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 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 Lesion-TOADS, a widely used and freely available automatic MS lesion segmentation method. We found the main limitation of our model to be the underestimation of very large lesions, but from our depth comparison we expect that this problem could be solved in future work by adding more network layers.**

***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], [37], [39], or a generative model of image patches representing a healthy population [40]. Alternatively, clustering methods have been used to segment healthy and lesion tissue, where lesions are modelled a separate tissue class [32], [35]. 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) [39] 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. [37] 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], [33] or gathered in a feature extraction step such as by deep learning [41]. Voxel-based segmentation algorithms [13], [41] 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 [33], [34] 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. [33] 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 [43] 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. 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 [43] 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

[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).

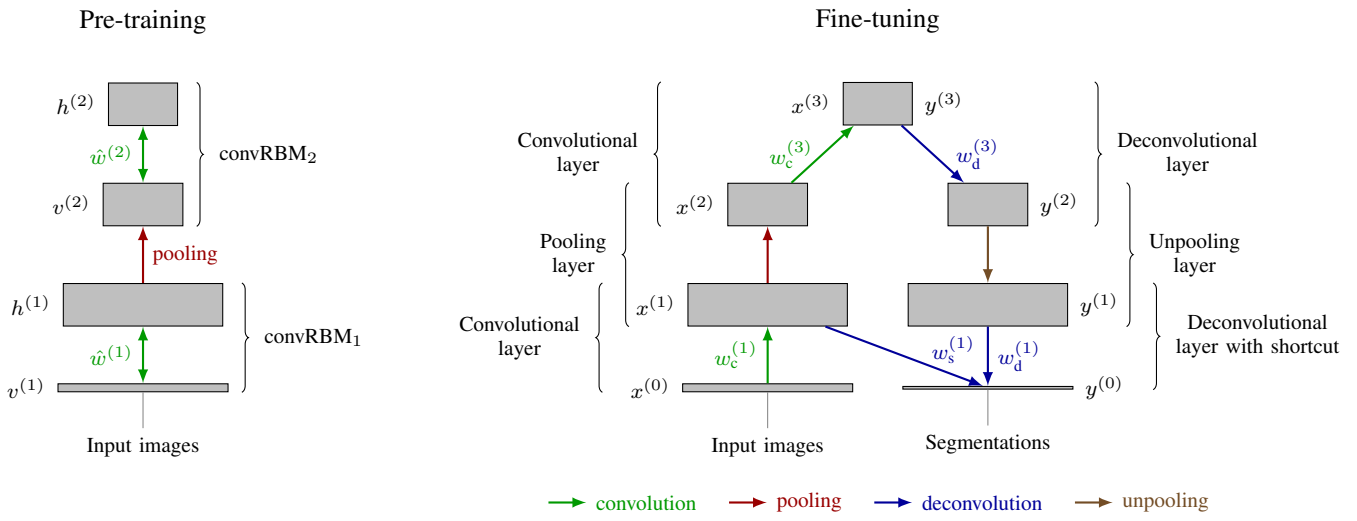


Fig. 1. Pre-training and fine-tuning of the 7-layer convolutional encoder network with shortcut that we used for our experiments. Pre-training is performed on the input images using a stack of convolutional RBMs. The pre-trained weights and bias terms are used to initialize a convolutional encoder network,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

from the input images. A convolutional layer is a deterministic function of the following form

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

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 $w_{\mathrm{c},ij}^{(l)}$, 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

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

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 [43], 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

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

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

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

where we use the sigmoid function defined as $\mathrm{sigm}(z) = (1 + \exp(-z))^{-1}, z \in \mathbb{R}$ 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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. We used a sensitivity ratio of 0.02 in all our experiments.

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

$$E = \frac{1}{2} \sum_{\vec{p}} \big(\alpha S(\vec{p}) + \beta(1 - S(\vec{p}))\big) \Big(S(\vec{p}) - y^{(0)}_1(\vec{p})\Big)^2. \tag{15}$$

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, $\delta^{(0)}_{\mathrm{d},1}$ can be derived analogously to the SSD case, and the new factor is simply carried over:

$$\delta^{(0)}_{\mathrm{d},1} = \big(\alpha S + \beta(1 - S)\big) \big(y^{(0)}_1 - S\big) y^{(0)}_1 \big(1 - y^{(0)}_1\big). \tag{16}$$

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 [36]. 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)

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

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

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 [42], 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 secondary progressive 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, compared against competing segmentation methods.

