Deep 3D Convolutional Encoder Networks with Shortcuts for Multiscale Feature Integration Applied to Multiple Sclerosis Lesion Segmentation

Tom Brosch, Lisa Y.W. Tang, Youngjin Yoo, David K.B. Li, Anthony Traboulsee, and Roger Tam

***Abstract*—We propose a novel segmentation approach based on deep 3D convolutional encoder networks with shortcut connections and apply it to the segmentation of multiple sclerosis (MS) lesions in magnetic resonance images. Our model is a neural network that consists of two interconnected pathways, a convolutional pathway, which learns increasingly more abstract and higher-level image features, and a deconvolutional pathway, which predicts the final segmentation at the voxel level. The joint training of the feature extraction and prediction pathways allows for the automatic learning of features at different scales that are optimized for accuracy for any given combination of image types and segmentation task. In addition, shortcut connections between the two pathways allow high- and low-level features to be integrated, which enable segmentation of lesions across a wide range of sizes. We have evaluated our method on two publicly available data sets with the results showing that our method performs on-par with state-of-the-art methods, even when only relatively small data sets are available for training. In addition, we have evaluation our method on a large data set from an MS clinical trial, with a comparison of network architectures of different depths and with and without shortcut connections. The results show that increasing depth from three to seven layers improves performance, and adding shortcut connections further increases accuracy. Overall, our method demonstrates consistently strong segmentation performance across a wide range of lesion loads, and in a direct comparison outperforms EMS, LST-LGA, and Lesion-TOADS, three widely used and freely available automatic MS lesion segmentation methods.**

***Index Terms*—Segmentation, multiple sclerosis lesions, magnetic resonance imaging (MRI), deep learning, convolutional neural networks, machine learning**

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 [12], 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 of, e.g., a subject specific generative model of tissue intensities [29], [31], [38], [40], or a generative model of image patches representing a healthy population [41]. Alternatively, clustering methods have been used to segment healthy and lesion tissue, where lesions are modelled as a separate tissue class [32], [36]. In many methods, spatial priors of healthy tissues are used to reduce false positives. For example, in addition to modelling MS lesions as a separate cluster, Lesion-TOADS [32] employs a topological and a statistical atlas to produce a topology-preserving segmentation of all brain tissues, while the expectation-maximization segmentation (EMS) [40] method uses a Markov random field (MRF) to produce a spatially consistent segmentation. To account for local changes of the tissue intensity distributions, Tomas-Fernandez et al. [38] combined the subject-specific model of the global intensity distributions with a voxel-specific model calculated from a healthy population, where lesions are detected as outliers of the combined model. A major challenge of unsupervised methods is that outliers may not be specific to lesions and can also be caused by intensity inhomogeneities, partial volumes, imaging artifacts, and small anatomical structures such as blood vessels, which leads to the generation of false positives. To overcome, Roura et al. [29] employed an additional set of rules to remove false positives, while Schmidt et al. [31] used a conservative threshold for the initial detection of lesions, which are later grown to yield an accurate delineation.

Current supervised approaches typically start with a simple or large set of features, either predefined by the user [13], [14], [34] or gathered in a feature extraction step such as by deep learning [42]. Voxel-based segmentation algorithms [13], [42] feed the features and labels of each voxel into a general classification algorithm, such as a random forest [2], to classify each voxel and to determine which set of features are the most important for segmentation in the particular domain. MRF-based approaches [34], [35] incorporate voxel features and the

T. Brosch and Y. Yoo are 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: brosch.tom@gmail.com, youngjin@msmri.medicine.ubc.ca)

A. Traboulsee is with the Multiple Sclerosis/Magnetic Resonance Imaging Research Group, Division of Neurology, The University of British Columbia, Vancouver, BC V6T 2B5, Canada (e-mail: t.traboulsee@ubc.ca)

L.Y.W. Tang, D.K.B. Li and R. Tam are 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: lisat@sfu.cu, david.li@ubc.ca, roger.tam@ubc.ca)

labels of neighboring voxels to produce a spatially consistent segmentation. To further reduce false positives, Subbanna et al. [34] combined the voxel-level MRF with a regional MRF, which integrates a large set of intensity and textural features extracted from the regions produced by the voxel-level MRF with the labels of neighboring regions. Library-based approaches leverage a library of pre-segmented images to carry out the segmentation. For example, Guizard et al. [14] proposed a segmentation method based on an extension of the non-local means algorithms [8]. The centers of patches at every voxel location are classified based on matched patches from a library containing pre-segmented images, where multiple matches are weighted using a similarity measure based on rotation-invariant features.

A recent breakthrough for automatic segmentation using deep learning comes from the domain of cell membrane segmentation, in which Ciresan et al. [6] proposed classifying the centers of image patches directly using a convolutional neural network (CNN) [24] 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, which allows the learning of features that are specifically tuned to the segmentation task. However, the time required to train patch-based methods can make the approach infeasible when the size and number of patches are large.

Recently, different CNN architectures [4], [20], [26], [28] have been proposed that are able to feed through entire images, 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. introduced the fully convolutional neural network (fCNN) for the segmentation of crowds in surveillance videos [20]. However, fCNNs produce segmentations of lower resolution than the input images due to the successive use of convolutional and pooling layers, both of which reduce the dimensionality. To predict segmentations of the same resolution as the input images, we recently proposed using a 3-layer convolutional encoder network (CEN) [4] for MS lesion segmentation. The combination of convolutional [24] and deconvolutional [44] layers allows our network to produce segmentations that are of the same resolution as the input images.

Another limitation of the traditional CNN is the trade-off between localization accuracy, represented by lower-level features, and contextual information, provided by higher-level features. To overcome, Long et al. [26] proposed fusing the segmentations produced by the lower layers of the network with the upsampled segmentations produced by higher layers. However, using only low-level features was not sufficient to produce a good segmentation at the lowest layers, which is why segmentation fusing was only performed for the three highest layers. Instead of combining the segmentations produced at different layers, Ronneberger et al. [28] proposed combining the features of different layers to calculate the final segmentation directly at the lowest layer using an 11-layer u-shaped network architecture called u-net. Their network is composed of a traditional contracting path (first half of the u), but augmented with an expanding path (last half of the u), which replaces the pooling layers of the contracting path with upsampling operations. To leverage both high- and low-level features, shortcut connections are added between corresponding layers of the two paths. However, upsampling cannot fully compensate for the loss of resolution, and special handling of the border regions is still required.

