks with shortcut connections. The joint training of the feature extraction and prediction pathways allows for the automatic learning of features at different scales that are tuned for a given combination of image types and segmentation task. We have evaluated our method on a large data set from an MS clinical trial showing that our method is able to segment MS lesions more accurately than Lesion-TOADS, a widely used and freely available method for the automatic segmentation of MS lesions. The gains in accuarcy are mostly due to the reduction of false positives especially for low lesion loads where the lesion size is also small.

The most challenging type of lesions to segment for our method are very large lesions, which can extend beyond the receptive field of a particular voxel. This reduces the network's ability to identify appearance features that are required for the accurate segmentation of lesions. For future work, we are planning to investigate the use of deeper networks. Increasing the depth would allow the network to learn features on a wider range of scales, which we expect will significantly improve the network's ability to segment even very large lesions. In contrast to fully convolutional networks and the u-net architecture, the size of the output segmentation of a CEN is independent of the size of the receptive field, which allows us to design networks that are able to learn features that cover large parts of the image, or even global features that cover the entire image. Such features would be able to estimate the global distribution of lesions and could act as an automatically learned lesion prior.

Another direction for future work is to incorporate prior knowledge directly into the segmentation pipeline. We would like to investigate the use of prior tissue classifications into the learning process. A tissue classification maps can be



(a) Spatially consistent segmentation

(b) Can handle multiple types of lesions (e.g., T1 black holes) correctly

(c) Robust to noise

(d) Underestimates very large lesions

Fig. 4. Four cases illustrating the strengths and weaknesses of our method compared to Lesion-TOADS. Our method is able a) to produce spatially consistent segmentations, b) to handle multiple different types of lesions correctly (e.g., T1 black holes), and c) is inherently robust to noise. (d) Our method might underestimate the size of very large lesions for some cases.

incorporate as additional channels to the input images and provide meaningful guidance for the lesion segmentation process. However, the usefulness of this approach depends greatly on the quality of the provided tissue classification. In the presents of lesions, the tissue classification might fail, which in turn will introduce confounding information into the segmentation process.

Acknowledgements

This work was supported by Natural Sciences and Engineering Research Council of Canada and the Milan and Maureen Ilich Foundation.

References

[1] Tom Brosch and Roger Tam. Efficient training of convolutional deep belief networks in the frequency domain for application to high-resolution 2d and 3d images. *Neural computation*, 2014.

[2] Tom Brosch, Youngjin Yoo, Lisa Y.W. Tang, David K.B. Li, Anthony Traboulsee, and Roger Tam. Deep convolutional encoder networks for multiple sclerosis lesion segmentation. In *A. Frangi et al. (Eds.): MICCAI 2015, Part III, LNCS, vol. 9351*, pages 3–11. Springer, 2015.

[3] D Ciresan, Alessandro Giusti, and J Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in Neural Information Processing Systems*, pages 1–9, 2012.

[4] Yann N Dauphin, Harm de Vries, Junyoung Chung, and Yoshua Bengio. Rmsprop and equilibrated adaptive learning rates for non-convex optimization. *arXiv preprint arXiv:1502.04390*, 2015.

[5] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[6] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[7] Daniel García-Lorenzo, Simon Francis, Sridar Narayanan, Douglas L Arnold, and D Louis Collins. Review of automatic segmentation methods of multiple sclerosis white matter lesions on conventional magnetic resonance imaging. *Medical image analysis*, 17(1):1–18, 2013.

[8] Ezequiel Geremia, Bjoern H Menze, Olivier Clatz, Ender Konukoglu, Antonio Criminisi, and Nicholas Ayache. Spatial decision forests for MS lesion segmentation in multi-channel MR images. In *Jian, T., Navab, N., Pluim, J., Viergever, M. (eds.) MICCAI 2010, Part I. LNCS, vol. 6362*, pages 111–118. Springer, Heidelberg, 2010.

[9] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[10] Geoffrey E. Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, Jul 2006.

[11] Sepp Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. *Diploma, Technische Universität München*, 1991.

[12] Mark Jenkinson, Mickael Pechaud, and Stephen Smith. BET2: MR-based estimation of brain, skull and scalp surfaces. In *Eleventh annual meeting of the organization for human brain mapping*, volume 17. Toronto, ON, 2005.

[13] Kai Kang and Xiaogang Wang. Fully convolutional neural networks for crowd segmentation. *arXiv preprint arXiv:1411.4464*, 2014.

[14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[16] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, 2009.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the 18th International Conferene on Medical Image Computing and Computer Assisted Interventions (MICCAI 2015)*, page 8, 2015.

[18] Dominik Scherer, Andreas Müller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *Artificial Neural Networks–ICANN 2010*, pages 92–101. Springer, 2010.

[19] Navid Shiee, Pierre-Louis Bazin, Arzu Ozturk, Daniel S Reich, Peter A Calabresi, and Dzung L Pham. A topology-preserving approach to the segmentation of brain images with multiple sclerosis lesions. *NeuroImage*, 49(2):1524–1535, 2010.

[20] Jean-Christophe Souplet, Christine Lebrun, Nicholas Ayache, and Grégoire Malandain. An automatic segmentation of T2-FLAIR multiple sclerosis lesions. In *MIDAS Journal - MICCAI 2008 Workshop*, 2008.

[21] Nick Weiss, Daniel Rueckert, and Anil Rao. Multiple sclerosis lesion segmentation using dictionary learning and sparse coding. In *Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013, Part I. LNCS 8149*, pages 735–742. Springer, Heidelberg, 2013.

[22] Youngjin Yoo, Tom Brosch, Anthony Traboulsee, David KB Li, and Roger Tam. Deep learning of image features from unlabeled data for multiple sclerosis lesion segmentation. In *Wu, G., Zhang D., Zhou L. (eds.) MLMI 2014, LNCS, vol. 8679*, pages 117–124. Springer, Heidelberg, 2014.

[23] Matthew D Zeiler. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[24] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In

2011 IEEE International Conference on Computer Vision (ICCV), pages 2018–2025. IEEE, 2011.