A. Data Sets and Pre-processing

a) Proprietary data set: The data set, acquired from 67 different scanning sites, consists of 377 T1-weighted (T1w), T2-weighted (T2w), proton density-weighted (PDw), and FLAIR images from 195 subjects with a resolution of $256 \times 256 \times 60$ voxels and a voxel size of $0.936\,\text{mm} \times 0.936\,\text{mm} \times 3.000\,\text{mm}$. 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], [38]). 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 training and test set such that images of the two sets were aquired from different clinical sites.

We used 300 image pairs for pre-training and fine-tuning, and the remaining 77 image pairs for evaluation. 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 image pairs can be segmented in less than one second. We compared our results to the lesion masks produced by Lesion-TOADS [32], a widely used tool for segmenting MS lesions.

b) MSLSC: Describe the data. From website.

c) LMSLSC: Describe the data. From website.

B. Competing methods

Short explanation of competing methods and how the parameters were determined.

C. Measures of Segmentation Accuracy

Update to reflect new measures

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).

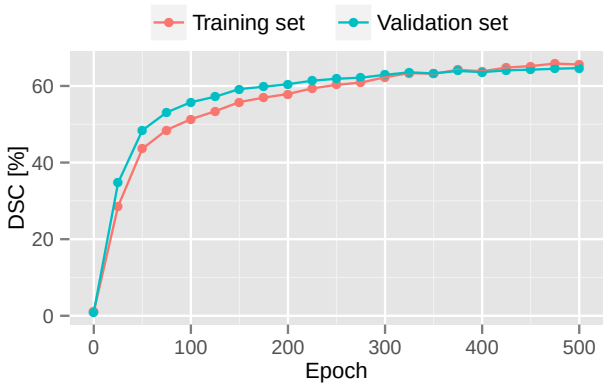


Fig. 2. Development of mean DSC on the training and test set during training. Only small improvements can be observed after 500 epochs.

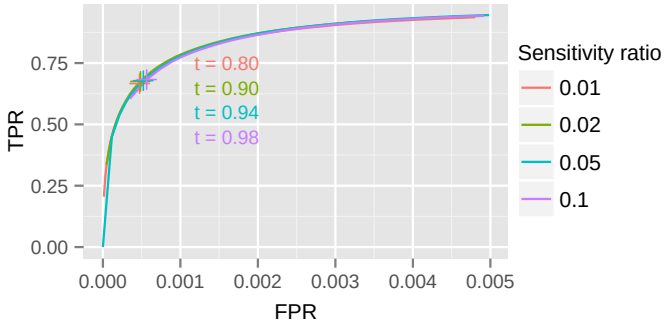


Fig. 3. ROC curves for different sensitivity ratios r. A plus marks the TPR and FPR of the optimal threshold. The ROC curves for different sensitivity ratios are almost identical and only causes a change of the optimal threshold t, which shows the robustness of our method with respect to the sensitivity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

D. Setting the Hyperparameters

Parameter are summarized in Table ??. For the challenge data sets, we choose an isotropic filter size of $9 \times 9 \times 9$. To roughly compensate for the unisotropic voxel size of the in-house data, we also chose unisotropic a filter size of $9 \times 9 \times 5$. The number of epochs used for training is a trade-off between accuracy and time required for training. Figure ?? shows the mean DSC on the training set of 250 images and the validation set of 50 images for increasing number of epochs during training. The mean DSC on the training and validation set keeps improving even after 400 epochs, while the improvements become smaller as the training progresses. After 500 epochs, the DSC reaches almost a flat point, so we decided to stop training after 500 epochs. For the experiments, we chose the number of epochs to be 2500, which keeps the number of gradient updates roughly the same compared to training on the in-house data set. No further tuning was performed to carry out the results on the publicly available data sets.

We found that our method is not sensitive to the sensitivity ratio r. To confirm, we have calculated ROC curves on the validation set for different choices of r. Also plotted with the threshold marked in the plot, confirming that the parameter r mostly affects the choice of the threshold used to binarize the segmentation, but does not affect the performance of the model for choices between 0.01 and 0.1. Consequently, we have used the same r for all experiments.

E. Comparison on Public Data Sets

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 MICCAI and ISBI challenges, we trained a CEN with only 3 layers on these data sets.

