



Convolutional Neural Network Based Segmentation of Demyelinating Plaques in MRI

Bartłomiej Stasiak¹, Paweł Tarasiuk¹, Izabela Michalska²,
Arkadiusz Tomczyk¹(✉), and Piotr S. Szczepaniak¹

¹ Institute of Information Technology, Lodz University of Technology,
Wolczanska 215, 90-924 Lodz, Poland

{bartlomiej.stasiak,pawel.tarasiuk,arkadiusz.tomczyk,
piotr.szczepaniak}@p.lodz.pl

² Department of Radiology, Barlicki University Hospital,
Kopcinskiego 22, 91-153 Lodz, Poland
izabela.anna.michalska@gmail.com

Abstract. In this paper a new architecture of convolutional neural networks is proposed. It is a fully-convolutional architecture which allows to keep the size of the processed image constant. This, in consequence, allows to apply it for image segmentation tasks where for a given image a mask representing sought regions should be produced. An additional advantage of this architecture is its ability to learn from smaller images which reduces the amount of data that must be propagated through the network. The trained network can be still applied to images of any size. The proposed method was used for automatic localization of demyelinating plaques in head MRI sequences. This work was possible, which should be emphasized, only thanks to the manually outlined plaques provided by radiologist. To present characteristic of the considered approach three architectures and three result evaluation methods were discussed and compared.

Keywords: Multiple sclerosis · Segmentation · Machine learning
Convolutional neural networks

1 Introduction

In recent years, thanks to the technological progress (computations with GPU) and growing access to large amount of labeled data, convolutional neural networks (CNN) achieved outstanding success in automatic analysis of the images containing scenes from the surrounding world. However, in the case of specialized, e.g. medical, images the advance is not that evident. The main reason is the lack of sufficiently large annotated training sets. Gathering of such data is hard because the group of domain experts able to annotate images is relatively small and the amount of data that must be analyzed may be bigger than it is

in the case of traditional images (e.g. 3D sequences). It is even harder in case of image segmentation task where every structure needs to be described with details which makes this process extremely time-consuming [1,2]. That is why it must be emphasized that research presented in this paper was only possible thanks to the hard work of radiologist who precisely outlined the regions of interest, demyelinating plaques, on every slice of head MRI sequence.

A typical application of CNNs is the whole image content classification task. In the case of image segmentation two basic approaches can be found in the literature. First one is a modified sliding window technique where CNN is used as a part of the classifier. In this case, however, the label is not assigned to the whole image but to the selected regions of that image (in particular to the regions representing neighbourhood of a given pixel). Consequently also to train such a classifier smaller image fragments, cut from the images manually annotated by an expert, are taken. Such a method was used, for example, in segmentation of anatomical regions in MRI images [3]. The second approach is so-called fully convolutional approach [4]. Here, the whole image constitutes the network input and as an output the mask, of the same size as an input image, describing object localization is expected. To gain such a result some modification must be made in CNN structure. Typical architecture contains convolutional and pooling layers which reduce the size of the intermediate results. That is why some new, upscaling (deconvolutional) layers need to be added to restore the original image size. And although this approach requires CNN modifications its advantage is fact that it can be trained using whole input images and expected masks without the need of cutting them into smaller fragments. This kind of approach was successfully used in e.g. analysis of transmitted light microscopy images [5] and MRI prostate examinations [6]. The latter approach is particularly interesting since it considers 3D convolution and the 3D MRI sequence is processed by CNN as a whole.

The solution proposed in this work to some extent possesses features of both above approaches. On the one hand, it tries to train CNN to act as a non-linear filter capable of indicating areas of interest. Consequently the output is the image of the same size as the input. In this case, however, no upscaling (deconvolutional) layers are required. On the other hand, it allows to train such a network using smaller image fragments without the necessity of processing as large amount of data as needed for the training based on the whole images.

The paper is organized as follows: the second section describes the considered dataset and medical background justifying the importance of demyelinating plaques localization, in the third section the proposed method is discussed and in the fourth and the fifth section the obtained results and their analysis are presented. Finally, the last section contains a short summary of the conducted research.

2 Medical Background

Multiple sclerosis (MS) is a common, chronic disease involving the central nervous system and leading progressively to different degrees of neurological

disability. In multiple sclerosis cells of the human immune system attack myelin sheaths of the nerve fibres which represent white matter in the brain and spinal cord. The consequence of the myelin damage is inflammation in the affected areas and then creating scar tissue. This process is known as demyelination and the afflicted areas within the nerves' sheathes are called demyelinating plaques. To diagnose MS the combination of clinical symptoms, typical history, cerebrospinal fluid examination and magnetic resonance imaging (MRI) of the central nervous system is required. MRI plays an important role in diagnosing MS as it enables not only to confirm the diagnosis and defining its pattern but also to assess the progress of the disease and the response to treatment. It is essential to know which areas of the brain are affected because the process of demyelination as well as some other lesions in the white matter could also be present in different neurological disorders. White matter lesions in MS occur in some characteristic locations. Thus most of the lesions appear typically in juxtacortical regions (that is close to the brain cortex), periventricularly (that is around the ventricles and these lesions tend to lay perpendicularly to the long axis of the lateral ventricles), in corpus callosum, cerebellum (within hemispheres and cerebellar peduncles) and peripherally in brainstem (that is in cerebral peduncles, pons and medulla oblongata).

T2-weighted images (T2WI) are MR scans which are the most sensitive in showing the white matter lesions that are presented as areas of a high signal (that means they are hyperintense and appear white on the images) in regard to normal white matter. More sensitive than conventional T2WI in detecting juxtacortical and periventricular lesions are FLAIR (fluid-attenuated inversion recovery) sequences because they suppress the signal of fluid, including cerebrospinal fluid which fills the ventricles and subarachnoidal space. As a result the cerebrospinal fluid has a low signal and appears black on the scans obtained in FLAIR technique, as compared to the white matter abnormalities which remain hyperintense. On conventional T2WI cerebrospinal fluid presents high signal like demyelinating lesions in the white matter thus it may be difficult to recognize plaques localized in the vicinity of the ventricles and juxtacortical areas. On the FLAIR images the contrast between cerebrospinal fluid and the white matter lesions disposed in its proximity is more clearer and makes the plaques can be better detected.

The present study is based on indicating demyelinating lesions in the white matter on head MR images. All MR scans chosen to the study were performed in FLAIR sequences, in axial plane, with 3 to 5 mm slices using 1,5 Tesla scanner. The study comprised a hundred patients with confirmed diagnosis of MS, including fifty men and fifty women in the age range between 19 and 66 years old and in various stages of the disease. All noticeable changes of the signal intensity within the white matter were considered as demyelinating lesions.

3 Method

Convolutional neural networks are biologically inspired [7] machine learning techniques, where the input has a form of a finite-dimensional linear space range.

They can be treated as a modification of multi-layer perceptron (MLP) with weights sharing and reduced connections between layers. As opposed to MLP, where the permutation of inputs does not influence the training process, in CNN the structure of input data is important and remains unaffected while processing. This and proper weights sharing cause that processing in CNN is translation invariant. Typically in CNN as an input images are given after optional initial preprocessing (scaling, normalization, etc.) [8]. The outputs of the hidden layers are called *feature maps* [9, 10] since they describe the actual localization of some image features.

An usual application area of CNN is image classification. A typical approach assumes that CNN performs some reduction of input image size which gives image representation that is later processed by some general-purpose classifier - MLP is preferred here since the whole CNN+MLP architecture can be trained at the same time using gradient based optimization methods [10]. Many of the winning solutions in ImageNet Large Scale Visual Recognition Challenge [11] are based on such architectures [8, 12, 13]. Some of those solutions were later successfully applied for other image recognition tasks [14]. And although classification is a typical application, there are also research works where CNN acts as a feature extractor [15] or is used directly for object localization [16, 17].