We propose a new convolutional network architecture that combines the advantages of a 3-layer CEN [4] and a u-net [28]. Our network is divided into two pathways, a traditional convolutional pathway, which consists of alternating convolutional and pooling layers, and a deconvolutional pathway, which consists of alternating deconvolutional and unpooling layers and predicts the final segmentation. Similar to the u-net, we introduce shortcut connections between layers of the two pathways. In contrast to the u-net, our network uses deconvolution instead of upsampling in the expanding pathway and predicts segmentations that have the same resolution as the input images and therefore does not require special handling of the border regions. We have evaluated our method on two widely used publicly available data sets for the evaluation of MS lesion segmentation methods and a large proprietary data set from an MS clinical trial, with a comparison of network architectures of different depths and with and without shortcuts[1]. The results will be presented in detail in Section III.

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 binary lesion masks S, where 1 denotes a lesion voxel and 0 denotes a non-lesion voxel. 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, $\mathcal{S}$, is modeled by the convolutional encoder network with shortcut connections (CEN-s) illustrated in Fig. 1. A CEN-s is a type of convolutional neural network (CNN) [24] that is divided into two interconnected pathways, the convolutional pathway and the deconvolutional [44] 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

[1] Where the risk of confusion is minimal, we will refer to the shortcut connections between two corresponding layers as a single shortcut (see Fig. 1).

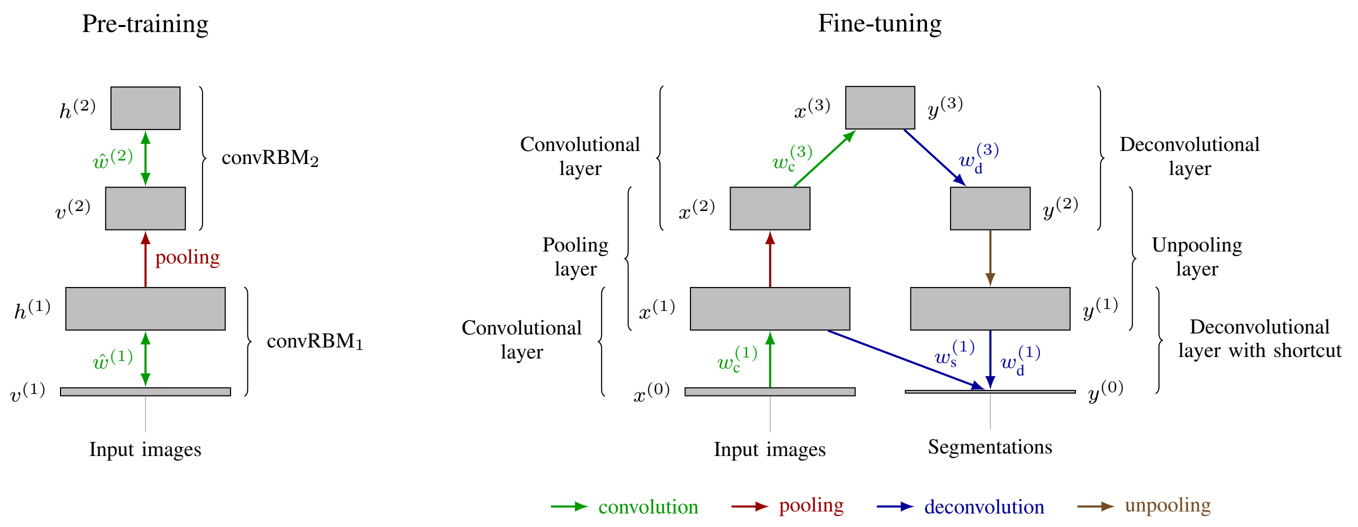

Fig. 1. Pre-training and fine-tuning of the 7-layer convolutional encoder network with shortcut that we used for our experiments. Pre-training is performed on the input images using a stack of convolutional RBMs. The pre-trained weights and bias terms are used to initialize a convolutional encoder network,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

convolutional layers automatically learn a feature hierarchy from the input images. A convolutional layer is a deterministic function of the following form

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

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 $w_{\mathrm{c},ij}^{(l)}$, 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, i.e., $\tilde{w}(a) = w(-a)$. To be consistent with the inference rules of convolutional restricted Boltzmann machines (convRBMs) [25], which are used for pre-training, convolutional layers convolve the input signal with flipped filter kernels, while deconvolutional layers calculate convolutions with non-flipped filter kernels. We use rectified linear units [27] in all layers except for the output layers, which have shown to improve the classification performance of CNNs [23]. A convolutional layer is followed by an average pooling layer [30] 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 deconvolutional and unpooling layers with shortcut connections to the corresponding convolutional layers. The first deconvolutional layer uses the extracted features of the convolutional pathway to calculate abstract segmentation features

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

where $y^{(L)} = x^{(L)}$, L denotes the number of layers of the convolutional pathway, $w_{\mathrm{d},ij}^{(L)}$ and $c_i^{(L-1)}$ are trainable parameters of the deconvolutional layer, and $\circledast$ denotes full convolution. To be consistent with the general notation of deconvolutions [44], the non-flipped version of w is convolved with the input signal.