I will have a comparison on the ISBI challenge training set using leave-one-subject-out cross-validation. To reduce the risk of overfitting, only trained 3-layer CEN. Describe the data: 5 subjects with 4 time points each. Winning method did not describe the evaluation process well enough to allow for an accurate production of the experiments. Therefore, only compare against the second and third best method. Following they protocol, we performed leave-one-subject out cross-validation and trained separate models on the GT masks produced by rater 1 and 2 and compared the model against the masks from rater 1 and 2, which yields 4 experiments in total: a) trained on rater 1 and compared against rater 1, b) trained on rater 1 and compared against rater 2, c) trained on rater 2 and compared against rater 1, and d) trained on rater 2 and compared against rater 2. Or, for simplicity, only trained on the masks of rater 1 and compared against rater 1 and rater 2 masks. Only 4 subjects per training fold. To reduce the risk of overfitting, only trained the 3-layer CEN. Table ?? shows a comparison with the other methods.

TABLE I
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.

We have also evaluated our method on the MICCAI 2008 lesion segmentation challenge. Data set consists of T1w, T2w, and FLAIR MRIs of 20 subjects. Due to the relatively small size of the training data set, we have trained a 3-layer CEN similar to the ISBI challenge and applied the model to the test data set 1. The evaluation was performed remotely by the challenge organizers. We placed 5th in the challenge. The results of the top 8 methods are summarized in Table II. Ranked 6th out of 52 entries.

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 [37]	84.46	46.90	44.60	45.60
6	Our method	84.07			
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.

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

Mention also different modalities used and update to reflect new measures and competing methods.

To determine the effect of network architecture, we compared the segmentation performance of three different networks. Specifically, we trained a 3-layer CEN and two 7-layer CENs, one with shortcut connections and one without, on the 300 training pairs. The parameters of the networks are given in Table IV and Table V. Lesion-TOADS was also included in the comparison as a known standard. A comparison of the segmentation accuracy of the trained networks and Lesion-TOADS is summarized in Table VI. All CEN architectures performed significantly better than Lesion-TOADS in overall segmentation accuracy, where the improvements of the mean DSC scores ranged from 9 pts for the 3-layer CEN to 14 pts for the 7-layer CEN with shortcut. The improved segmentation performance was mostly due to a reduction of the false positives. Lesion-TOADS achieved a mean PPV of only $39.86\,\%$, whereas the CEN with shortcut achieved a mean PPV of $66.58\,\%$—an improvement of 27 pts. The mean TPRs were roughly the same (slightly less than $50\,\%$) for all methods except for the 7-layer CEN with shortcut, which performed slightly better than the other methods with a mean TPR of $52.75\,\%$.

This experiment 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 2 pts. The improvement was confirmed to be statistically significant using a one-sided paired t-test (p-value $= 0.002$). Adding a shortcut 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 $< 1 \times 10^{-10}$.

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

Layer type	Kernel Size	#Filters	Image Size
Input	—	—	$164 \times 206 \times 156 \times 2$
Convolutional	$9 \times 9 \times 9 \times 2$	32	$156 \times 198 \times 148 \times 32$
Deconvolutional	$9 \times 9 \times 9 \times 32$	1	$164 \times 206 \times 156 \times 1$

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	$164 \times 206 \times 52 \times 1$

G. Comparison for Different Lesion Sizes

Update groups table to show lesion sizes. Also redo the plots for lesion sizes. Give a better justification for dividing by lesion size. Small lesions can be more difficult to miss, while large lesions also come with some problems.

To examine the effect of lesion load on segmentation performance, we stratified the test set into five groups based on their reference lesion loads as summarized in Table VII. Most segmentation performance measures deteriorate with lower lesion loads, because when there are only a few true lesion voxels, even small segmentation errors can translate into large relative errors. A comparison of segmentation accuracy of a 3-layer CEN and a 7-layer CEN with shortcut for different lesion loads is illustrated in Fig. **??**. Adding four layers and shortcut connections improved the segmentation accuracy for all lesion load groups, where the improvements were largest for the highest lesion loads. In MS, 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

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	$164 \times 206 \times 52 \times 1$

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]	49.10	49.22	42.71	41.30
7-layer CEN	51.04	39.57	25.88	40.82
7-layer CEN-s	54.02	48.74	35.85	39.70
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 [39]	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 [39]	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.

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

Group	Lesion load [mm^3]	#Samples	Lesion size [mm^3]
Very low	$[0, 3250]$	17	
Low	$(3250, 6500]$	16	
Medium	$(6500, 10000]$	18	
High	$(10000, 25000]$	18	
Very high	> 25000	8	

lesion loads, where larger lesions are also more prevalent.