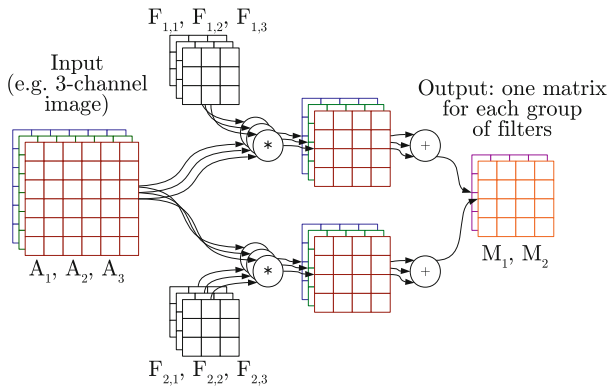


Fig. 1. Structure of the convolutional layer. Sample input matrices A_1, A_2, A_3 are processed with 2 groups of $F_{i,j}$ filters (3 filters in each group). The results produced by each filter group are summed up constituting separate output matrices: M_1, M_2 . Image originally published in [18].

Taking into account structure of feature maps, it is possible to define object localization as a task of generating a specific feature map. It requires, however, pure convolutional architecture without a classifier, since it destroys information about spatial structure. Moreover, to obtain such a feature map, which can be easily interpreted as an object localization mask, it must be ensured that input and output dimensions are the same. That is why the approach proposed in this

work, instead of reducing the size of feature maps, keeps their dimensions constant. The details and consequences of that approach are described in Sect. 3.2. Thanks to the normalization of the CNN output (e.g. using unipolar sigmoid function), a fuzzy mask is created where the value assigned to each point can be interpreted as a probability of its belonging to the object. Further processing (noise removal, thresholding) leads to a binary mask which is useful in some applications. Our approach to thresholding is described in Sect. 3.3.

3.1 Formal Description

To describe a *convolutional layer*, the basic unit of CNN, let us denote the input data as a tuple of matrices $A_1 \dots A_p$ of a fixed $n_a \times m_a$ size (for the first layer it could be for example a multi-channel digital image). The key element of the layer are q *filter groups* where each group is a tuple of p matrices of $n_f \times m_f$ size ($F_{i,j}$ for $i = 1 \dots p, j = 1 \dots q$). The output is a tuple of *feature maps* $M_1 \dots M_q$ where for each $i = 1 \dots q$

$$M_i = Z_i + \left(\sum_{j=1}^p \right) A_j * F_{i,j}.$$

The Z_i used in the formula above is a bias matrix of the same size as M_i . Matrix convolution $A_j * F_{i,j}$ is a matrix of elements $(A_j * F_{i,j})_{r,c}$ for $r = 1 \dots (n_a) - (n_f) + 1, c = 1 \dots (m_a) - (m_f) + 1$ such that

$$(A_j * F_{i,j})_{r,c} = \left(\sum_{d_n=0}^{n_f-1} \right) \left(\sum_{d_m=0}^{m_f-1} \right) (F_{i,j})_{(n_f-d_n), (m_f-d_m)} \cdot (A_j)_{(r+d_n), (c+d_m)}.$$

The resulting M_i matrices size is $n_a - n_f + 1 \times m_a - m_f + 1$ (Fig. 1).

Since the output is a tuple of matrices it can be processed by the next convolutional layer. However, since the matrix convolution with a fixed $F_{i,j}$ is a linear transformation, it is recommended to apply some non-linear activation function between those layers for every element of the output matrices. Although the sigmoid-like functions are known to work here, it is recommended to use ReLU (rectified linear unit) [8] or PReLU (parametrized extension of ReLU) [19] functions. Additionally, in typical applications, the maximum- or average-pooling layers are used between the convolutional layers. This reduces the matrix dimensions by a certain factor [9]. In our task this would be counterproductive, and that is why such pooling layers are not used.

If the size of filters is other than 1×1 , A_j matrices have different size than M_i . To overcome that problem we add zero-padding to the A_j to increase the input size to $(n_a + n_f - 1) \times (m_a + m_f - 1)$. This, of course, does not lead to any information loss since using padding of the proposed size makes it possible to construct the identity operator ($F_{i,j}$ of odd dimensions with 1 in a central element and 0 everywhere else). Moreover, the padding size (adding $(n_f - 1)$

rows and $(m_f - 1)$ columns) is independent from the input size – it is related only to the filter size.

Each element of convolutional layer output is a result of processing some $n_f \times m_f$ rectangle taken from each A_j . For the first feature map, $n_f \times m_f$ is a size of *visual field* [7]. For further layers, the size of visual fields can be easily calculated by tracking down the range of CNN input pixels affecting each output element. Should the network consist of convolutional layers and element-wise operations only, the visual field size would be $n_z \times m_z$ where $n_z = (n_{f_1} + \dots + n_{f_t}) - t + 1$ and $m_z = (m_{f_1} + \dots + m_{f_t}) - t + 1$. In these formulas t denotes a number of convolutional layers and $n_{f_w} \times m_{f_w}$ is w -th layer filter size for $w = 1 \dots t$.

3.2 Detector Training and Usage

The proposed CNN architecture is a superposition of: zero-padding (of a size which will keep the feature map size constant) [20], convolutional layers and element-wise activation functions. Such a network can be trained to associate inputs $A_1 \dots A_p$ with the resulting maps representing object localization binary masks. Naturally, to obtain satisfactory training results, the neighboring pixels that represent the context of the analysed regions must be also taken into account.

Thanks to the translation invariance of CNN, if the object location on the image changes and context remains sufficient, the proposed solution guarantees that output will be translated as well. It makes application of CNN easier, than it would be for a naive solution which would require techniques such as sliding window.

The advantage of the described approach goes even further than that. Consider image $B_1 \dots B_p$, similar to $A_1 \dots A_p$ but of different size. For example $B_1 \dots B_p$ could be a bigger image including some objects to be detected. If it is used as an input of it can be remarked that still:

- padding and convolution layer keep the image size unchanged, since no parameters depend on input size;
- convolution is possible to calculate as long as feature maps are larger than filters (which is automatically satisfied if B_j are larger than A_j);
- element-wise functions are independent of the map sizes.

Consequently, the output map would still show the proper mask of a detected object [17]. In other words, without any additional utilities – after training on the small samples (which is remarkably faster than processing a big image with a small object) we get an object detector with support of any greater input size, as it is shown in Fig. 2. Detecting multiple objects works out of the box as well. If there is some space between the objects to detect, so the visual fields do not intersect, the process becomes equivalent to the detection of a single object.

Using some context around the object in the training images already prevents CNN from picking any points of the included background, but it leaves the network unprepared for any phenomena that occur only in greater distance

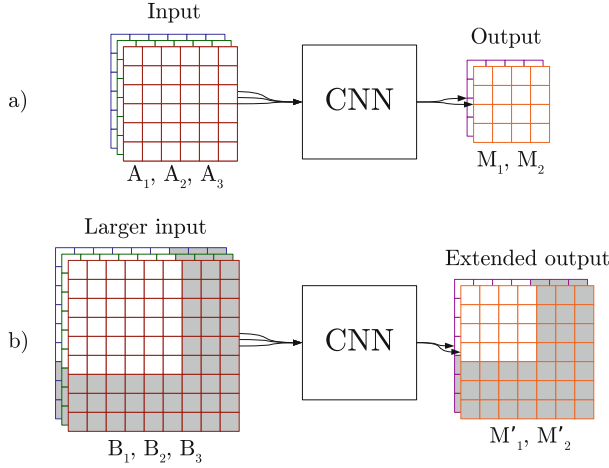


Fig. 2. For each additional row/column of input matrices, you get one more row/column of the output: (a) - the original configuration, (b) - the extended input. In (b) for a rectangle of the same size as A_j matrices results are similar to (a) configuration (it would work in this way for the white pixels of B_j and M'_i). Image originally published in [18].

from the detected objects. In order to avoid such problems with input regions appearing in the bigger image but not present in smaller training samples, the training set should include negative samples as it is described in Sect. 4.1.