Subsequent deconvolutional layers use the activations of the previous layer and corresponding convolutional layer to calculate more localized segmentation features

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

where l is the index of a deconvolutional layer with shortcut, and $w_{\mathrm{s},ij}^{(l+1)}$ are the shortcut 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

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

where we use the sigmoid function defined as $\operatorname{sigm}(z) = (1 + \exp(-z))^{-1}, z \in \mathbb{R}$ instead of the rectified linear function in order to obtain a probabilistic segmentation with values in the range between 0 and 1. To produce a binary lesion mask from the probabilistic output of our model, we chose a fixed threshold such that the mean Dice similarity coefficient [10] is maximized on the training set and used 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 for each sample of the training set, which requires the calculation of the gradient of E with respect to the model parameters [24]. Typically, neural networks are trained by minimizing the sum of squared differences (SSD), which can be calculated for a single image as follows

$$E = \frac{1}{2} \sum_{\vec{p}} \left(S(\vec{p}) - y^{(0)}(\vec{p}) \right)^2, \tag{6}$$

where $\vec{p} \in \mathbb{N}^3$ are the coordinates of a particular voxel. The partial derivatives of the error with respect to the model parameters can be calculated using the delta rule and are given by

$$\frac{\partial E}{\partial w_{\mathrm{d},ij}^{(l)}} = \delta_{\mathrm{d},i}^{(l-1)} * \tilde{y}_j^{(l)}, \qquad \frac{\partial E}{\partial c_i^{(l)}} = \sum_{\vec{p}} \delta_{\mathrm{d},i}^{(l)}(\vec{p}), \tag{7}$$

$$\frac{\partial E}{\partial w_{\mathrm{s},ij}^{(l)}} = \delta_{\mathrm{d},i}^{(l-1)} * \tilde{x}_j^{(l)}, \tag{8}$$

$$\frac{\partial E}{\partial w_{\mathrm{c},ij}^{(l)}} = x_i^{(l-1)} * \tilde{\delta}_{\mathrm{c},j}^{(l)}, \text{ and } \qquad \frac{\partial E}{\partial b_i^{(l)}} = \sum_{\vec{p}} \delta_{\mathrm{c},i}^{(l)}(\vec{p}). \tag{9}$$

For the first layer, $\delta_{\mathrm{d},1}^{(0)}$ can be calculated by

$$\delta_{\mathrm{d},1}^{(0)} = \big(y_1^{(0)} - S\big) y_1^{(0)} \big(1 - y_1^{(0)}\big). \tag{10}$$

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

$$\delta_{\mathrm{d},j}^{(l)} = \big(\tilde{w}_{\mathrm{d},ij}^{(l)} * \delta_{\mathrm{d},i}^{(l-1)}\big)\mathbb{I}\big(y_j^{(l)} > 0\big), \tag{11}$$

$$\delta_{\mathrm{c},i}^{(l)} = \big(w_{\mathrm{c},ij}^{(l+1)} \circledast \delta_{\mathrm{c},j}^{(l+1)}\big)\mathbb{I}\big(x_i^{(l)} > 0\big), \tag{12}$$

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

$$\delta_{\mathrm{c},j}^{(l)} = \big(\delta_{\mathrm{c},j}^{(l)\prime} + \tilde{w}_{\mathrm{s},ij}^{(l)} * \delta_{\mathrm{d},i}^{(l-1)}\big)\mathbb{I}\big(x_j^{(l)} > 0\big), \tag{13}$$

where $\delta_{\mathrm{c},j}^{(l)\prime}$ denotes the activation of unit $\delta_{\mathrm{c},j}^{(l)}$ 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 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:

$$\begin{aligned} E = {} & r \frac{\sum_{\vec{p}} \big(S(\vec{p}) - y^{(0)}(\vec{p})\big)^2 S(\vec{p})}{\sum_{\vec{p}} S(\vec{p})} \\ & + (1-r) \frac{\sum_{\vec{p}} \big(S(\vec{p}) - y^{(0)}(\vec{p})\big)^2 \big(1 - S(\vec{p})\big)}{\sum_{\vec{p}} \big(1 - S(\vec{p})\big)}. \end{aligned} \tag{14}$$

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 based on preliminary experimental results [4], the algorithm is stable with respect to changes in r, which largely affects the threshold used to binarize the probabilistic output. A detailed evaluation of the impact of the sensitivity ratio on the learned model is presented in Section III-D.

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 $\delta_{\mathrm{d},1}^{(0)}$. 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

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

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:

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

C. Training

At the beginning of the training procedure, the model parameters need to be initialized and the choice of the initial parameters can have a big impact on the learned model [37]. In our experiments, we found that initializing the model using pre-training [16] on the input images was required in order to be able to fine-tune the model using the ground truth segmentations without getting stuck in an early local minimum. Pre-training can be performed layer by layer [15] using a stack of convRBMs (see Fig. 1), thereby avoiding the potential problem of vanishing or exploding gradients [17]. The first convRBM is trained on the input images, while subsequent convRBMs are trained on the hidden activations of the previous convRBM. After all convRBMs have been trained, the model parameters of the CEN-s can be initialized as follows (showing the first convolutional and the last deconvolutional layers only, see Fig. 1)

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

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

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.

A major challenge for gradient-based optimization methods is the choice of an appropriate learning rate. Classic stochastic gradient descent [24] 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 can require different learning rates. In recent years, there has been an increasing interest in developing methods for automatically choosing independent learning rates. Most methods (e.g., AdaGrad [11], AdaDelta [43], RMSprop [9], and Adam [22]) 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, where the optimal learning rates often differ greatly for each layer. In our initial experiments, networks obtained by training with AdaDelta, RMSprop, and Adam performed comparably well, but AdaDelta was the most robust to the choice of hyperparameters, so we used AdaDelta for all results reported.

D. Implementation

Pre-training and fine-tuning were performed using highly optimized GPU-accelerated implementations of 3D convRBMs and CENs that performs training in the frequency domain [3]. Our frequency domain implementation significantly speeds up the training by mapping the calculation of convolutions to simple element-wise multiplications, while adding only a small number of Fourier transforms. This is especially important for the training on 3D volumes, due to increased number of weights of 3D kernels compared to 2D. Internal tests have shown that our frequency domain implementation calculates the most time-consuming operations of the training procedure 6 times faster than an implementation based on cuDNN [5], a library for calculating deep learning primitives, which is used internally by many publicly available deep learning frameworks [1], [7], [19].