Fig. **??** shows a comparison of the 7-layer CEN with shortcut and Lesion-TOADS. The CEN approach outperformed Lesion-TOADS for all lesion load groups, except the group with very high lesion loads, where Lesion-TOADS achieved a slightly higher mean DSC than the CEN approach, but the difference is much smaller than the gains in accuracy achieved by the CEN for the other lesion loads. The 7-layer CEN with shortcut also performed more consistently across lesion load groups, whereas Lesion-TOADS decreased more strongly in performance for the smaller lesion loads. As a result, the greatest differences between the two methods are seen in the lowest lesion load groups. Table **??** shows a more detailed comparison. While the PPV increased consistently with higher lesion loads for both methods, the TPR was highest for low to medium lesion loads and decreased 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.

H. Qualitative Results

Update images to show good, mean and bad segmentation. It's more about showing the range of difficulty in the data set from easy images to difficult images. If time permits, show some filters and say something about them.

A qualitative comparison of segmentation performance for four characteristic cases is shown in Fig. 6. Our method uses a combination of automatically learned intensity and appearance features, which makes it inherently robust to noise (see Fig. 6a), while still being able to detect small isolated lesions (see Fig. 6b). Furthermore, our method is able to learn a wide spectrum of lesion shapes and appearances from training data, which allows our method to correctly identify multiple different types of MS lesions. For example, our method was able to correctly identify a large T1 black hole that was partially missed by Lesion-TOADS (see Fig. 6c), which has a known limitation of sometimes misclassifying T1 black holes due to different intensity profiles of partially overlapping T1 hypointense and T2 hyperintense regions [32]. Figure 6d shows one of the most challenging cases for our method. Very large lesions can extend beyond the size of the receptive field of the CEN, which reduces its ability to extract characteristic lesion features. Consequently, in some cases our method can underestimate the size of very large lesions.

IV. DISCUSSION

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. In addition, shortcuts between the two pathways allow high- and low-level features to be leveraged at the same time for more consistent performance across scales. We have evaluated our method on a large data set from an MS clinical trial, with the results showing that our method is able to segment MS more accurately than the widely used Lesion-TOADS. The substantial gains in accuracy were mostly due to the reduction of false positives, especially for low lesion loads where the lesion size is also small. Overall, the CEN with shortcuts architecture performed consistently well over a wide range of lesion loads. For future work, we are planning to extend our comparison to other freely available lesion segmentation tools, such as the lesion growth algorithm [31] as implemented in the LST toolbox (www.statistical-modelling.de/lst.htm).

The most significant limitation of the tested architecture is that very large lesions can still extend beyond the receptive field of a particular voxel. This reduces the network's ability to extract appearance features that would help the identification of lesion voxels. For future work, we are planning to investigate the use of deeper networks for increasing accuracy for very large lesions. This work would require greater training times and a larger sample of scans with high lesion loads, but we expect it to 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

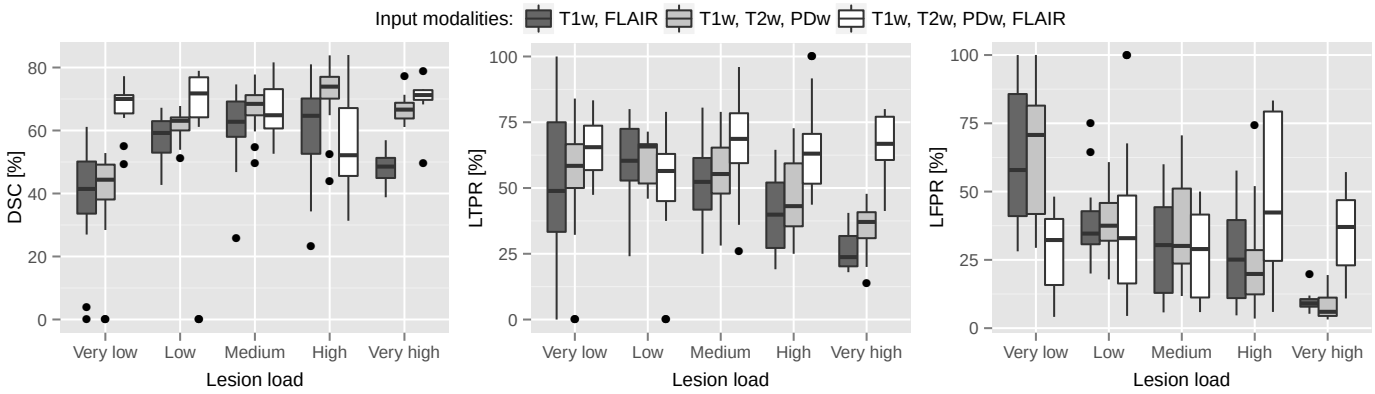


Fig. 4. Comparison of modalities. No single modality is the most important. Combining all four modalities gives significantly better results. Outliers are denoted by small black circles.

