Investigating efficient CNN architecture for multiple sclerosis lesion segmentation

Alexandre Fenneteau[1,2,3], Pascal Bourdon[2,3], David Helbert[2,3], Christine Fernandez-Maloigne[2,3], Christophe Habas[4,3], and Rémy Guillevin[3,5,6]

[1]Siemens Healthcare, Saint Denis, France
[2]XLIM Laboratory, University of Poitiers, UMR CNRS 7252; Poitiers, France
[3]I3M, Common Laboratory CNRS-Siemens, University and Hospital of Poitiers ; Poitiers, France
[4]Neuroimaging Department, Quinze Vingts Hospital; Paris, France
[5]Poitiers University Hospital, CHU; Poitiers, France
[6]DACTIM-MIS/LMA Laboratory University of Poitiers, UMR CNRS 7348; Poitiers, France

Abstract

Purpose: The automatic segmentation of multiple sclerosis lesions in magnetic resonance imaging has the potential to reduce radiologists' efforts on a daily time-consuming task and to bring more reproducibility. Almost all new segmentation techniques make use of convolutional neural networks, with their own different architecture. Architectural choices are rarely explained. We aimed at presenting the relevance of a U-net like architecture for our specific task and at building an efficient and simple model.
Approach: An experimental study was performed by observing the impact of applying different mutations and deletions to a simple U-net like architecture.
Results: The power of the U-net architecture is explained by the joint benefits of using an encoder-decoder architecture and by linking them with long skip connections. Augmenting the number of convolutional layers and decreasing the number of feature maps allowed us to build an exceptionally light and competitive architecture, the MPU-net, with only approximately 30,000 parameters.
Conclusion: The empirical study of the U-net has led to a better understanding of its architecture. It has guided the building of the MPU-net, a model far less parameterized than others (at least by a factor of seven). This neural network achieves a human level segmentation of multiple sclerosis lesions on FLAIR images only. It shows that this segmentation task does not necessitate overly complicated models to be achieved. This gives the opportunity to build more explainable models which can help such methods to be adopted in a clinical environment.

Keywords— MRI, CNN, segmentation, multiple sclerosis, architecture, efficient

1 Introduction

Multiple Sclerosis (MS) is an autoimmune and inflammatory disease affecting the central nervous system. It affects up to 0.2% of people depending on the geographical region. It was estimated to cause approximately 19,000 deaths and the disability of more than 1,000,000 people worldwide in 2016 [1, 2]. The disease induces demyelination and inflammatory lesions. Magnetic Resonnance (MR) exams are commonly used for the diagnosis and the monitoring of the disease as they allow radiologists to localize and characterize lesions, by using the McDonald criteria [3]. This screening is a repetitive, time-consuming, and prone to inter-observer variability [4]. Consequently, its automation has been encouraged via few segmentation challenges [5–7].

1.1 State of the art

In the last decade, several supervised and unsupervised attempts have been performed to automatically segment the MS lesions. The review of Danelakis *et al.* [8] shows that the recently published automatic lesion segmentation algorithms use clustering techniques, such as the Lesion Toolbox [9], classification techniques such as K-nearest neighbors [10] or Random Forests [11] and more recently artificial neural networks.

The use of neural networks is promising due to their potential to deliver better performances. To the best of our knowledge the first known attempt to use artificial neural networks in segmentation of MS lesions began in 1998 with the work of Goldberg *et al.* [12]. They used such networks for segmentation artifact removal. Later, Kuwazuru *et al.* [13] used a fully connected network for false positive removal. More recently, Convolutional Neural Networks (CNNs) have been increasingly used with success by researchers.

Among the top three best proposals of the Medical Image Computing & Computer Assisted Intervention (MICCAI) 2016 MS challenge [14], two were using CNNs: Valverde *et al.* [15] with only two convolutional layers followed by a dense one and McKinley *et al.* [16] with their encoder-decoder network, the Nabla net.

More recently, other approaches have been proposed using CNNs. Nair *et al.* [17] proposed a method for estimating prediction uncertainty. Brosch *et al.* [18] proposed to pretrain their CNN as convolutional restricted Boltzmann machines. Other work to pretrain neural networks has been conducted using self-supervision [19]. Another proposition for improving lesion segmentation consists of multi-class tissue segmentation at the same time as lesion segmentation [20]. Hashemi *et al.* [21] proposed an encoder-decoder CNN and studied the influence of β parameter of the F_β loss function. All these studies used very different CNNs for assessing their training methods and achieved state-of-the-art performances. However, we do not know the extent of the influence of architecture on predictions not why these authors made particular network design choices.

1.2 Proposal

Generally, the design part of the neural networks is not clearly stated in articles and it is difficult to evaluate how architectural choices are made. We present here our experimental approach for the design of a new neural network that has been created for being compact and competitive. This manuscript provides discoveries and explanations about architecture for segmenting MS lesions. As eXplainable Artificial Intelligence (XAI) techniques [22] are gaining importance in the medical imaging area with some successful attempts [23], and with the idea that a simpler model is easier to explain [24], we performed this work with the idea of attempting to define an efficient CNN in terms of number of learnable parameters.

With the success of using adaptations of U-net [25] in medical imaging in general such as in brain tumor segmentation [26], pancreas tissue segmentation [27] but also in MS lesion segmentation [17, 21, 28], this study focuses on U-net architecture.

The influence of design choices was analyzed throughout different experiments on a simple U-net like architecture. The analysis focused on the following points:

- Feature map down-sampling and up-sampling;
- Use or not of skip-connections;
- Complexity.

After experimenting, we took the observations on performances into account and designed our own U-net like architecture, the Minimally Parameterized U-net (MPU-net). It is designed with the constraint of having a very limited number of learnable parameters. We compared its performance to state-of-the-art methods by submitting the results to the Institute of Electrical and Electronics Engineers (IEEE) International Symposium on Biomedical Imaging (ISBI) 2015 MS segmentation [7] online leader-board[1].

Most of the time, the relevance of CNN architectures for MS lesion segmentation is rarely explained, except by a final performance score. The aim of the study is to experimentally assess architectural choices in a U-net, in order to build an efficient architecture.

2 Materials and Methods

2.1 Data

The work has been performed using the ISBI 2015 MS segmentation challenge [7] data set. It has been chosen since it is the newest publicly available data set containing a training and a testing set, which can be submitted and compared to other sets, by using its online leader-board. Only one data set was used among all publicly available MS segmentation data sets, thus avoiding working with images that were differently acquired and preprocessed.

The training set consists of 21 preprocessed MR exams from different time points of 5 patients with ground truth and the test set provides 61 MR exams from 14 patients without expert segmentations. Each consecutive time point is separated by almost a year. All exams are preprocessed and the segmentation of MS lesions was performed by two different experts.