III. Experiments and Results

We evaluated our method on two publicly available data sets, which allows for the direct comparison with state-of-the-art methods. In addition, we have used a very challenging data set containing four different MRI sequences of relapsing-remitting MS patients from a multi-center MS clinical trial, which represents the large variability in lesion size, shape, location and intensity as well as varying contrasts produced by different scanners. The trial data set was used to carry out a detailed analysis of different CEN architectures using different combinations of modalities, with a comparison to publicly available state-of-the-art methods.

A. Data Sets and Pre-processing

1) Trial data set: The data set was collected from 67 different scanning sites using different $1.5\,\mathrm{T}$ and $3\,\mathrm{T}$ scanners for a clinical trial in relapsing-remitting MS, and consists of 377 T1-weighted (T1w), T2-weighted (T2w), proton density-weighted (PDw), and FLAIR MRIs from 195 subjects. The image dimensions are $256 \times 256 \times 60$ voxels with 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) [18], followed by an intensity normalization to the interval $[0, 1]$, and a 6 degree-of-freedom intra-subject registration. To speed-up the training, all images were cropped to a $164 \times 206 \times 52$ voxel region-of-interest with the brain roughly centered. The ground truth segmentations were produced using an existing semiautomatic 2D region-growing technique, which has been used successfully in a number of large MS clinical trials (e.g., [21], [39]). To carry out the segmentation, each lesion was manually identified by an experienced radiologist and then interactively grown from the seed point by a trained technician.

We divided the data set into a training ($n = 250$), validation ($n = 50$) and test set ($n = 77$) such that images of each set were acquired from different scanning sites. The three data sets were used for training our model and parameter tuning of the competing methods, for monitoring the training progress, and to carry out a detailed analysis of variants of our method and competing methods, respectively. Pre-training and fine-tuning of a 7-layer CEN-s took approximately 27 hours and 37 hours, respectively, on a single GeForce GTX 780 graphics card. However, once the network is trained, new multi-contrast images can be segmented in less than one second.

2) Public data sets: The data set of the MICCAI 2008 MS lesion segmentation challenge [33] consists of 43 T1w, T2w, and FLAIR MRIs, divided into 20 training cases for which ground truth segmentation are made publicly available, and 23 test cases. After training the model on the 20 training cases, we used the trained model to segment the 23 test cases, which were send to the challenge organizers for evaluation.

In addition, we evaluated our method on the T1w, T2w, PDw, and FLAIR MRIs of the 21 publicly available labeled cases from the ISBI 2015 Longitudinal MS lesion segmentation challenge. The challenge was not open for new submissions at the time of writing this article. Therefore, we evaluated our method on the training. To allow for a direct comparison with the results of the second and third placed methods, we followed their evaluation protocol and performed leave-one-subject-out cross-validation.

B. Competing methods

We compared our method with four publicly available methods that are widely used (e.g., [14], [34], [36]) as a reference point for the comparison of MS lesion segmentation methods: a) the expectation maximization segmentation (EMS) method [40], b) the lesion growth algorithm (LST-LGA) [31] as implemented in the LST toolbox version 2.0.11, c) the lesion prediction algorithm (LST-LPA) also implemented in the LST toolbox, and d) Lesion-TOADS version 1.9 R [32]. The Lesion-TOADS method has no tunable parameters, so we used the default parameters to carry out the segmentation using T1w and FLAIR MRIs. The performance of EMS depends on the choice of the Mahalanobis distance κ, the threshold t used to binarize the probabilistic segmentation, and the modalities used. We carried out the segmentation using two combinations of modalities: a) the modalities used in the original paper [40] (T1w, T2w, and PDw), and b) all four available modalities (T1w, T2w, PD2, FLAIR). We compared the segmentations produced for all combinations of $\kappa = 2, 2.2, \ldots, 4.6$ and $t = 0.05, 0.1, \ldots, 1$ with the ground truth segmentations on the training set and chose the values that maximized the average DSC ($\kappa = 2.6, t = 0.75$; $\kappa = 2.8, t = 0.9$). We used the same procedure to tune the initial threshold κ of LST-LGA using T1w and FLAIR MRIs for $\kappa = 0.05, 0.1, \ldots, 1$ and the threshold t used to binarize the probabilistic segmentations produced by LST-LPA also using T1w and FLAIR MRIs for $t = 0.05, 0.1, \ldots, 1$. The optimal parameters were $\kappa = 0.1$ and $t = 0.45$, respectively.

C. Measures of Segmentation Accuracy

Lisa has kindly agreed to update this section (thanks Lisa!). It will include DSC, lesion TPR, lesion FPR, and volume difference.

We used three different measures to evaluate segmentation accuracy, with the primary measure being the Dice similarity coefficient (DSC) [10], which computes a normalized overlap value between the produced and ground truth segmentations, and is defined as

$$\text{DSC} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}, \tag{19}$$

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 underestimation 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 lesion regions in the ground truth that are correctly identified by an automatic method. It is defined as

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{20}$$

where a value of $100\,\%$ indicates that all true lesion voxels are correctly 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 fraction of true lesion voxels out of all identified lesion voxels

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{21}$$

where a value of $100\,\%$ indicates that all voxels that are classified as lesion voxels are indeed lesion voxels as defined by the ground truth (no false positives).

D. Setting the Training Parameters

The most important parameters of the training method are the number of epochs and the sensitivity ratio. Fig. 2 shows the mean DSC evaluated on the training and validation set during training of a 7-layer CEN-s for increasing number of epochs. The mean DSC scores keep improving even after 400 epochs, albeit as a much slower rate. The optimal number of epochs is a trade-off between accuracy and time required for training. Due to the relatively small improvements after 400 epochs, we decided to stop the training procedure after 500 epochs. Due to the relatively small size of the challenge data sets, we did not use a dedicated validation set to choose the number of epochs. Instead, we set the number of epochs to 2500, which results in roughly the same number of gradient updates compared to the trial data set.