3.3 Evaluation

As mentioned above, the size of the feature maps is kept constant from layer to layer in the proposed neural network. We also do not use MLP layers at the output and the goal of the training is regression instead of, as it would be typical for CNNs, image classification. The raw MR scans are put on the CNN input, and we expect the output to take a form of the same-sized image, clearly marking the MS lesions as white regions, surrounded by black, neutral background. In practice, however, the output image will not be truly black-and-white, and the intensity of a given output pixel may be interpreted in terms of the probability that it is a part of a lesion. Therefore, we have to apply thresholding in order to make the final decision and to obtain a black-and-white result that may be directly compared to the expert-generated ground-truth mask.

The value of the threshold $T \in [0, 1]$, used for this purpose, determines the standard evaluation measures of a binary classifier: *precision* and *recall*. Low threshold means that many brighter regions of the output image will be interpreted as sufficiently bright to represent demyelinating lesions, thus increasing the recall. For the lowest possible threshold value, $T = 0$, the *whole* image will be regarded as a brain tissue lesion, and hence every actual ground-truth lesion will be marked as properly detected (100% recall). The precision of this detection,

however, will be very low. On the other hand, using high value of T will result in the opposite: only the most outstanding regions will be detected as lesions, yielding high precision, but many actual lesions will not have sufficient intensity and they will remain undetected, yielding low recall. Therefore, the frequently applied “balanced” measure of classification efficacy is to compute the harmonic mean of precision and recall, known as the *F-measure*.

In our approach the F-measure is used to find the optimal value of the threshold T . After the training is finished, we threshold the output images (obtained for the input images from the training set) with several values of T , recording the resulting F-measure values. The value of T maximizing the F-measure then becomes the final threshold, which we subsequently use to compute the classification results on a separate set of images (the testing set).

However, it should be noted that the exact method of computation of precision and recall may be defined in various ways. Below we will present 3 approaches that were used in the present study to obtain different evaluations of the classification results.

Per-Pixel Evaluation (PPE). In the first evaluation method we measure the coincidence between the regions detected by the network and the ground-truth annotations by means of simple raw pixel count. For this purpose, we define three sets of pixels – the *true positive* pixels (TP), the *false positive* pixels (FP) and the *false negative* pixels (FN). The pixel at coordinates (x, y) belongs to one of these sets under one of the following conditions:

$$\begin{aligned} \text{TP} &: (I_{thres}(x, y) = 1) \wedge (I_{targ}(x, y) = 1), \\ \text{FP} &: (I_{thres}(x, y) = 1) \wedge (I_{targ}(x, y) = 0), \\ \text{FN} &: (I_{thres}(x, y) = 0) \wedge (I_{targ}(x, y) = 1), \end{aligned}$$

where I_{thres} and I_{targ} denote the image obtained at the network output (subjected to thresholding) and the target ground-truth image provided by the human expert, respectively. Having computed the number of pixels in each set, the precision and the recall are defined as:

$$\begin{aligned} \text{precision} &= \frac{|\text{TP}|}{|\text{TP}| + |\text{FP}|} = \frac{|\text{TP}|}{|\text{P}|}, \\ \text{recall} &= \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}|} = \frac{|\text{TP}|}{|\text{T}|}, \end{aligned}$$

where $|X|$ denotes the cardinality of the set X . The precision is hence defined as the proportion of the number of TP pixels (correctly reported within the lesion areas) to all of the *actually detected* pixels (*positive* pixels, P). Similarly, the recall is the proportion of TP to all the pixels that *should be reported* (*true* pixels, T) [18].

Connected Component Evaluation (CCE). The pixel-based approach described above is straightforward and unambiguous. However, it tends to underestimate the results, even when all the lesions have been properly found, in the case of significant mismatch of their shape or size. Counting pixels seems a bit simplistic here. It is also counterintuitive – we typically want to give more priority to finding the lesions than to marking their exact shape, especially when we consider the limited precision of the manually generated annotations. The conducted experiments revealed that the lesions marked in the output images were often much smaller than expected, due to relatively high values of the threshold T . Naturally, lowering the threshold would make them bigger – and closer in size to the corresponding annotations – but on the cost of generating many false-positive lesions, which would eventually decrease the precision significantly.

Therefore, in order to concentrate more on the number of detected lesions, instead of on the number of pixels, we decided to construct a different evaluation measure on the basis of the connected components (CC) representing regions identified in the thresholded output image and in the target image. Similarly to the pixel-based approach, we define several sets (of connected components in this case) and we compute the proportions of their respective sizes. Four sets are necessary here: the set of all *true* CCs in the ground-truth image (T), the set of all *positive* CCs in the thresholded output image (P), the set of all “matched” true CCs (MT) and the set of all “matched” positive CCs (MP), where the latter two are defined as:

$$\begin{aligned} \text{MT} &= \{cc_0 \in T : \exists(cc_1 \in P) cc_0 \cap cc_1 \neq \emptyset\}, \\ \text{MP} &= \{cc_0 \in P : \exists(cc_1 \in T) cc_0 \cap cc_1 \neq \emptyset\}. \end{aligned}$$

In other words, the output region is matched if it contains at least one pixel coincident to a lesion region in the ground truth image and similarly for the matched regions in the target image. It should be noted that we need the distinction between MP and MT, because several different CCs in the target image may be matched by a single connected component in the thresholded output image and vice versa. The precision and recall are then defined as:

$$\begin{aligned} \text{precision} &= \frac{|\text{MP}|}{|\text{P}|}, \\ \text{recall} &= \frac{|\text{MT}|}{|\text{T}|}. \end{aligned}$$

Region-of-Interest Evaluation (RIE). The CCE approach, as defined above, operates on a higher level of image representation (connected components instead of the pixels). Matching the lesions irrespective of their size and shape seems appealing, but it also has some drawbacks, unfortunately. The problem is that even the smallest regions, including single isolated pixels appearing in the thresholded output image are now considered separate CCs, having equal importance to bigger “visually relevant” regions. This often leads to a significant, yet quite

“artificial” increase of the number of the positive regions (P) followed by the drop of the precision value.

In order to overcome this and to make our evaluation more intuitive, we decided to introduce the third evaluation, based on post-processing of the thresholded output images. We aim at defining regions of interest (ROI) within them, so that a single ROI may cover several nearby connected components. This is done in several steps. First, we draw a bounding rectangle around every connected component found in the thresholded output image. Every bounding box is then padded (enlarged) by 10 pixels from all the four sides and this enlarged rectangle is filled with foreground (white) pixels. After that, it is possible, that individual nearby CCs got merged, so we repeat the search of connected components obtaining the final set of detected regions. On this “second-level” representation we compute the standard evaluation measures (precision, recall and F-measure) in the same way as in the CCE approach.

Additionally – for visualization purposes – we draw the bounding rectangle around every of those enlarged and merged “second-level” regions. In this way we obtain a very practical and useful outcome, that may be directly used by a specialist to immediately spot the regions of interest, potentially containing the demyelinating lesions in the MRI scan (Fig. 3).