The data sets provide T1, T2, T2-fluid-attenuated inversion recovery (FLAIR) and Proton Density (PD) images for each MR exam. In T1 images, MS lesions can be seen as hypo-intensities. In this set, T1 images are mainly used for anatomical purposes since no contrast enhancing agent is injected to patients. The T2 images are more valued and lesions appear as hyper-intensities. One problem with the T2 sequence is that some periventricular lesions can be difficult to detect. FLAIR images counter this effect by attenuating fluid signal in ventricles while keeping the advantages of the T2 sequence. This is the reason why, this sequence is preferred in practice. The PD images are sometimes used as an assistance, since in these images, lesions can be seen as hyper-intensities and can help to reveal subtentorial lesions.

Among all available MR sequences, we chose to select only the FLAIR images. The main reason for that was that we wished to keep the training as simple as possible since our study required to train many different architectures and that adding input would augment the complexity of the task. According to the McDonald criteria [3], the MS lesions can be seen in T2 images, and in practice, FLAIR images are more frequently used. Furthermore, according to *Feng et al.* [28], FLAIR imaging is the most effective. To the best of our knowledge, this sequence is used in every recent deep learning approach [18, 29, 30].

All the images are provided in $181 \times 217 \times 181$ voxels with voxel size of $1 \times 1 \times 1$ mm^3. These images were previously preprocessed for the challenge by Carass *et al.*, including the registration and resampling steps. The original size of FLAIR images before preprocessing were $256 \times 256 \times 35; 70$ voxels with voxel size of

[1] available at `https://smart-stats-tools.org/lesion-challenge`

$0.8281 \times 0.8281 \times 4.4; 2.2$ mm^3.

2.2 Additional preprocessing

Before using the images, few image preprocessing steps were performed in order to have compact images, with comparable and normalized intensities.

The images were all cropped in the same way to avoid unnecessary calculations. Then all images were histogram matched [31] using Insight Toolkit (ITK) [32,33], with the image with the maximum dynamic range, in order to make intensities comparable while not reducing the dynamic range. The final step was to normalize each image. The intensity mean and standard deviation were calculated only for brain voxels in training images. Normalization was performed by subtracting the mean and by dividing by the standard deviation. The background voxels were excluded from the calculation of statistics because they do not represent the data directly and would influence the normalization process.

2.3 CNNs training

Following the work on patch size by Snehashis *et al.* [34], we decided to train our neural network with multiple small patches of size $32 \times 32 \times 32$ voxels/mm^3. For any lesion size, a patch can capture the lesion, or a great part of it, and the surrounding brain texture (see Fig. 1). This makes it possible to segment very locally while occupying little memory space. This multiple small patch approach has also the benefit of augmenting the number of training examples. This is the only type of data augmentation that we have performed.

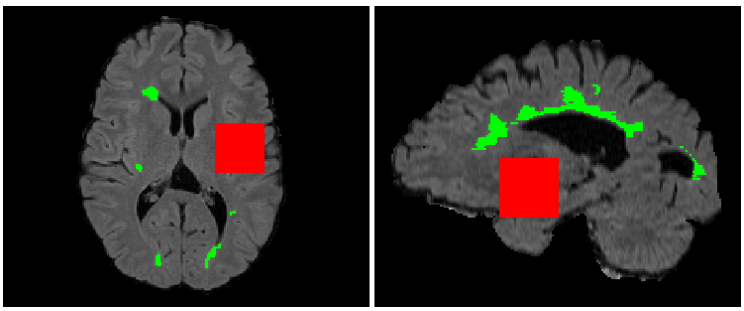


Figure 1: Patch size (in red) and lesion size (in green). Axial view at the left and sagittal view at right. Pictures are not taken from the same patients and are not in the same location

During training each epoch contains 100 batches. For each batch, the gradient is calculated and back-propagated. One batch consists of 5 patches per exam, extracted randomly in a specific area different for each batch. The CNN learns each time from all training patients in a specific brain area. Patch positions are constrained in order to have at least 80% of brain area. The training is performed so as to present many patches from all patients to the neural network. The patches cover all the brain in one epoch as uniformly as possible.

During testing, for each model, the prediction masks are the mean of patch predictions extracted in sliding window manner with strides $8 \times 8 \times 8$. These stride values were set empirically to have a sufficient overlapping prediction consensus with a reasonable number of individual patch predictions.

As stated in 2.1, there are two manual segmentation maps: one for each radiologist involved in annotation. The CNNs were trained with both maps at the same time, so as to evaluate their capacity to mimic each radiologist differently.

2.4 On designing reference

Guided by the success of the U-net architecture, we decided to take a very simple adaptation of this architecture as reference (see Fig. 2). We did not add any architecture regularizers, such as batch normalization [35] or dropout [36] even if they may have improved performances. For the same reason, architecture testing began with only one convolution at each level of the neural network. Transposed convolutions were replaced by up-samplings for simplifying and avoiding checkerboard artifacts [26]. All convolutions were followed by a Leaky ReLU function [37] as non-linear activation with alpha=0.3 and convolutional filters were initialized with the Glorot uniform technique [38].

For designing architectures, some arbitrary rules were followed to limit the complexity of architectures:

- The number of output feature maps of convolution has to be the same in the encoder and decoder at the same level, except for the output;
- The number of output feature maps of convolution has to be crescent with level depth;
- The number of feature maps rapidly augments for the first levels up to 80 and then slowly augments.

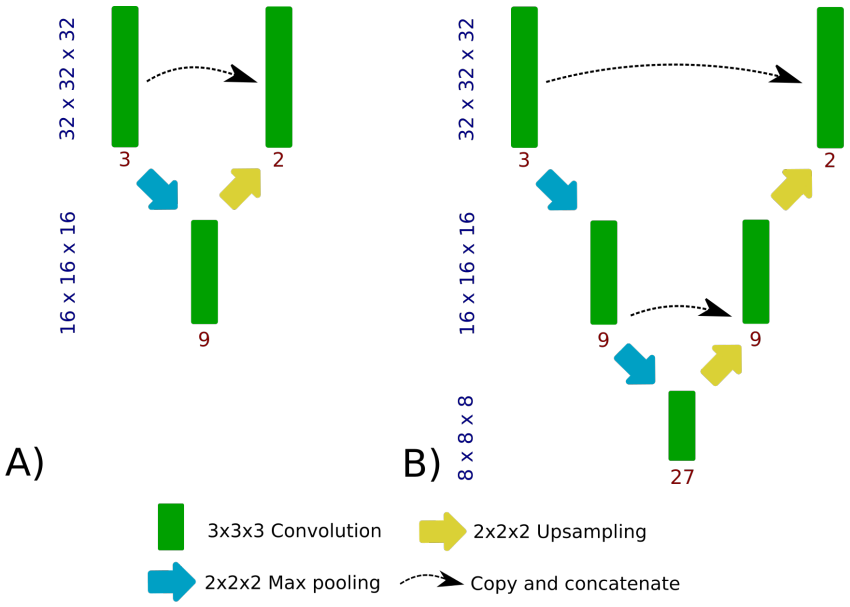


Figure 2: A) Two leveled and B) three leveled versions of the reference architecture. The size of output feature maps is written vertically in blue at each level and the number of feature maps is written in red below them.