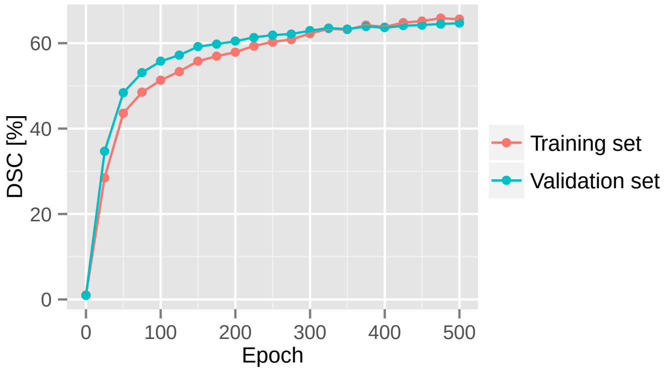

Fig. 2. Development of mean DSC on the training and test set during training. Only small improvements can be observed after 500 epochs.

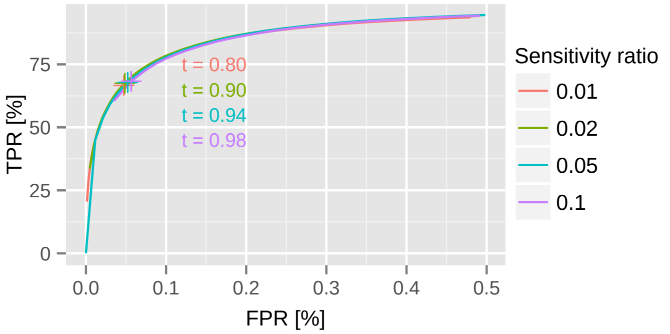

Fig. 3. ROC curves for different sensitivity ratios r. A plus marks the TPR and FPR of the optimal threshold. The ROC curves for different sensitivity ratios are almost identical and only causes a change of the optimal threshold t, which shows the robustness of our method with respect to the sensitivity ratio.

Fig. 3 shows a set of ROC curves for different choices of the sensitivity ratio ranging from 0.01 to 0.1. A plus marks the TPR and FPR after thresholding using a threshold that maximizes the DSC on the training set. The plots confirm our initial findings that our method is not sensitive to the choice of the sensitivity ratio, which mostly affects the optimal threshold. We chose a fixed sensitivity ration of 0.02 for all our experiments.

E. Comparison on Public Data Sets

To allow for a direct comparison with other state-of-the-art methods, we have evaluated our method on the MICCAI 2008 MS lesion segmentation challenge [33] and the ISBI 2015 longitudinal MS lesion segmentation challenge. In our previous paper [4], we showed that approximately 100 images are required to train the 3-layer CEN without overfitting and we expect the required number of images to be even higher when adding more layers. Due to the relatively small size of the training data sets provided by the two challenges, we trained a CEN with only 3 layers on these data sets, to reduce the risk of overfitting. The parameters of the models are summarized in Table I.

TABLE I
PARAMETERS OF THE 3-LAYER CEN FOR THE EVALUATION ON THE CHALLENGE DATA SETS.

Layer type	Kernel Size	#Filters	Image Size
Input	—	—	$164 \times 206 \times 156 \times c$
Convolutional	$9 \times 9 \times 9 \times c$	32	$156 \times 198 \times 148 \times 32$
Deconvolutional	$9 \times 9 \times 9 \times 32$	1	$164 \times 206 \times 156 \times 1$

Note: The number of input channels c is 3 for the MICCAI challenge and 4 for the ISBI challenge.

TABLE II
SELECTED METHODS OUT OF THE 52 ENTRIES SUBMITTED FOR EVALUATION TO THE MICCAI 2008 MS LESION SEGMENTATION CHALLENGE.

Rank	Method	Score	LTPR	LFPR	VD
1	Jesson et al.	86.94	48.70	28.25	80.15
2	Guizard et al. [14]	86.11	49.85	42.75	48.80
4	Tomas-Fernandez et. al [38]	84.46	46.90	44.60	45.60
6	Our method	84.07	51.55	51.25	57.75
11	Roura et al. [29]	82.34	50.15	41.85	111.60
13	Geremia et al. [13]	82.07	55.10	74.10	48.90
24	Shiee et al. [32]	79.90	52.40	72.70	74.45

Note: Only the best entry per method is shown for multiple submission. Columns LTPR, LFPR, and VD show the mean scores of the two raters in percent. Last updated: Dec 2, 2015.

A comparison of our method with other state-of-the-art methods evaluated on the MICCAI challenge test data set is summarized in Table II. Our method ranked 6th out of 52 entries submitted to the challenge, outperforming popular methods such as SLS by Roura et al. [29], the random forest approach by Geremia et al. [13], and Lesion-TOADS by Shiee et al. [32], but performing worse than the patch-based segmentation approach by Guizard et al. [14], or the MOPS approach by Tomas-Fernandez et al. [38], which used additional images to build the intensity model of a healthy population. This is a very impressive result given the simplicity of the used model, the relatively small training set size, and because we have not tuned our method for this particular data set in contrast to most other methods, for which multiple entries were submitted to the challenge.

In addition, we evaluated our method on the 21 publicly available labeled cases from the ISBI 2015 longitudinal MS lesion segmentation challenge. The challenge organizers have not yet released the challenge results on their website. Therefore, we have only compared our method to methods, who have given sufficient details about the test produce and a summary of the mean DSC, LTPR and LFPR scores for both raters to allow for a direct comparison. Following the evaluation protocol of the second and third best method of the challenge, we trained our model using leave-one-subject-out cross-validation on the training images and compared our results to the segmentations provided by both raters. Table III summarizes the performance of our method and competing methods as well as the performance of the two rates when compared against each other. Although our method performs slightly worse the the second and third best method on the challenge, it produces DSC scores that are in the range of the competing methods. Furthermore, our method is more sensitive to lesions then the other methods, but also produces more false positives.

TABLE III
COMPARISON OF OUR METHOD WITH THE SECOND AND THIRD RANKED METHODS FROM THE ISBI MS LESION SEGMENTATION CHALLENGE.