4 Experiments

4.1 Dataset Preparation

The initial data set consisted of 100 scans from different patients. In order to guarantee the consistent image format, with fixed image resolutions and number of scan levels for each patient, data from 4 scans was discarded. The processed data was split into the set used for training and validation purposes (77 patients) and the test set (19 patients). This means that the evaluation on the test set

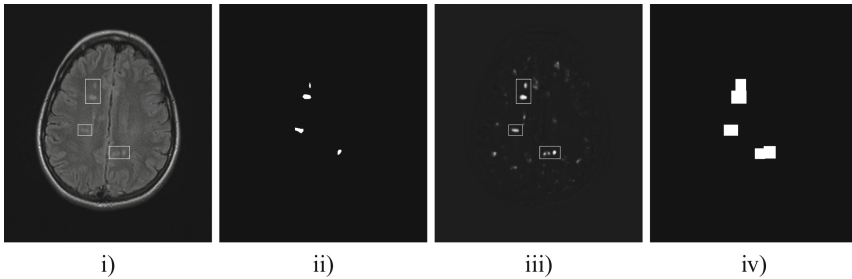


Fig. 3. Illustration of Region-of-Interest Evaluation. From left to right: an input image with the detected regions marked by the bounding rectangles (i), the target ground-truth image (ii), the raw output of the network (before thresholding) with the detected regions marked by the bounding rectangles (iii) and the filled rectangles of the individual connected components (iv).

is based only on the patients that were not known during the stages of weights adaptation and model selection. MR scans taken from the patients were converted to sets digital images, 448×512 pixels each. The scans that contained demyelinating plaques were used in the further processing. This yielded 982 training-and-validation images and 242 test images.

The 982 images selected for training and validation purposes were further processed in order to provide the training set where a significant part of surface consists of the plaques of demyelination. Instead of the whole images, selected 50×50 pixel tiles were used. The objective of this step was to reduce the computational complexity of training when compared to the full-resolution images, since the tile surface is 90 times smaller than the surface of the whole image. What is more, selecting tiles where the demyelinating plaques were overrepresented was intended to prevent the stochastic gradient-based training from reaching the local minima of parameters that yielded “all-zero” results, erroneously indicating that every analyzed scan is completely free of MS symptoms. The initial approach was to use only tiles with centered occurrences of demyelinating plaques. The initial attempts to create the working solution revealed that this method of building the training set does not cover all the phenomena visible in the MRI scans. In the result, bright objects that were underrepresented in the training set, such as skull bones, adipose body of the orbit and paranasal sinuses resulted in falsely-positive labeling of the MS lesions. In order to provide the model that recognizes such cases, additional tiles from the other regions of the scans were included in the training set as well (Figs. 4 and 5).

The selected data sets can be summarized in the following way:

- **Training set** – 7856 tiles of 50×50 size picked from the 982 training-and-validation images. Tiles were selected in a pseudo-random way, but areas with high average brightness or contrast were preferred. Approximately one tile out of three contained only healthy tissues, without demyelinating plaques. In case of MS lesions that were positioned close to the image boundaries, the image was extended appropriately. This data set is used for the weights adaptation in the presented neural networks.
- **“Quasi-validation set”** – 982 full-sized (448×512) images. This set serves similar purpose to the typical validation set, but due to limited amount of labeled data, it is not separate from the training set. It must be emphasized, however, that this set contains remarkably more data than the training set. The quasi-validation set is used for monitoring the learning progress and selection of additional parameters of the final solution, such as threshold level.
- **Test set** – 242 full-sized (448×512) images, separate from all the other data sets not only in terms of images extracted from MRI scans, but in terms of the set of patients that were examined. This set is used only for the final benchmarks of the selected models.

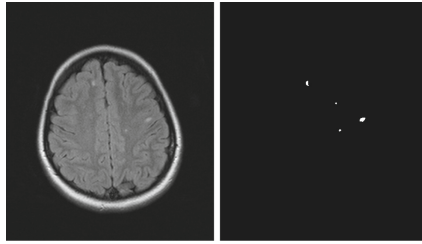


Fig. 4. Example of the MRI scan used in the test set (left) and the corresponding reference demyelinating plaques mask (right).

4.2 CNN Architecture

The structure of the network, i.e. the number of layers, the number of neurons, the size of the receptive fields and the non-linearity types as well as different training procedures were the subject of intensive experiments in the presented study. Three selected solutions are described below.

All the experiments were done with Caffe deep learning framework on a cluster node with Tesla K80M GPU accelerator. The training set of 7856 50×50 tiles was fed to the network in mini-batches of 199 tiles each. The proposed solution is a CNN composed of convolutional layers only (no MLP layers), which makes it behave like an image *filter*, which accepts input images of any size, without any changes to the architecture or the weights. This mechanism was explained in detail in Sect. 3.2. This property makes it possible to calculate mean square error (Euclidean loss) between the network outputs and the ground-truth masks achieved for the full-sized scans (“quasi-validation set”). This error value was used as the indicator of the training progress. The “quasi-validation set” set contained 982 images of 448×512 size from which training tiles were cut.

Basic Architecture. In order to provide a full description of the neural network architecture and the training process, a vast number of parameters needs to be decided manually. The series of trial-and-error attempts lead us to some general remarks about the optimal values of certain parameters. Six convolutional layers make the neural network deep enough to recognize complex objects and allow the back-propagation training to adapt all the weights in the network. Greater amount of consequent layers would make it difficult to train the filters of the initial layers (closer to the data input) because of the vanishing gradient. Standard momentum rate of 0.9 and the learning rate of 0.00001 seem to provide stable and effective training for the selected architecture. Images were processed by the neural network in batches, each containing 199 images to be processed simultaneously.

The specific architecture of the “basic” experiment is illustrated in the diagram 6. The standard approach involved using PReLU (parametric rectified linear units) activation functions between the layers and the unipolar sigmoid

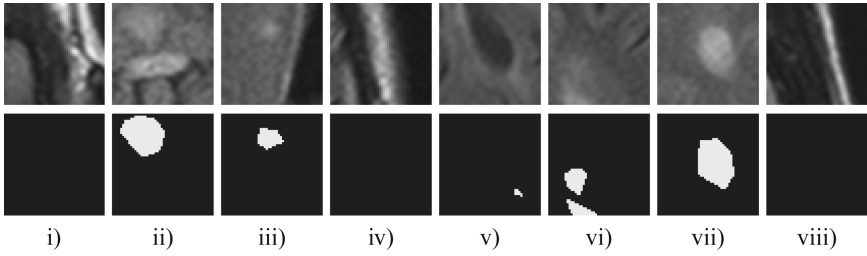


Fig. 5. Example of tiles cut from the training-and validation set (top) with the corresponding lesion masks (bottom). Both tiles and the masks were cut from the full-resolution images of the same format as it was illustrated in Fig. 4. Note, that tiles (i), (iv), (viii) do not contain the lesions. Tiles (i) and (iv), however, present some of the great number of possible big, bright structures that are likely to cause false positives when detecting the lesions.

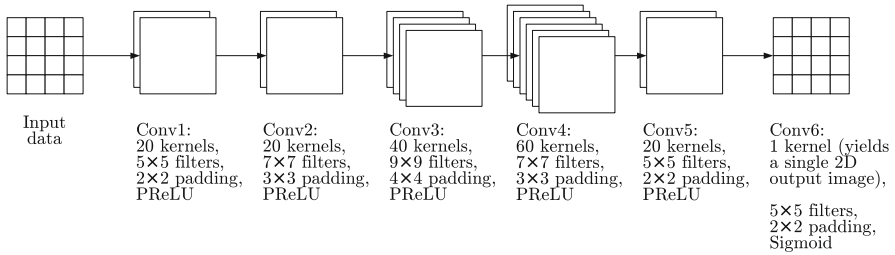


Fig. 6. The sequence of layers used in the convolutional neural network used in the basic architecture.

activation function on the output of the final convolutional layer. The final layer yields a single matrix, which can be later compared to the expected mask with marked demyelinating plaques.

The slow, but stable convergence in terms of mean squared error on quasi-validation set is presented in the top plot of Fig. 7. For as long as 24 millions of image propagations in the training process, the error on the quasi-validation set clearly decreases. The network learns to detect lesions on the training tiles, and the result is general enough to apply to the full-sized images. The minimum of mean squared error is **199.0**.

In order to provide a practical verification of the network effectiveness, the network output was thresholded to obtain the binary image, which can be compared directly to the target mask. The value of the threshold selected in order to maximize the F-measure, as described in Sect. 3.3. It should be noted, however, that the characteristics of the training set, which was composed of small tiles, were so different from the testing set containing the full scans. The appropriate way to select the most useful threshold was to maximize the F-measure on the quasi-validation set.