Each experimented architecture possesses four different versions. Each version has a different number of "levels" in encoder and decoder and so has different depths. In this study a "level" designates the region with the same input size: see Fig. 2 for graphical explanation. This figure only shows the 2 and 3 leveled versions. The 4 and 5 leveled versions are not illustrated for clarity purposes but continue as in Fig. 2. The numbers of the different feature maps corresponding to each level can be found in Table 1.

Level 1	Level 2	Level 3	Level 4	Level 5
3	9	27	81	99

Table 1: Number of feature maps at each level.

The deep learning part is developed with Keras [39] and Tensorflow [40] backend; the models are all trained with the Dice score loss and the Adam optimizer [41]; the learning rate is set to 0.001 and the learning is stopped after 20 epochs without improvements over the test set.

3 Consequences of architectural changes

3.1 Training and testing subsets

The original testing set does not include the ground truth as its performances can only be assessed through submission, as described previously in 2.1. However, we needed to evaluate a lot of experiments for assessing architectural choices and did not want to flood the ISBI online leader-board. So, the original training set was split into a sub-training set of 17 exams from 4 different patients and a sub-test set of 4 exams from the remaining patient. All the models of the current section were trained and evaluated on these subsets.

3.2 Experiment setup

The experiments were performed to assess the influence of:

- Feature map size change throughout the U-net
- Skip connections
- Augmenting architectures in terms of:
  - Depth
  - Number of feature maps

To do so, the reference architecture in Fig. 2 and Fig. 3 was mutated. Again for greater clarity, the figures present only the architecture versions with two levels.

3.3 Metrics

The task of segmenting MS lesion voxels is a challenging task since only few voxels have to be detected in a large volume. To address this problem this study uses two interesting simple measures:

- $Sensitivity = \frac{TP}{P}$, where TP and P denote true positives and ground truth positives, also known as the recall or True Positive Rate (TPR). Here, this metric measures the aptitude to detect a maximum number of voxel lesions.
- $Precision = \frac{TP}{TP+FP}$, where FP denotes false positives, also known as the Positive Predictive Value (PPV). This metric is used to assess the capacity to detect correct voxels.

The evaluation of MS lesion segmentation is commonly computed by using the Dice score [18, 20, 28, 29]. This measure can be written as the harmonic mean of precision and sensitivity Equation 1. So, here the Dice score, measures equally the capacity of our predictors to detect a maximum number of voxel lesions and a

maximum of correct ones.

$$Dicescore = \frac{2TP}{2TP + FP + FN} \qquad Dicescore = 2 \cdot \frac{Precision \cdot Sensitivity}{Precision + Sensitivity} \tag{1}$$

As Dice score measure is relevant for assessing segmentation quality, all architectures use it as loss function.

3.4 Experiments

Each version of each type of architecture has been trained 5 different times. The plots show the mean performances of these trainings with the 99% confidence interval, which was computed using Bootstrap Methods [42]. The confidence interval is used here to represent the dispersion of performance for one specific version. It helps to evaluate the robustness and stability of the technique.

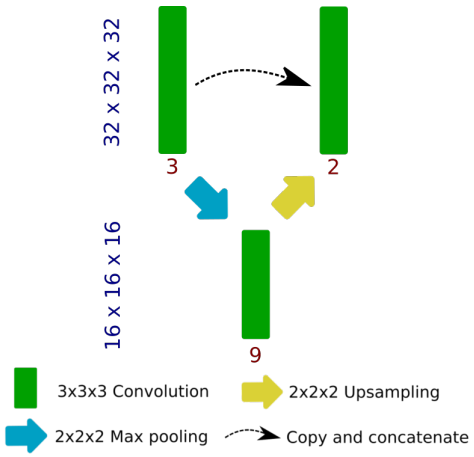


Figure 3: The two leveled version of the reference architecture used in experiments.

For each different architecture, the segmentation maps are compared with those of the reference architecture (in Fig. 3).

3.4.1 Change of feature map size

Due to our interest in size reduction/increase in the U-net, we performed three experiments:

- Size reduction of feature maps with max-pooling in the encoder path and size increase with simple up-sampling (our reference): Fig. 3.
- Size reduction of feature maps with $3 \times 3 \times 3$ convolution with stride of 2 in the encoder path and size increase with corresponding transposed convolution: Fig. 4.
- No feature map size change: Fig. 5

In the Fig. 6, the couple max-pooling / up-sampling seems to have the best performances for each architecture variation. For each version, it gives about 0.1 Dice score improvement, compared to architectures without size change. The feature map size change with convolution and transposed convolution is equivalent to the max-pooling / up-sampling only for the two last versions, with the cost of adding one convolutional layer.

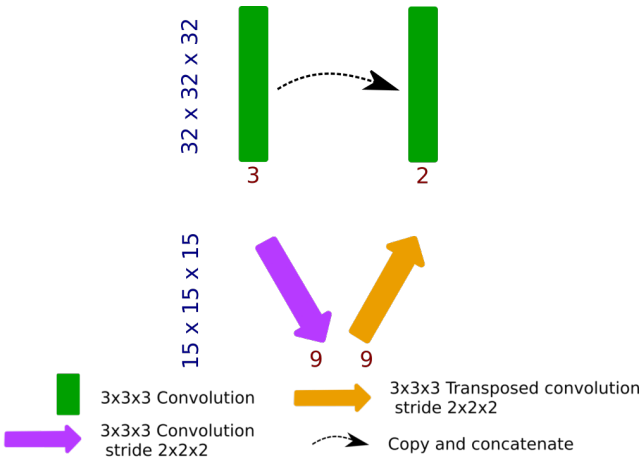


Figure 4: The two leveled version of the *two strided convolution* architecture.

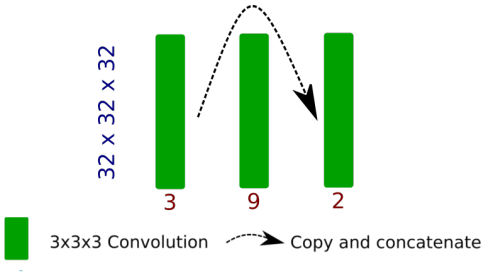


Figure 5: The two leveled version of the architecture without feature map size change.