Method		Rater 1			Rater 2	
	DSC	LTPR	LFPR	DSC	LTPR	LFPR
Rater 1	—	—	—	73.2	64.5	17.4
Rater 2	73.2	82.6	35.5	—	—	—
Jesson et al.	70.4	61.1	13.5	68.1	50.1	12.7
Maier et al. (GT1)	70	53	48	65	37	44
Maier et al. (GT2)	70	55	48	65	38	43
Our method (GT1)	68.4	74.5	54.5	64.4	63.0	52.8
Our method (GT2)	68.3	78.3	64.5	65.8	69.3	61.9

Note: The evaluation was performed on the training set using leave-one-subject-out cross-validation. GT1 and GT2 denote that the model was trained with the segmentations provided by the first and second rater as the ground truth, respectively.

F. Comparison of Network Architectures, Input Modalities, and Competing Methods

To determine the effect of network architecture and input modalities, we compared the segmentation performance of five different networks. Specifically, we trained a 3-layer CEN and two 7-layer CENs, one with shortcut connections and one without, on T1w and FLAIR MRIs, and two additional 7-layer CEN-s on the modalities used by EMS (T1w, T2w, PDw) and all four modalities (T1w, T2w, PDw, FLAIR). The parameters of the networks are given in Table IV and Table V. To roughly compensate for the anisotropic voxel size of the input images, we chose an anisotropic filter size of $9 \times 9 \times 5$. We also included the four competing methods discussed in Section III-B. A comparison of the segmentation accuracy of the trained networks and competing methods is summarized in Table VI. All CEN architectures performed significantly better than the best performing competing method LST-LGA in overall segmentation accuracy, where the improvements of the mean DSC scores ranged from 3 percentage points (pp) for the 3-layer CEN to 17 pp for the 7-layer CEN with shortcut trained on all four modalities. The improved segmentation performance was mostly due to an increase in lesion sensitivity. LST-LGA achieved a mean lesion TPR of only $37.50\,\%$, whereas the CEN with shortcut achieved a mean lesion TPR of $54.55\,\%$ when trained on the same modalities, and a mean lesion TPR of 62.49 when trained on all four modalities, while achieving a comparable number of lesion false positives. The mean lesion FPRs and mean volume differences of LST-LGA and the 7-layer CEN-s were roughly the same, when trained on the same modalities, and could be further reduced when trained on different modalities.

This experiment also showed that increasing the depth of the CEN and adding the shortcut connections improves the segmentation accuracy. Increasing the depth of the CEN from three layers to seven layers improved the mean DSC by 3 pp.

TABLE IV
PARAMETERS OF THE 3-LAYER CEN USED TO EVALUATE 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 V
PARAMETERS OF THE 7-LAYER CEN-S USED TO EVALUATE 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	$164 \times 206 \times 52 \times 1$

The improvement was confirmed to be statistically significant using a one-sided paired t-test (p-value $= 1.3 \times 10^{-5}$). Adding a shortcut to the network further improved the segmentation accuracy as measured by the DSC by 3 pp. A second one-sided paired t-test was performed to confirm the statistical significance of the improvement with a p-value of less than 1×10^{-10}.

The impact of increasing the depth of the network on the segmentation performance of very large lesions is illustrated in Fig. 4, where the true positive, false negative, and false positive voxels are highlighted in green, yellow, and red, respectively. 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 marked by two white circles. 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, with and without shortcut, is able to learn a feature set that captures large lesions much better than the 3-layer CEN, which results in an improved segmentation. However, increasing the depth of the network without adding shortcut connections reduces the networks sensitivity to very small lesions as illustrated in Fig. 5. In this example, the 3-layer CEN was able to detect three small lesions, indicated by the white circles, which were missed by the 7-layer CEN. Adding shortcut connections enables our model to learn a feature set that spans a wider range of lesion sizes, which increases the sensitivity to small lesions and, hence, allows the 7-layer CEN-s to detect all three small lesions (highlighted by the white circles), while still being able to segment large lesions (see Fig. 4).

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

Method	DSC [%]	LTPR [%]	LFPR [%]	VD [%]
Input modalities: T1w and FLAIR				
3-layer CEN [4]	49.24	57.33	61.39	43.45
7-layer CEN	52.07	43.88	29.06	37.01
7-layer CEN-s	55.76	54.55	38.64	36.30
Lesion-TOADS [32]	40.04	56.56	82.90	49.36
LST-LGA [31]	46.64	37.50	38.06	36.77
LST-LPA [31]	46.07	48.02	52.94	41.62
Input modalities: T1w, T2w, and PDw				
7-layer CEN-s	61.18	52.00	36.68	29.38
EMS [40]	42.94	44.80	76.58	49.29
Input modalities: T1w, T2w, FLAIR, and PDw				
7-layer CEN-s	63.83	62.49	36.10	32.89
EMS [40]	39.70	49.08	85.01	34.51

Note: The table shows the mean of the Dice similarity coefficient (DSC), lesion true positive rate (LTPR), and lesion false positive rate (LFPR). Because the volume difference (VD) is not limited to the interval $[0, 100]$, a single outlier can heavily affect the calculation of the mean. We therefore excluded outliers before calculating the mean of the VD for all methods.

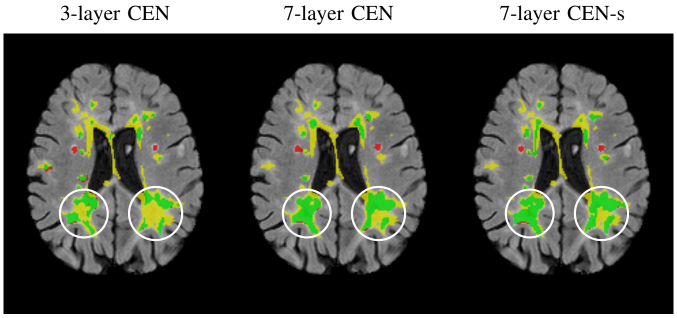

Fig. 4. Large lesion problematic for 3-layer CEN due to limited size of the receptive field. The adding layers increases the size of the receptive field, which improves the detection of very large lesions.