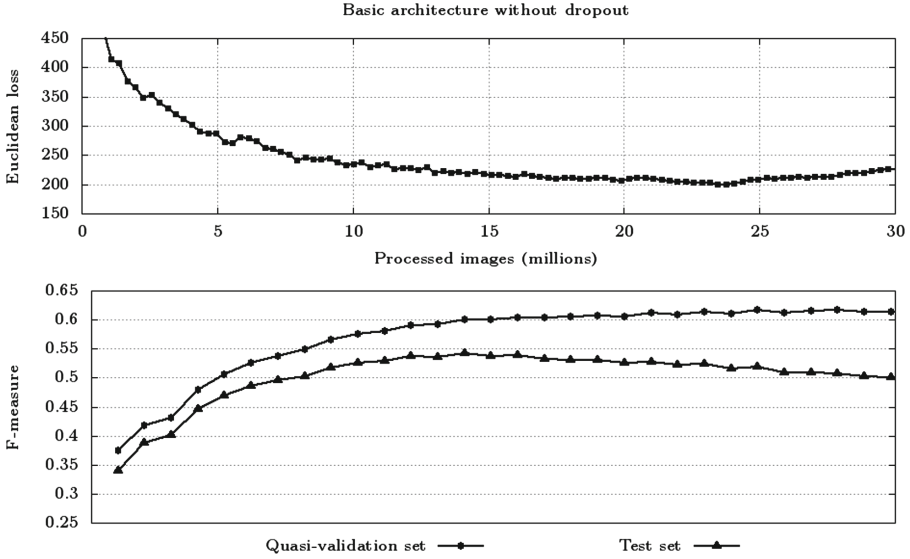


Fig. 7. Training of the basic architecture without dropout. Top: learning curve (Euclidean loss) on quasi-validation set; bottom: F-measure on the quasi-validation set and the testing set. The unit on the horizontal axis corresponds to 10^6 tiles, which were grouped in batches of 199 images.

The maximum F-measure value on the test set is 0.551, but we have no formal way of selecting that exact model. The best F-measure on the quasi-validation set is 0.622, but the corresponding model is visibly overtrained and yields only 0.496 on the test set. Following the lowest mean squared error would result in selecting a model that yields F-measure values of **0.617** on the quasi-validation set and **0.545** on the test set.

The per-pixel F-measure value observed on the quasi-validation set keeps growing as well, as we can see in the bottom plot of Fig. 7. The second curve presented in that plot, however, describes the dynamics of F-measure on the test set. This result starts decreasing much earlier than the MSE from the top plot – the generalization error becomes visible after processing 14 millions images. The network is apparently getting overtrained, as the result keeps losing its general properties. It must be emphasized, however, that this effect happens only after the whole training set was iterated over for almost 1800 times, which corresponds to ca. 26 h of training.

Basic Architecture with Dropout. The proposed extension to the architecture from the previous section involves a basic application of the dropout mechanism [21]. As it is presented in Fig. 8, the additional layer with $p = 50\%$ dropout probability was added directly before the final convolutional layer. In order to compensate for the reduced amount of data in the training phase

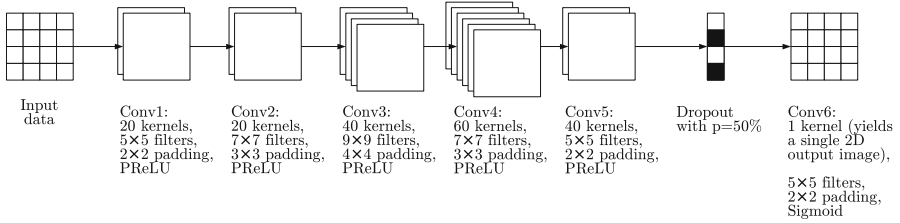


Fig. 8. The sequence of layers used in the convolutional neural network used in the basic architecture with dropout mechanism.

(half of the input values for Conv6 layer were replaced with zeros), the number of filters in Conv5 layer was increased twofold. The change involved only one level of the network, so the resulting training speed decrease amounted for only 15% when compared to the basic architecture.

As we can see in the plots from Fig. 9, the effects of network overtraining in the architecture with dropout are remarkably less intense than in the basic architecture. The minimum of minimal square error on the quasi-validation set occurred after ca. 24 millions of image propagations, which is similar to the previous experiment. The rate of error increase in the overtraining stage, however, is much lower than it was without dropout. Similar remark can be observed in the bottom plots of Figs. 7 and 9. The F-measure on the quasi-validation has similar dynamics in both cases. The F-measure on the test set, however, reaches the minimum notably later in case of the model with dropout (after 20 million images instead of 15 million), and does not start to drop as rapidly as it did in the previous experiment. The minimum of mean squared error is **195.7**, which is slightly lower than it was without dropout.

In order to compare the F-measure values to the previous model, we use the model that minimizes the mean square error again. This model, when used with the optimal threshold, generates F-measure values of **0.620** on the quasi-validation set and **0.539** on the test set, which is comparable to the previous experiment.

Improved Architecture. After the series of experiments on Tesla K80M GPU accelerator, the CNN architecture presented in Fig. 10 was proposed. The specific design of this architecture is supposed to take advantage from the fact that larger filters are easier to adapt when they are closer to the network output, because of the vanishing gradient making it difficult to adapt the convolutional layers close to the network input. Similar remark was a reason for increasing the number of filters in the first convolutional layer – since the filters are small and difficult to adapt, using the increased number of randomly-initialized filters is intuitively desirable. What is more, the dropout mechanism was used even more extensively than in the “basic architecture with dropout” – there were two levels of layers where some of the data (30% and 50%, respectively) was dropped out. The architecture with multiple dropouts means increased amount of necessary time

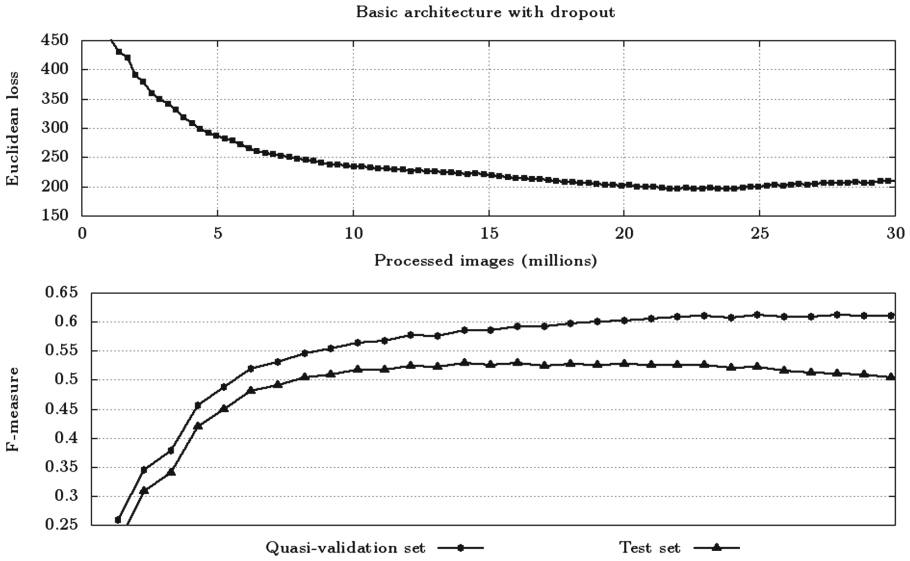


Fig. 9. Training of the basic architecture with dropout. Top: learning curve (Euclidean loss) on quasi-validation set; bottom: F-measure on the quasi-validation set and the testing set. The unit on the horizontal axis corresponds to 10^6 tiles, which were grouped in batches of 199 images.