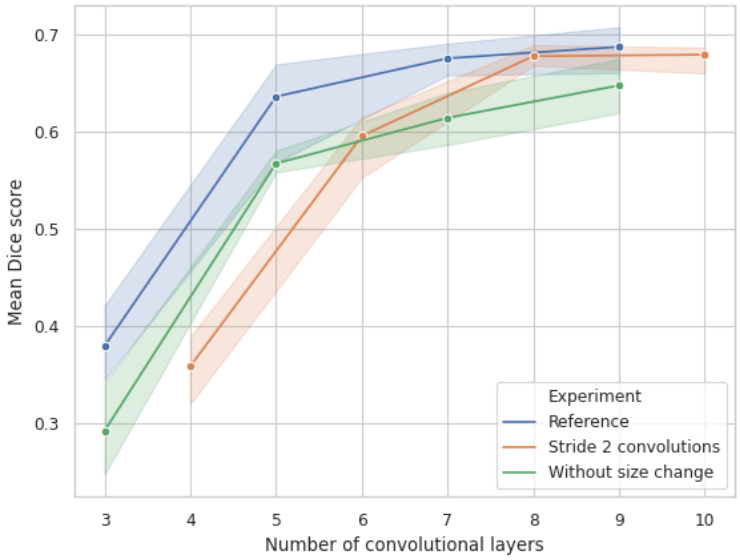


Figure 6: Feature map size change influence with confidence interval. Feature map size change improves segmentation. The couple max-pooling / up-sampling (reference) seems to have better performances with fewer convolutional layers than stride convolutions.

However it seems to bring more stability for these versions.

As the max-pooling / up-sampling version surpassed others in terms of Dice score and is slightly simpler than the strided convolution alternative for size reduction, we kept it as a reference for the following experiments. As depicted here, size reduction is one clue indicating why U-net architectures perform well. One other main feature of U-net is skip-connections.

3.4.2 Skip-connections

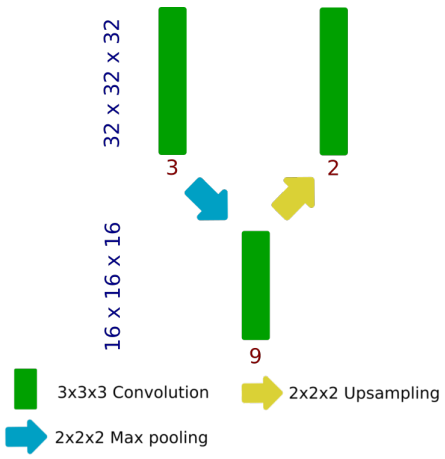


Figure 7: The two leveled version of the architecture without skip-connections.

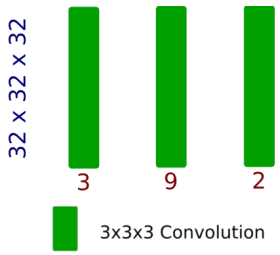


Figure 8: The two leveled version of plain architecture.

In U-net, skip-connections are made between convolution of the same level in the encoder and decoder. In order to demonstrate the utility of skip-connections, two additional experiments were performed:

- Without skip-connections: Fig. 7.
- Without skip-connections and size reduction (plain architecture): Fig. 8

Without skip connections, the quality of segmentation seems to collapse with depth in Fig. 9. Surprisingly, plain architectures (without skip connections and any change of feature map size), do not collapse with depth and performances are only slightly below the reference. Furthermore, compared to architecture without size change performances (Fig. 6), plain architecture performs better without skip-connections.

The previously analyzed architecture choices concerned structure, and the reference structure seemed the best. The following section focuses on the complexity of models.

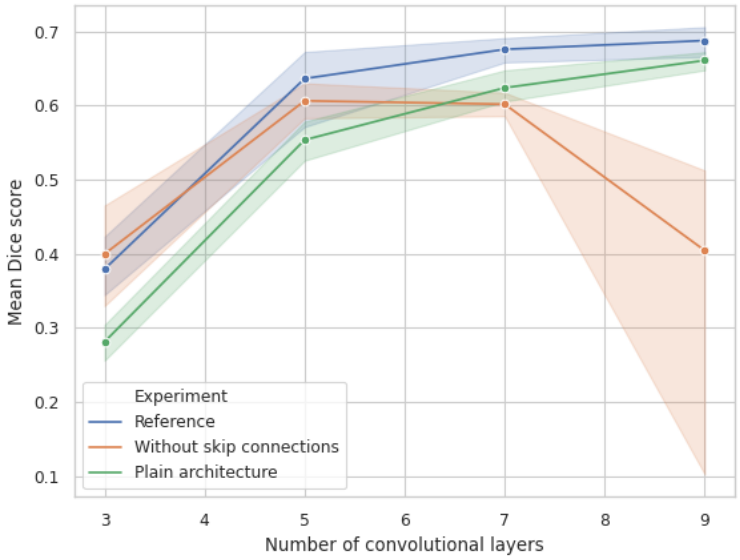


Figure 9: Skip connection study with confidence interval. Skip-connections avoid performance collapsing with feature map size change.

3.4.3 On augmenting architecture

Architectures are augmented by adding more convolutional layers and by adding more feature maps.

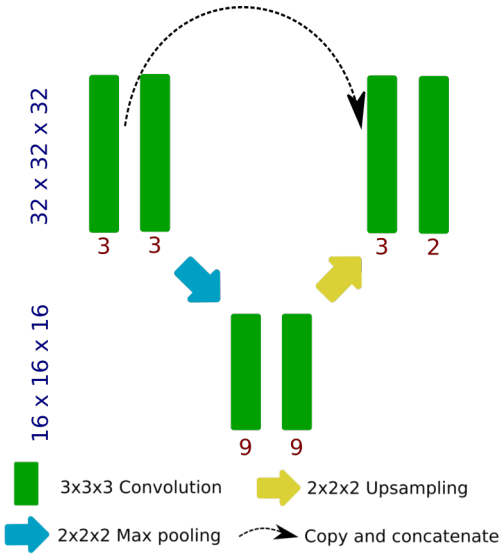


Figure 10: The two leveled version of the *two convolutions by level* architecture.

More convolutional layers

As in the U-net, an architecture with two convolutional layers by level (Fig. 10) is tested in this experiment.