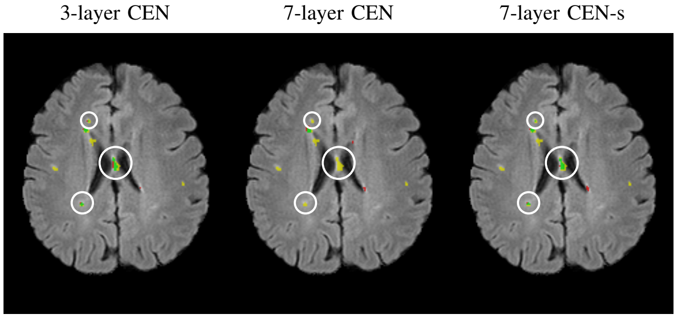

Fig. 5. Comparison of segmentation performance of different CEN architectures for small lesions. The white circles indicate lesions that were detected by the 3-layer CEN and the 7-layer CEN with shortcut. Increasing the size of the receptive field decreases the sensitivity to small lesions. The addition of a shortcut allows the detection of small lesions, while still being able to detect large lesions (see Fig. 4).

G. Comparison for Different Lesion Sizes

To examine the effect of lesion size on segmentation performance, we stratified the test set into five groups based on their mean reference lesion size as summarized in Table VII. A comparison of segmentation accuracy and lesion detection measures of a 7-layer CEN-s trained on different input modalities and the best performing competing method LST-LGA for different lesion sizes is illustrated in Fig. 6. The 7-layer CEN-s outperforms LST-LGA for all lesions sizes except for

TABLE VII
LESION SIZE GROUPS AS USED FOR THE DETAILED ANALYSIS.

Group	Mean lesion size [mm^3]	#Samples	Lesion load [mm^3]
Very small	$[0, 70]$	6	1457 ± 1492
Small	$(70, 140]$	24	4298 ± 2683
Medium	$(140, 280]$	24	$12\,620 \pm 9991$
Large	$(280, 500]$	14	$13\,872 \pm 5814$
Very large	> 500	9	$35\,238 \pm 27\,531$

very large lesions when trained on T1w and FLAIR MRIs, and for all lesion sizes when trained on all four modalities. The differences are larger for smaller lesions, which are generally more challenging to segment for all methods. The differences between the two approaches are caused by a higher sensitivity to lesions as measured by the lesion TPR, especially for smaller lesions, while producing approximately the same number of false positives for all lesion sizes.

IV. DISCUSSION

The automatic segmentation of MS lesions is a very challenging task due to the large variability in lesion size, shape, intensity, and location, as well as the large variability of imaging contrasts produced by different scanners used at different scanning sites. Most unsupervised methods model lesions as an outlier or a separate cluster in a subject-specific model, which makes them inherently robust to inter-subject and inter-scanner variability. However, outliers may not be specific to lesions and can also be caused by intensity inhomogeneities, partial volumes, imaging artifacts, and small anatomical structures such as blood vessels, which leads to the generation of false positives. On the other hand, supervised methods can learn to discriminate between lesion and non-lesion tissue, but are more sensitive to the variability in lesion appearance and different contrasts produced by different scanners. To overcome those challenges, supervised methods require large data sets that span the variability in lesions and careful pre-processing to match the imaging contrast of new images with those of the training set. Library-based approaches have shown great promise for the segmentation of MS lesions, but do not scale well to very large data sets due to the large amount of memory required to store comprehensive sample libraries and the time required to scan such libraries for matching patches. On the other hand, parametric deep learning models such as convolutional neural networks scale much better to large training sets, because the size required to store the model is independent of the training set size, and the operations required for training and inference are inherently parallelizable, which allows them to take advantage of very fast GPU-accelerated computing hardware. Furthermore, the combination of many nonlinear processing units allows them to learn features that are robust under large variability, which is crucial for the segmentation of MS lesions.

Convolutional neural networks were originally designed to classify entire images and designing networks that can segment images remains an important research topic. Early approaches have formulated the segmentation problem as a patch-wise classification problem, which allows them to directly use established classification network architectures for image segmentation. However, a major limitation of patch-based deep learning approaches is the time required for training and inference. Fully convolutional networks can perform the segmentation much more efficiently, but lack the precision to perform voxel-accurate segmentation and can not handle unbalanced classes.

To overcome these challenges, we have presented a new method for the automatic segmentation of MS lesions based on deep 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. Shortcuts between the two pathways allow high- and low-level features to be leveraged at the same time for more consistent performance across scales. In addition, we have proposed a new objective function based on the combination of sensitivity and specificity, which makes the objective function inherently robust to unbalanced classes such as MS lesions, which typically comprise less than $1\,\%$ of all image voxels. We have evaluated our method on two publicly available data sets and a large data set from an MS clinical trial, with the results showing that our method performs on-par with state-of-the-art methods, even for relatively small training set sizes, and is able to segment MS more accurately than widely-used competing methods such as EMS, LST-LGA, and Lesion-TOADS, when a suitable training set is available. The substantial gains in accuracy were mostly due to an increase in lesion sensitivity, especially for small lesions. Overall, the CEN with shortcuts architecture performed consistently well over a wide range of lesion sizes.

Our segmentation framework is very flexible and can be easily extended. One such extension could be to incorporate prior knowledge about the tissue type of each non-lesion voxel into the segmentation procedure. The probabilities of each tissue class could be precomputed by a standard segmentation method, after which they can be added as an additional channel to the input units of the CEN, which would allow the CEN to take advantage of intensity information from different modalities and prior knowledge about each tissue class to carry out the segmentation. In addition, our method can be applied to other segmentation tasks. Although we have only focused on the segmentation of MS lesions in this paper, our method does not make any assumptions specific to MS lesion segmentation. The features required to carry out the segmentation are solely learned from training data, which allows our method to be used to segment different types of pathology or anatomy when a suitable training set is available.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