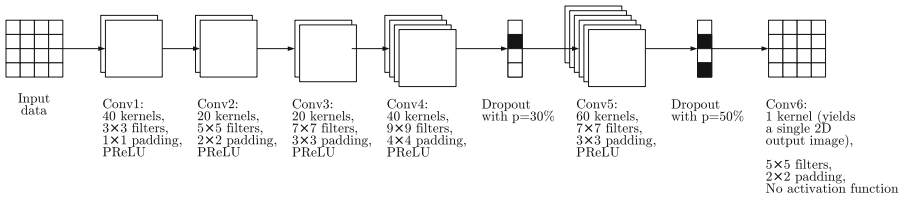


Fig. 10. The sequence of layers used in the convolutional neural network used in the new, improved architecture.

per processed image, which makes the training of this model almost 35% slower than the basic architecture.

The plots presented in Fig. 11, indicate that the general properties are similar to the basic architecture with dropout – the overtraining effects are not nearly as intense as in the basic architecture without dropout, but occur nonetheless. The number of processed images related to the mean square loss and F-measure optima is similar to the basic architecture with dropout as well. The achieved minimum of the mean squared error function, however, is the best amongst the three models, assuming value of **189.5**.

In order to compare the F-measure values to the previous model, we use the model that minimizes the mean square error again. This model, when used with the optimal threshold, generates F-measure values of **0.620** on the

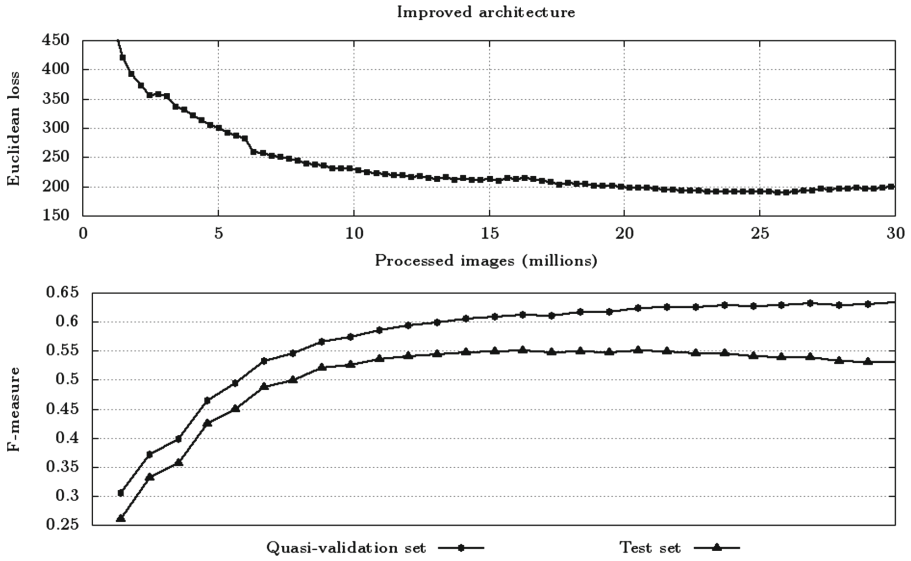


Fig. 11. Training of the improved architecture. Top: learning curve (Euclidean loss) on quasi-validation set; bottom: F-measure on the quasi-validation set and the testing set. The unit on the horizontal axis corresponds to 10^6 tiles, which were grouped in batches of 199 images.

quasi-validation set and **0.542** on the test set. This is apparently slightly better than the basic architecture with dropout, but not necessarily than the basic architecture. The three proposed solutions, despite of the differences, yielded vastly similar F-measure values.

4.3 Threshold Selection

Obtaining the final classification results required thresholding of the raw network output images, as described in Sect. 3.3. The threshold selection was based on the results obtained on the quasi-validation set, which in turn depended both on the network model (basic, basic with dropout, improved) and on the evaluation measure (PPE, CCE, RIE). This generated 9 possible experiment settings, and the threshold was computed for each of them individually. The obtained thresholds were quite similar, although some differences were evident between the pixel-based evaluation scheme and CC-based evaluation scheme (including also RIE). The representative plots are presented in Figs. 12 and 13.

As may be observed, the obtained threshold was higher for the measure based on the connected-components. This may easily be explained, if we consider that in the case of the CCE even a single pixel is enough to have the corresponding ground-truth lesion “matched“. We may therefore increase the threshold, removing more pixels (which in the case of the PPE approach would be punished), reducing also the number of false positive regions and boosting the precision in this way.

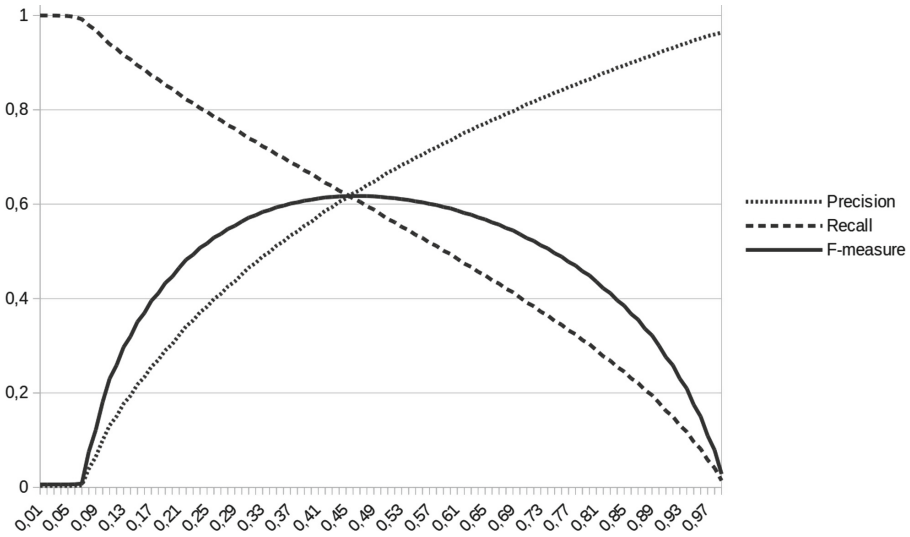


Fig. 12. Threshold selection for the PPE measure.

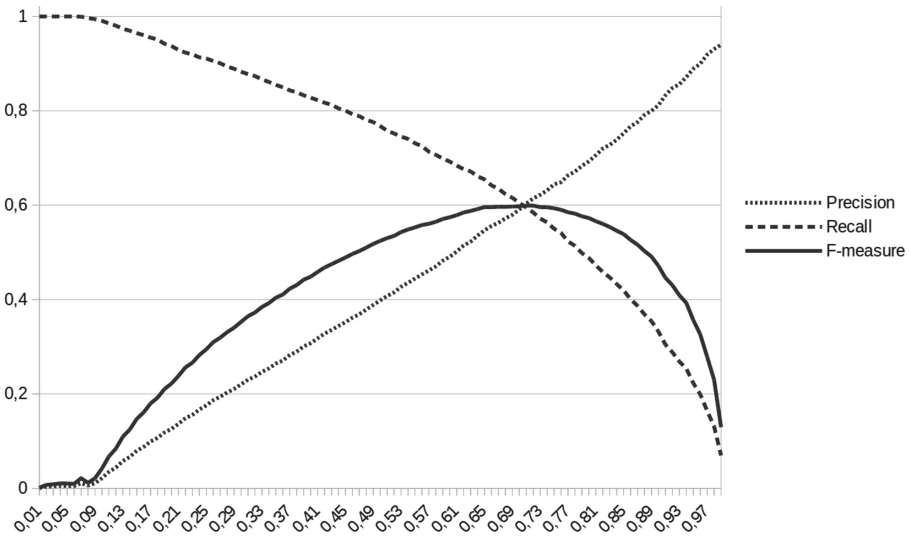


Fig. 13. Threshold selection for the CCE measure.