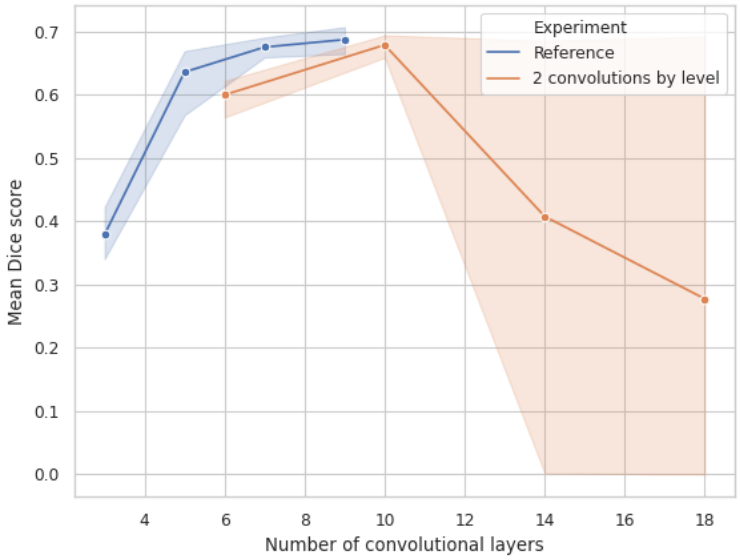


Figure 11: The effect of doubling the number of convolutional layers with confidence interval. Doubling convolutions rapidly lead to instability.

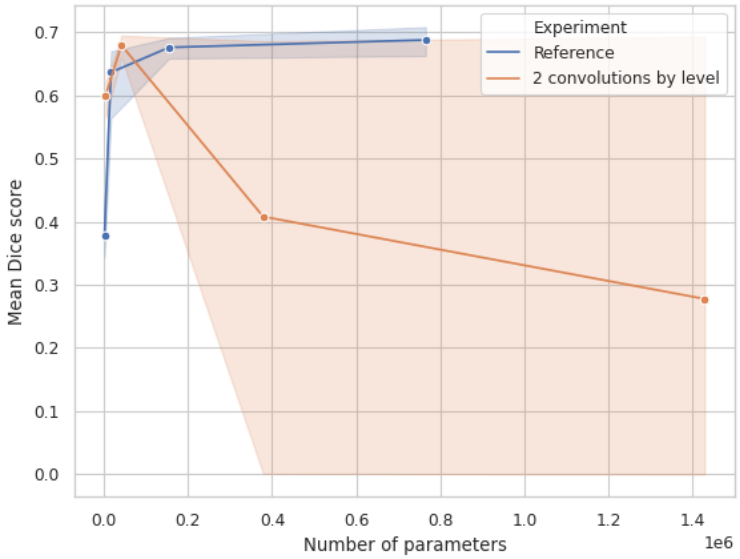


Figure 12: The effect of doubling the number of convolutional layers on the number of parameters with confidence interval. Doubling convolution can reduce the number of required learnable parameters, but rapidly loss stability with increasing complexity.

In Fig. 11, doubling the number of convolutional layers gives acceptable results for the first two points but induces a large decrease of performance for the last two points in addition to having a very large dispersion on the last point. This dispersion reveals that this architecture version failed for some trainings. This indicates that it is not stable and not very reproducible.

However, interestingly, in Fig. 12, with a smaller number of parameters, the *two convolution by level* architecture surpasses the reference, but with more parameters, the performances collapse.

More feature maps

The original U-net uses a larger amount of feature maps than here, ans consequently to ensure that our choice was acceptable, we performed additional experiments in order to see the influence of augmenting this number. Obviously, adding more feature maps increases the number of learnable parameters without changing the depth.

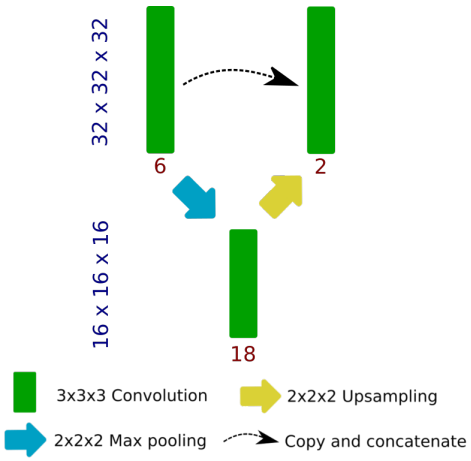


Figure 13: The two leveled version of the architecture beginning with 6 feature maps.

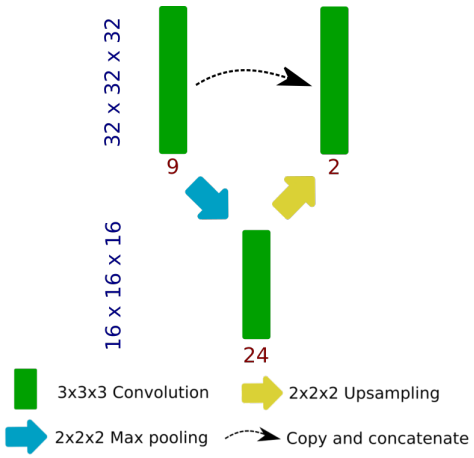


Figure 14: The two leveled version of the architecture beginning with 9 feature maps.

Level 1	Level 2	Level 3	Level 4	Level 5
6	18	36	96	108
9	24	45	108	112

Table 2: Number of feature maps at each level on new experiments.

Two different numbers of feature maps are evaluated, as detailed in Table 2:

- Beginning with 6 feature maps: Fig. 13
- Beginning with 9 feature maps: Fig. 14

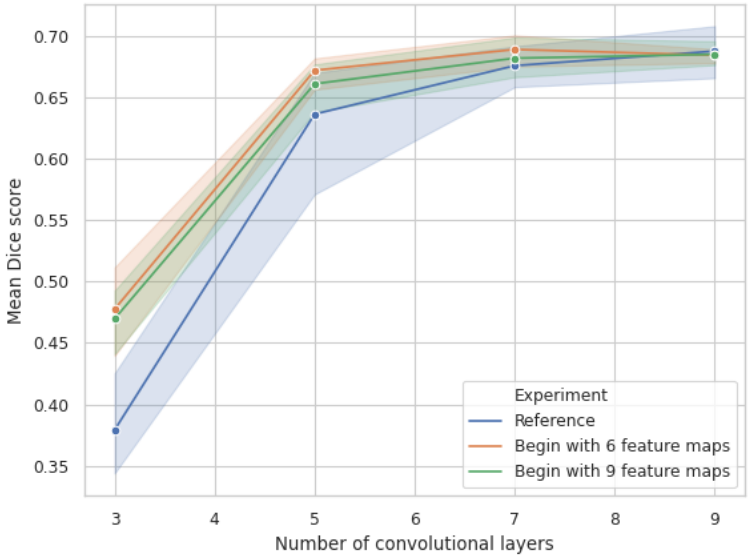


Figure 15: The effect of augmenting the number of feature maps with confidence interval. Slightly augmenting the number of feature maps improves segmentation mostly for smaller models, and generally improves stability. However, augmenting this number excessively may cause a loss of performance.