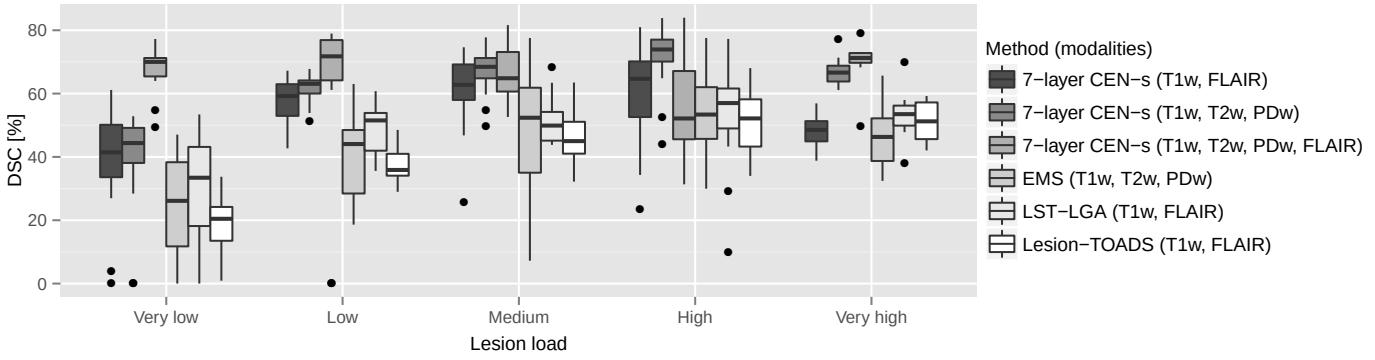
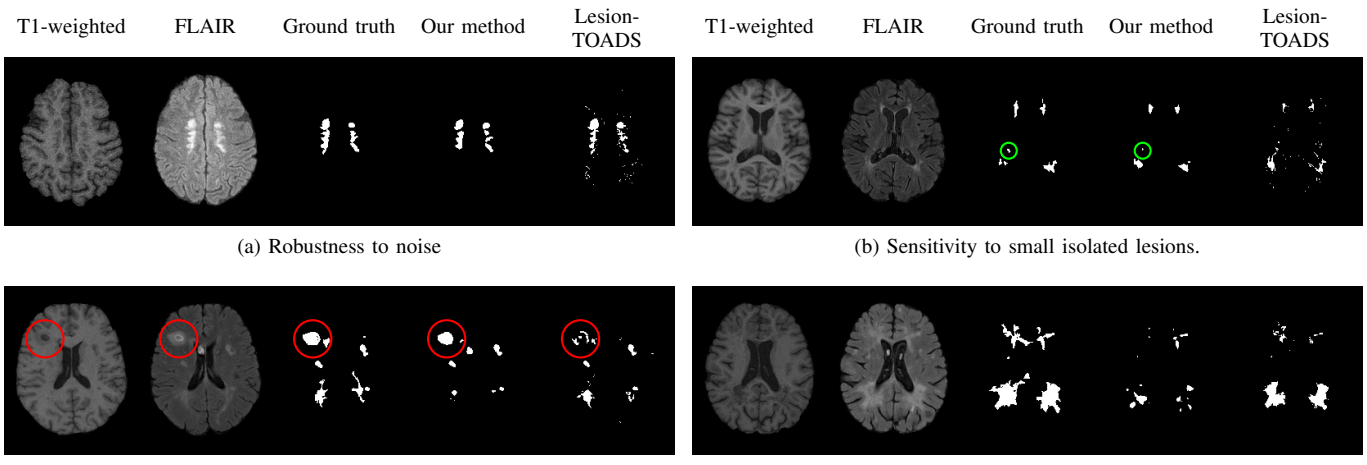


Fig. 5. Comparison of CEN architestures with state-of-the-art methods. CEN is better for all combinations of modalities. The 4 modality CEN significantly outperforms the other methods. Outliers are denoted by small black circles.



(a) Robustness to noise

(b) Sensitivity to small isolated lesions.

(c) Example of a T1 black hole that was correctly identified by our method

(d) Very large lesions can be underestimated

Fig. 6. Four cases illustrating the strengths and limitations of our method compared to Lesion-TOADS. Our method is inherently robust to noise (a), while still being able to detect small isolated lesions (b). Furthermore, our method is able to detect multiple different types of lesions correctly (e.g., T1 black holes). However, in some cases our method can underestimate the size of very large lesions (d).

act as an automatically learned lesion prior, further improving the robustness of our method.

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.

[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] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

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

[43] 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.