4.4 Sample Results

The results reported hereby have been obtained on the test set, with the three presented network models (basic, basic with dropout, improved), and the three evaluation methods (PPE, CCE and RIE). The threshold value was computed individually for each of the nine experiment settings on the quasi-validation set,

as described above. Table 1 presents the precision, recall and F-measure values for all the three network architectures and the PPE evaluation scheme. Similarly, Tables 2 and 3 show the results for the connected-component-based approaches. It should be noted the reported numbers of pixels and regions are counted across the whole test dataset (292 images).

Table 1. Experiment 1 (PPE) – the results.

Model	Threshold	T	P	TP	Precision (TP/P)	Recall (TP/T)	F-measure
Basic	0.46	143207	150444	80045	0.5321	0.5589	0.5452
Basic + Dropout	0.54	143207	148367	78595	0.5297	0.5488	0.5391
Improved	0.51	143207	156686	81362	0.5193	0.5681	0.5426

The results are generally quite similar, in terms of the obtained F-measure values. Interestingly, the evaluation scheme involving the connected components introduces only very small improvement over the pixel-based approach, although it is based on completely different assumptions. Only the region-based method is quite significantly better, exceeding 60%, probably due to the additional enlargement and merging of the connected components.

Table 2. Experiment 2 (CCE) – the results.

Model	Threshold	T	P	MT	MP	Precision (MP/P)	Recall (MT/T)	F-measure
Basic	0.71	1022	1319	605	711	0.5390	0.5920	0.5643
Basic + Dropout	0.77	1022	1341	620	704	0.5250	0.6067	0.5629
Improved	0.74	1022	1416	615	709	0.5008	0.6018	0.5466

Table 3. Experiment 3 (RIE) – the results.

Model	Threshold	T	P	MT	MP	Precision (MP/P)	Recall (MT/T)	F-measure
Basic	0.71	1022	912	675	514	0.5636	0.6605	0.6082
Basic + Dropout	0.78	1022	910	664	501	0.5505	0.6497	0.5960
Improved	0.75	1022	976	666	504	0.5164	0.6517	0.5762

Example images where the quality of lesion detection was remarkably good (Table 5), remarkably poor (Table 7) and acceptably successful (Table 4) were

presented in the tables that consist of: the input image, the demyelinating plaques mask (ground-truth annotations), and the results of all the three neural networks presented in this paper. Additionally, the successful image, which can be considered easy in terms of lesion labeling, is used to demonstrate that the presented methods need to be intelligent even in order to solve the simplest tasks. Simple thresholding can be considered as an alternate method of marking “bright, important points” in the image. However, when compared to the neural network, such a simplistic attitude is likely to act remarkably poor, as it is shown in Table 6.

Table 4. Example image: acceptable detection of multiple small demyelination plaques.

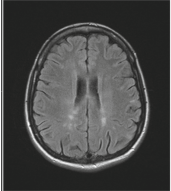

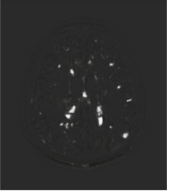
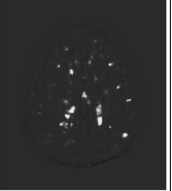
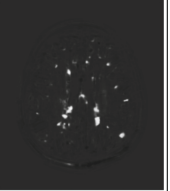



Input	Target	Basic	Basic+Drop	Improved
				
				

Table 5. Example image: remarkably successful detection in case of all models.

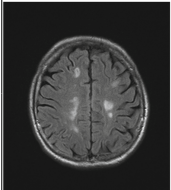

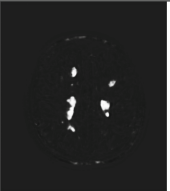
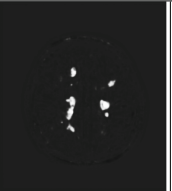
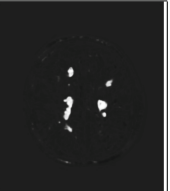



Input	Target	Basic	Basic+Drop	Improved
				
				

Table 6. Successful detection compared to simple thresholding.



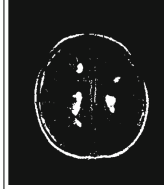
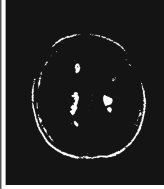
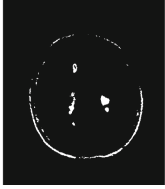
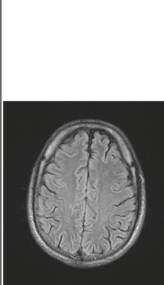

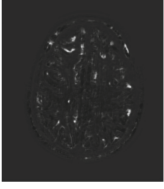
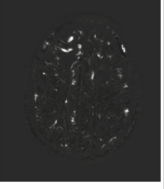
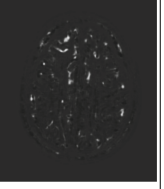



Target	Improved	Threshold 50%	Threshold 55%	Threshold 60%
				

Table 7. Example image: selected difficult case.

Input	Target	Basic	Basic+Drop	Improved
				
				

5 Analysis

The sample results shown in the previous section are intended to demonstrate the general possibilities of CNNs application for the image detection task. The detection of typical demyelinating lesions was successful often enough to consider the achieved results promising. It must be emphasized that the ability to detect the demyelinating plaques is not only based on their intensity, but on their shape and the characteristics of the surrounding tissue as well. This property can be illustrated by the comparison of the results from Tables 5 and 6, where the same sample is processed with the proposed neural network and with a simple threshold operator.

It can be observed that the threshold of 50% is too low to detect only the lesions, as it marks some other points between the cerebral hemispheres as well. Higher threshold values, however, result in heavily reducing (55%) or totally omitting (60%) one of the lesions. At the same time, even the threshold level of 60% is not sufficient to ignore the skull. Convolutional neural networks, however, have no problem with labeling all of the demyelinating lesions contained in this sample and with ignoring the skull. This specific sample is, apparently, easy enough to process by all the proposed models.

The general result of F-measure remaining around the range of 55%–60% even in the most optimal of the presented solutions is sufficient to be considered useful, yet far from the perfect outcome. The reasons for this can be investigated in more detail in order to formulate possible ways to improve this result.

One of the problems is related to the structures such as bones and meninges being the sources of false positives. As it was described in Sect. 4.1, the problem can be partially addressed by extending the training set with additional samples where no demyelinating plaques are present, and the average brightness and contrast are relatively high. As it was shown in Tables 5 and 4, omitting the skull is a simple problem. Structures such as eyes, which are present only in some of the scans, are more likely to cause problems. Scans close to the top of the skull are difficult to process as well, as the bones of the calvaria are not perpendicular to the projection plane. This special case makes the bones appear too thick to be properly recognized as unrelated to the lesions – example of this phenomenon is visible in Fig. 5.

One of the possible solutions is to increase the number of training samples containing such structures, in order to make the network train directly to solve the task of recognizing them. Another remedy worth considering would be to increase the size of tiles. The 50×50 images that are included in the training set are insufficient to contain some of the brain tissues with enough context to train the CNN to ignore the bright regions of great surface, such as eyes. Both suggested solutions lead to increase of the overall area of training data unrelated to the lesions. This property is, however, undesirable – it makes the network more likely to learn to generate plain black output images. The “all-zero” outputs are usually related to the local minimum of the cost function used in the learning process. The greater the percentage of black points in the reference output is, the more difficult it is to force the network to do anything else. More promising direction is therefore to use some external tools to remove the irrelevant parts of the input MR scans. Some automatic or semi-automatic solution can be applied prior to the training and testing with CNN-based solutions. Such an approach would make the whole solution less universal, but it would let the CNN concentrate on the cerebral tissue only – and that is where we are expected to find MS lesions.