In Fig. 15, adding more feature maps, and so increasing the number of parameters, helps the CNN to perform a better segmentation for the shallower versions. In addition to that, it also seems to limit the dispersion of results, so it seems to help to obtain more robust learning models.

Interestingly, the version beginning with 6 feature maps seems to be better than the version beginning with 9 feature maps, which has more learnable parameters. The number of parameters has to be sufficient, but this number should not be too high in order to deliver the best performances.

We firstly thought that beginning with only 3 feature maps would not be enough to capture every brain texture detail. The last points of Fig. 15 indicates that with a sufficient number of parameters, beginning with only 3 feature maps was completely equivalent to beginning with more than 3.

3.4.4 Complexity and performances

In Fig. 16, we observe the influence of the depth of neural network on the Dice score. For almost all experiments, there is improvement with depth up to 7 convolutional layers and then a plateau. Almost all architectures converge to a mean Dice score around 0.6-0.7. This convergence seems to appear roughly from 5-7 convolutions. The best Dice score is obtained with 7 convolutions (4 levels) with architecture beginning with 6 feature maps. For two experiments, there is a collapse of performance with depth.

Fig. 17 demonstrates similar results for the number of learnable parameters. The convergence of performance is easier to see in this plot. This convergence seems to appear with approximately 100,000 learnable

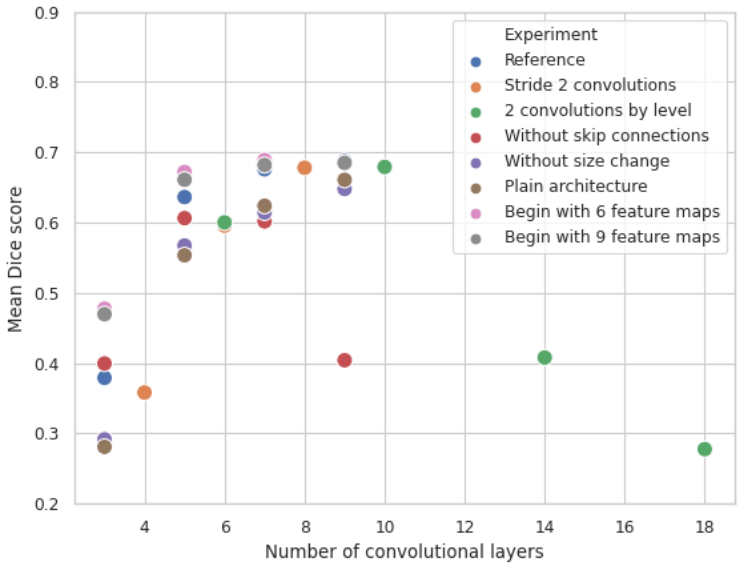


Figure 16: Dice mean of each experiment in function of the number of convolutional layers. Acceptable performances appear from 5 convolutional layers only, but adding more layers slightly improves then decreases the quality of segmentation.

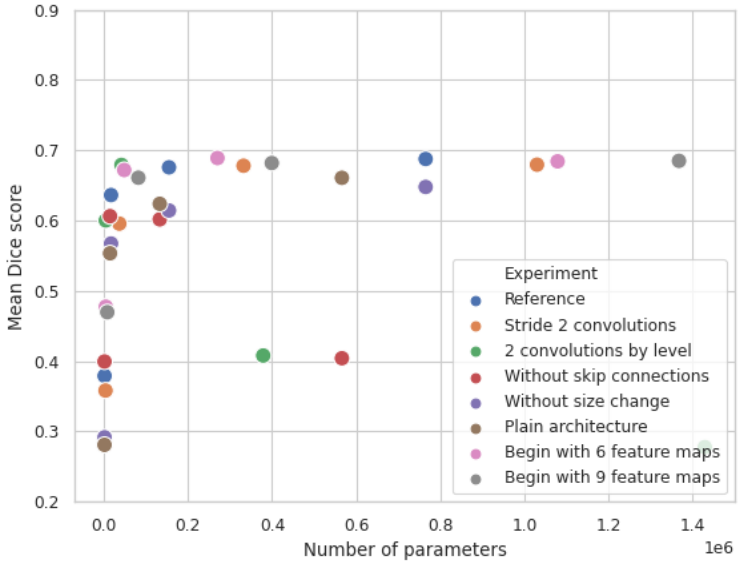


Figure 17: Dice mean of each experiment in function of the number of parameters to learn. Augmenting the number of parameters up to 300,000 for the given U-net like architectures is not necessary.

parameters, and our best Dice scores are with approximately 270,000 learnable parameters. The experiment with two convolutions per level delivers a better performance than others with the same number of parameters but rapidly collapses.

3.4.5 Sensitivity and precision

As described in 3.3, the predictor has to compromise between its sensitivity (its aptitude to detect the maximum correct voxels) and its precision (its ability to detect only true lesion voxels).

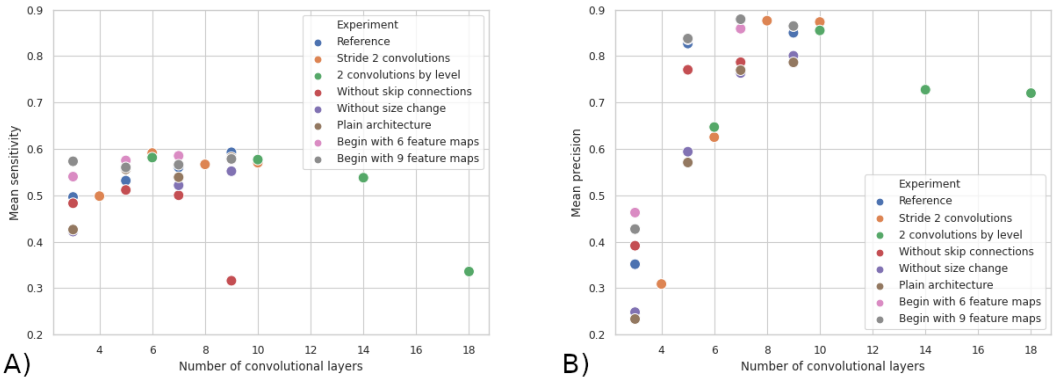


Figure 18: Sensitivity depending on the number of convolutional layers (A). Precision depending on the number of convolutional layers (B). Sensitivity of segmentation is far less improved than precision with bigger models.