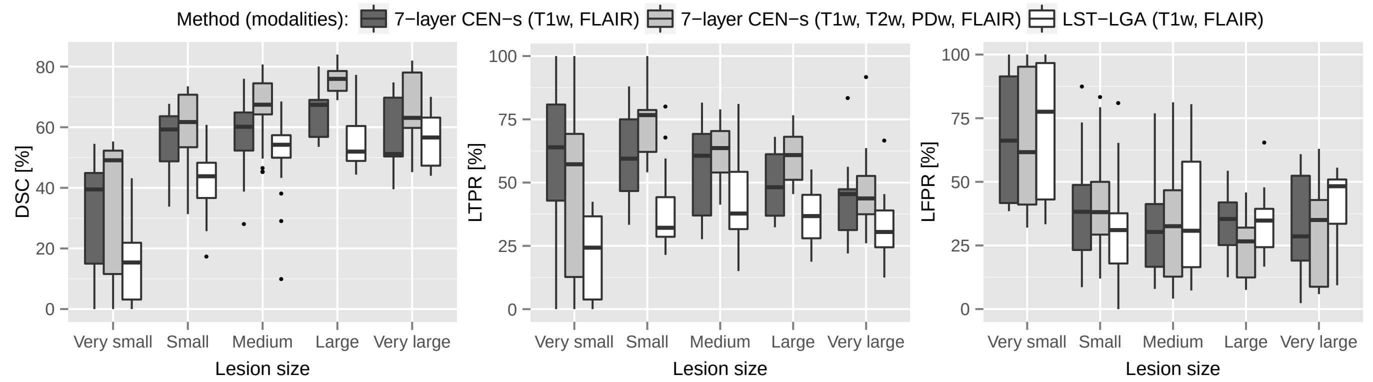

Fig. 6. Comparison of segmentation accuracy and lesion detection measures of a 7-layer CEN-s trained on different input modalities and the best performing competing method LST-LGA for different lesion sizes. The 7-layer CEN-s outperforms LST-LGA for all lesions sizes except for very large lesions when trained on T1w and FLAIR MRIs, and for all lesion sizes when trained on all four modalities, due to a higher sensitivity to lesions, while producing approximately the same number of false positives. Outliers are denoted by black dots.

[2] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] 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.

[4] 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.

[5] Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. cuDNN: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014.

[6] 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.

[7] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.

[8] Pierrick Coupé, José V Manjón, Vladimir Fonov, Jens Pruessner, Montserrat Robles, and D Louis Collins. Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation. *NeuroImage*, 54(2):940–954, 2011.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] Nicolas Guizard, Pierrick Coupé, Vladimir S Fonov, Jose V Manjón, Douglas L Arnold, and D Louis Collins. Rotation-invariant multi-contrast non-local means for MS lesion segmentation. *NeuroImage: Clinical*, 8:376–389, 2015.

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

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

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

[18] 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.

[19] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

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

[21] L Kappos, A Traboulsee, C Constantinescu, J-P Erälinna, F Forrestal, P Jongen, J Pollard, Magnhild Sandberg-Wollheim, C Sindic, B Stubinski, et al. Long-term subcutaneous interferon beta-1a therapy in patients with relapsing-remitting MS. *Neurology*, 67(6):944–953, 2006.

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

[23] Alex Krizhevsky, I Sutskever, and G Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1–9, 2012.

[24] 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.

[25] 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.

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[27] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th Annual International Conference on Machine Learning*, pages 807–814, 2010.

[28] 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.

[29] Eloy Roura, Arnau Oliver, Mariano Cabezas, Sergi Valverde, Deborah Pareto, Joan C Vilanova, Lluís Ramió-Torrentà, Àlex Rovira, and Xavier Lladó. A toolbox for multiple sclerosis lesion segmentation. *Neuroradiology*, 57(10):1031–1043, 2015.

[30] 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.

[31] Paul Schmidt, Christian Gaser, Milan Arsic, Dorothea Buck, Annette Förschler, Achim Berthele, Muna Hoshi, Rüdiger Ilg, Volker J Schmid, Claus Zimmer, et al. An automated tool for detection of FLAIR-hyperintense white-matter lesions in multiple sclerosis. *NeuroImage*, 59(4):3774–3783, 2012.

[32] 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.

[33] Martin Styner, Joohwi Lee, Brian Chin, M Chin, Olivier Commowick, H Tran, S Markovic-Plese, V Jewells, and S Warfield. 3D segmentation in the clinic: A grand challenge II: MS lesion segmentation. *MIDAS Journal - MICCAI 2008 Workshop*, pages 1–6, 2008.

[34] Nagesh Subbanna, Doina Precup, Douglas Arnold, and Tal Arbel. IMaGe: Iterative multilevel probabilistic graphical model for detection and segmentation of multiple sclerosis lesions in brain MRI. In *Information Processing in Medical Imaging*, pages 514–526. Springer, 2015.

[35] NK Subbanna, M Shah, SJ Francis, S Narayanan, DL Collins, DL Arnold, and T Arbel. Ms lesion segmentation using markov random fields. In *Proceedings the MICCAI 2009 Workshop on Medical Image Analysis on Multiple Sclerosis*, pages 1–12, 2009.

[36] Carole Sudre, M Jorge Cardoso, Willem Bouvy, Geert Biessels, Josephine Barnes, and Sebastien Ourselin. Bayesian model selection for pathological neuroimaging data applied to white matter lesion segmentation. *IEEE Transactions on Medical Imaging*, 34(10):2079–2102, 2015.

[37] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pages 1139–1147, 2013.

[38] Xavier Tomas-Fernandez and Simon Keith Warfield. A model of population and subject (MOPS) intensities with application to multiple sclerosis lesion segmentation. *IEEE Transactions on Medical Imaging*, 34(6):1349–1361, 2015.

[39] A Traboulsee, A Al-Sabbagh, R Bennett, P Chang, DKB Li, et al. Reduction in magnetic resonance imaging T2 burden of disease in patients with relapsing-remitting multiple sclerosis: analysis of 48-week data from the EVIDENCE (EVidence of Interferon Dose-response: European North American Comparative Efficacy) study. *BMC Neurology*, 8(1):11, 2008.

[40] Koen Van Leemput, Frederik Maes, Dirk Vandermeulen, Alan Colchester, and Paul Suetens. Automated segmentation of multiple sclerosis lesions by model outlier detection. *IEEE Transactions on Medical Imaging*, 20(8):677–688, 2001.

[41] 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.

[42] 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.

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

[44] 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.