Another problem that has proven to have negative influence on the obtained results is the quality and quantity of the collected scans. The training set is small, and in order to use as much data as possible, the quasi-validation set was not separate from the training set. Greater number of patients in the database would be likely to improve not only the result, but the overall possibility to use the most proper experimental methods as well. The remark on quality of the data set is related to the visibility of some of the lesions. While the most of the demyelinating plaques are clearly visible in the scans, there are many small or very faint lesions present in the scans as well. Lesions like that are likely to pose problems in the task of unequivocal identification as MS plaques. The problems with overtraining and generalization, indicated in Figs. 7, 8, 9, 10 and 11 suggests that it would be advantageous to use a significantly bigger set of the training images. More consistent standards of image annotation, perhaps

involving several independent specialists, would undoubtedly improve the quality of the data set as well.

Yet another issue worth mentioning is related to the precision of defining the actual shape of the lesions by human annotators. It must be emphasized, that in terms of per-pixel F-measure, even if all the lesions were properly detected in the output image, the size and shape provided by the neural network is likely to differ from the ground-truth mask. This issue was partially solved by the connected component-based approach presented in this paper. The suggested approach to splitting/joining of adjacent lesions is still dependent on the size of the plaques suggested by the human annotator, but it is a notable step towards the more credible quality measure of the suggested solutions.

6 Conclusions and Future Work

Diagnosis of the MS requires careful and time-consuming analysis of the brain MRI scans. The final interpretation and decision about the treatment always belongs to the human expert with appropriate medical knowledge. This process, however, can be assisted with an automatic tool which is prepared with the techniques of machine learning. The phenomenon of “demyelinating plaques” that are visible in the MRI scans is precisely defined and well known to the radiology specialists, but it is difficult to imagine any explicit, concise mathematical formula that describes a plaque. The presented work is intended to provide the best possible suggestions that can be obtained with the convolutional neural networks.

The data set used in this paper consisted of MRI scans of 100 patients. The groups was intended to be representative, so it included patients from different age groups. Multiple slices from each MRI scans were stored in digital images of relatively high resolution, which is 448×512 pixels. Images of that size were cut into 50×50 for the purpose of CNN training. Multiple neural network models proposed in this paper were trained from scratch, starting with randomly initialized filter contents. Due to the specific architecture which was based solely on convolutional layers, the solution was able to process images of different sizes, as it was described in Sect. 3.2. This means that the network trained with 50×50 tiles could be used for full-resolution 448×512 images without any changes to the architecture or the weights that were achieved through the network training.

One of the goals of this paper was to analyse alternative methods of evaluation of the classification results. In addition to the approach used in previous works [18], based on counting individual pixels, two additional methods have been tested in the present study. Analysis of the connected components brought about a very moderate increase of the reported results, while defining the regions of interest from the enlarged and merged connected components enabled to present the detected areas of demyelination in a potentially useful and visually appealing way.

The best results in terms of per-pixel F-measure were close to the 55% on the test set. While this result is not perfect, it can be considered as sufficient to get the general location of the most of the plaques, which is already useful

when the task of diagnosis assistance is considered. The expected masks used in both training and evaluation of the neural networks consisted of polygons, which were marked approximately by the medical specialists. Repeating the same polygon shapes is virtually impossible – either for the neural network or for another expert. Some of the significant sources of errors were related to the large bright areas resulting in the false positives. This includes temporal bones and optic nerves. Another common source of errors was related to the small regions of noise that were erroneously detected as demyelinating plaques. Additional consideration was devoted to the points that were detected in a general area of the MS lesions – the proposed modifications to the measure of the object detection quality provides us with some deeper insight into the results analysis.

The obtained result is promising, but the further room for improvement remains apparent. The crucial room for improvement is related to the data set size – greater number of training samples, covering better variety of cases, would be likely to improve the result. The selected size of the training samples, which is 50×50 , is another parameter that might require further discussion. Larger tile size would make it easier to include the whole temporal bones and optic nerves in the training samples. Greater tile size, however, makes the training additionally difficult, because generating plain black outputs becomes a remarkable local minimum of the neural network cost function. This problem can be addressed either by cost function modification or manual region growing on the expected outputs that would increase the number of white points. Alternatively, the irrelevant parts of the input images – namely, everything but the cerebral tissue – can be removed manually by the separate tool.

The general field of the CNNs application for the medical image processing is usually affected by the difficulty with collecting the sufficient data sets. Unsurprisingly, this problem is visibly present in our work as well. Our analysis, however, can be considered as an initial step towards even more efficient solutions. Object localization based solely on convolutional layers, dynamic threshold selection, and detailed description of the results involving F-measure and connected components are some ideas, that – when used together – form an elegant solution that can be applied to the great diversity of object localization problems.

Another way to improve the proposed method is to involve some well-known pretrained CNN models instead of starting from the random weights. Neural networks such as AlexNet [8] or VGG [22] consist of carefully trained weights that are known to be useful in detection and classification of multiple normal, real life objects. The mentioned networks are usually used as classifiers, but application to the scale-preserving object localization solution is possible as well. Using parts of the network with maximum-pooling layers is not necessarily impossible in this task – the problem of restoring original resolution can be addressed with techniques such as deconvolutional neural networks [12].

Acknowledgements. This project has been partly funded with support from National Science Centre, Republic of Poland, decision number DEC-2012/05/D/ST6/03091.

Authors would like to express their gratitude to the Department of Radiology of Barlicki University Hospital in Lodz for making head MRI sequences available.

References

1. Tomczyk, A., Spurek, P., Podgórski, M., Misztal, K., Tabor, J.: Detection of elongated structures with hierarchical active partitions and CEC-based image representation. In: Burduk, R., Jackowski, K., Kurzyński, M., Woźniak, M., Żołnierek, A. (eds.) Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015. AISC, vol. 403, pp. 159–168. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-26227-7_15
2. Tomczyk, A., Szczepaniak, P.S.: Adaptive potential active contours. *Pattern Anal. Appl.* **14**, 425–440 (2011)
3. de Brebisson, A., Montana, G.: Deep Neural Networks for Anatomical Brain Segmentation. ArXiv e-prints [arXiv:1502.02445](https://arxiv.org/abs/1502.02445) (2015)
4. Shelhamer, E., Long, J., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. ArXiv e-prints [arXiv:1605.06211](https://arxiv.org/abs/1605.06211) (2016)
5. Milletari, F., Navab, N., Ahmadi, S.A.: V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. ArXiv e-prints [arXiv:1606.04797](https://arxiv.org/abs/1606.04797) (2016)
6. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. ArXiv e-prints [arXiv:1505.04597](https://arxiv.org/abs/1505.04597) (2015)
7. Hubel, D.H., Wiesel, T.N.: Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *J. Neurophysiol.* **28**, 229–289 (1965)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012)
9. LeCun, Y., Bengio, Y.: Convolutional networks for images, speech, and time-series. In: Arbib, M.A. (ed.) *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge (1995)
10. Cireşan, D.C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J.: Flexible, high performance convolutional neural networks for image classification. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, IJCAI 2011, vol. 2, pp. 1237–1242. AAAI Press (2011)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR 2009 (2009)
12. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. CoRR abs/1311.2901 (2013)
13. Nguyen, T.V., Lu, C., Sepulveda, J., Yan, S.: Adaptive nonparametric image parsing. CoRR abs/1505.01560 (2015)
14. Cheng, G., Zhou, P., Han, J.: Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **54**, 7405–7415 (2016)
15. Mopuri, K.R., Babu, R.V.: Object level deep feature pooling for compact image representation. CoRR abs/1504.06591 (2015)
16. Matsugu, M., Mori, K., Mitari, Y., Kaneda, Y.: Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Netw.* **16**, 555–559 (2003)
17. Dai, J., He, K., Sun, J.: Convolutional feature masking for joint object and stuff segmentation. CoRR abs/1412.1283 (2014)

18. Stasiak, B., Tarasiuk, P., Michalska, I., Tomczyk, A., Szczepaniak, P.: Localization of demyelinating plaques in MRI using convolutional neural networks. In: Proceedings of the 10th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2017), BIOIMAGING, vol. 2, pp. 55–64. SCITEPRESS (2017)
19. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. CoRR abs/1502.01852 (2015)
20. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, pp. 2278–2324 (1998)
21. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)