Fig. 18A presents the sensitivity of all our architecture versions. It appears that sensitivity seems to converge at approximately 0.6. Some architectures reach the same sensitivity with only three convolutional layers.

Fig. 18B, the predictors seem to converge to a precision of more than 0.8. However, these levels are only reached after at least five convolutional layers. Furthermore, increasing model complexity benefits more to precision than to sensitivity.

These experiments indicate that:

- Reaching top level sensitivity requires fewer convolutional layers than with precision.
- Tested CNNs tend to be more precise than sensitive.

3.5 Segmentation visualization

Segmentation visualization can help find indices on the true nature of algorithm segmentation. In this part, only a 2 -dimensional slice of segmentation for one test exam is presented, although the algorithm generates 3-dimensional segmentation.

On FLAIR images, lesions appear as hyper intensities. On the example images in Fig. 19 and 20, there are periventricular lesions. The presence and number of this type of lesion is taken into account in the McDonald criteria [3]. In the figures below, we can visually compare the mean segmentation maps performed by different architectures.

In Fig. 19, the segmentation of each version of the reference architecture is shown. All segmentation maps are similar, but this is not the case for the two leveled version. The latter is the shallowest version with the smallest number of learnable parameters. This version tends to segment any hyper-intensity with fewer considerations for more intense texture of the healthy cortex.

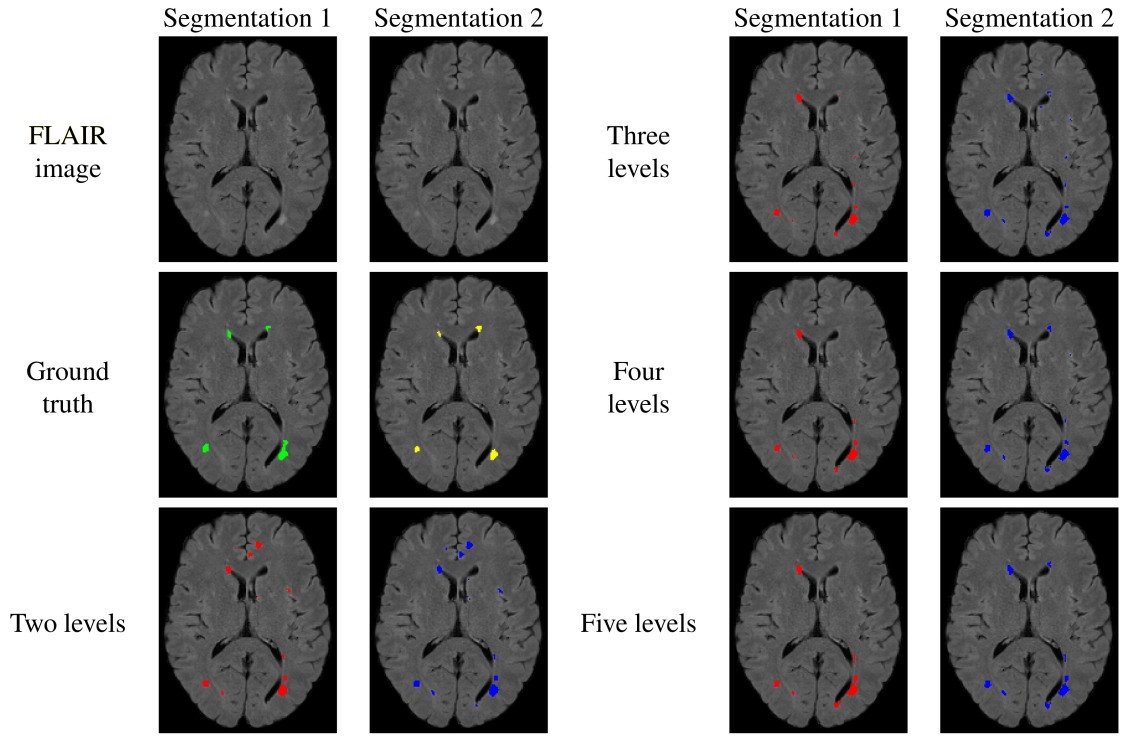


Figure 19: Mean segmentation of all versions of the reference. From the three to five leveled versions, segmentation is similar. The two leveled version has more false positives.

In Fig. 20, it is quite clear that the best segmentation versions of each architecture type are very similar. Globally the same hyper-intensities are detected by the best versions, whatever the architecture.

We also observe, that for each CNN prediction, segmentation maps are similar in both predictions, although ground truth maps are different. Segmentation 2 tends to highlight slightly more voxels than segmentation 1.

The segmentation predictions seem here to slightly overestimate lesions near the right occipital right horn of the lateral ventricle (in the bottom left part of images). In our training set, and for MS in general, periventricular lesions are much more represented than other types of lesions and some are very large. It is possible that, regarding the training set, the CNNs tend to be more sensitive to hyper-intensities near the ventricles.

4 Assessment of design choices

4.1 Feature map size change

Our results tend to show that adding feature map size reductions in the encoding path as well as feature map size increases in the decoding path improves the CNN performances. In other words, the conventional encoder and decoder model is relevant for segmentation.

Empirically, we showed that learning this feature map size change with strided convolutions gave no improvements in our experimental setup. The couple max-pooling and up-sampling was in practice better. These transformations are predefined and do not need to be learned, and this should simplify the learning process. The max-pooling is commonly used in CNNs in general [43] and is known for creating position invariance. Max-

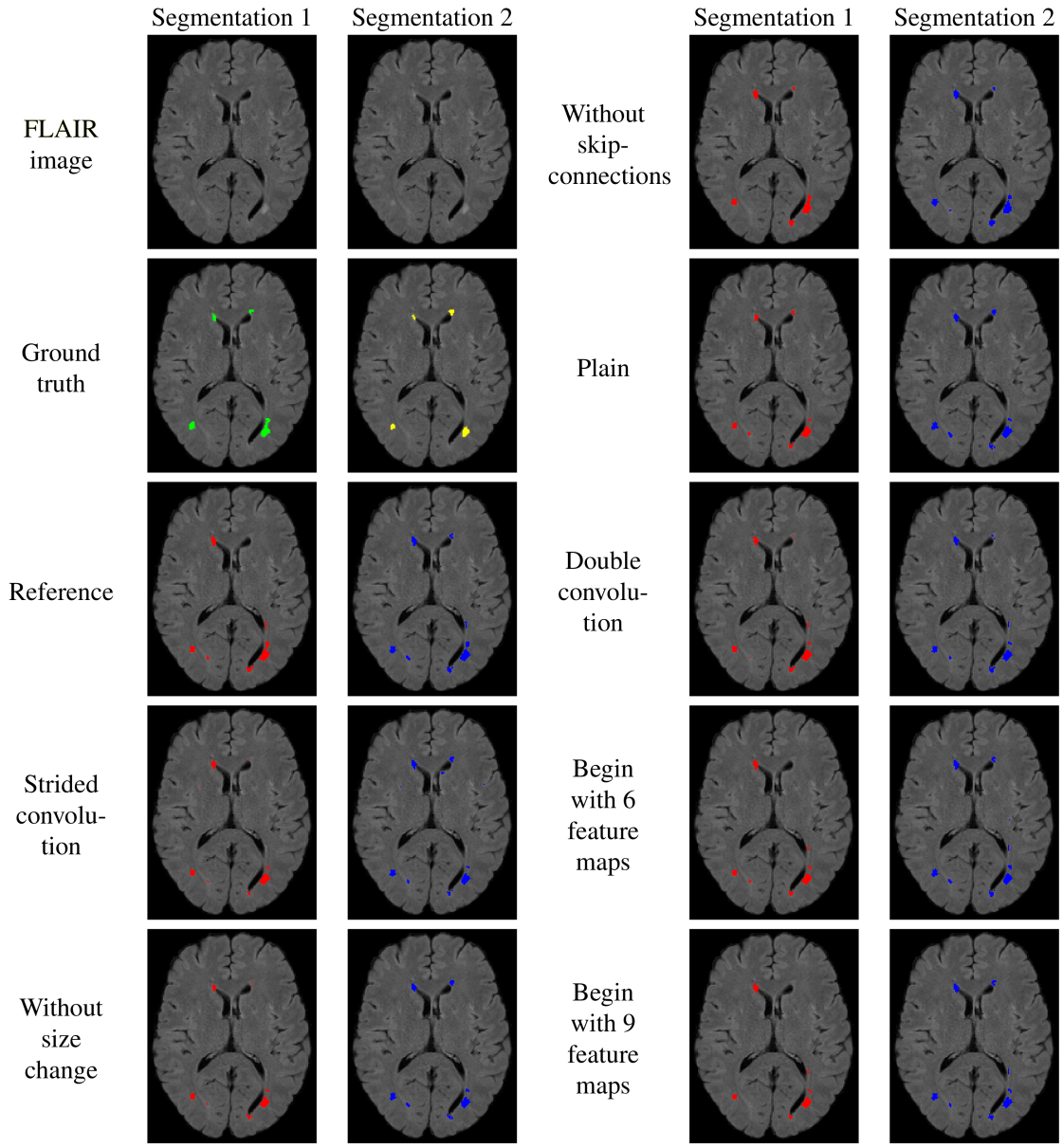


Figure 20: Mean segmentation of the best versions of all experiments. All segmentations of the best predictors are similar, and close to ground truth. However, none of them are perfectly matching ground truth.

pooling decreases the size of its inputs, reduces complexity and helps to have fast convergence, by selecting strong features bringing generalization improvements [44]. Up-sampling transformations have been shown to avoid checkerboard artifacts compared to strided transposed convolution [26].

However, feature map size change is not the only design choice that makes U-nets so widely used for segmentation.

4.1.1 Skip-connections

One key element of U-net is the use of long skip-connections between the encoder and the decoder. Skip connections are known for helping gradient to flow in deep neural networks, to allow gain of performance even with a very deep architecture and a faster convergence [45, 46].

Interestingly, in experiments, without feature map size change, skip connections are not needed. In addition, compared to plain architecture, they disturb segmentation. This can be explained by the fact that max-pooling is also known for causing overfitting [43, 47]. In the specific case of U-net, skip-connections may resolve the overfitting issues. The previously described max-pooling gain can thus be obtained.

4.2 Complexity and performances

Roy *et al.* [34]	**Best tested architecture (Fig. 17)**	Valverde *et al.* [29]	Zhang *et al.* [48]	Aslani *et al.* [49]
~ 240k	~ 270k	~ 470k	~ 29M	< 75M

Table 3: Number of parameters of some other existing methods

It is interesting to notice that there seems to exist a minimum number of necessary layers or parameters to perform our specific task. According to our experiments, we obtained our best performances with approximately 270 000 parameters, which is not the lowest number of parameters, compared to other existing approaches (see Table 3). This limit seems much lower than what can be found in the very deep and complex models used in natural image classification. One can argue that this extremely low need of parameters derives from the fact that the task of segmenting MS lesions on FLAIR images is quite simple since the inputs are very similar to each other. With brain MR images and preprocessing, the problems of pose or lighting, common in natural images, are very limited or null here.

In their work, Bianchini *et al.* [50] and Bengio *et al.* [51], indicate that deep neural networks implement more complex tasks with the same resources by augmenting depth. This phenomenon is regularly cited, such as in Zagoruyko *et al.* [46] and He *et al.* [52] and explains that very deep neural networks are targeted for state-of-the-art performances. Our empirical observations are in accord with this principle, since doubling convolutions for one level allows good performances to be delivered with only about 40,000 parameters (Fig. 12).

In some experiments, we also identified a descending of performances when complexity is too high, also observed by He *et al.* [52]. This can be explained by the curse of dimensionality, as it is also applicable to the number of parameters [51], but also by the vanishing gradient as deeper architectures seem to lose performance more quickly than the shallower architectures with the same number of parameters.

For almost all experiments we identified a convergence of performances from a given complexity of the model with a plateau of performances. This phenomenon is interesting as it indicates that the best qualities of segmentation obtained in experiments is reached at a given complexity and augmenting it has not improved results significantly in our experiments. As suggested by He *et al.* [52], with more complexity and additional regularizers to avoid fall of performance described in the paragraph above, a new plateau with better performances, closer to the radiologist analysis, could be reached.

4.3 Sensitivity and precision

We noticed that the shallower models can be sensitive but not precise. Precision is considerably improved with complexity than with sensitivity. Concerning MS lesion segmentation at least, it appears that this segmentation needs few convolutional layers to reach an acceptable level of sensitivity and precision comes with more depth according to our previous analysis [19]. For all our experiments, neural networks seem to find a compromise between sensitivity and precision. They tend to prefer precision to sensitivity, even if the Dice score loss gives as much importance to both metrics. This indicates that with a sufficient number of convolutional layers, it is easier to increase precision than sensitivity.

4.4 Dealing with two ground truths

Each tested architecture version predicts two segmentation maps: one for each ground truth mask given with the data set. It appears that both segmentations for all predictors are almost the same. So, tested neural networks make their own consensus and are not able to reproduce the behavior of two different radiologists. More complex neural networks could do so.

After observing and analyzing the above results, we decided to design and rigorously evaluate a last architecture by taking into account what we had learned.

5 Applying lessons in one shot

The main objective of the study was to design a minimum architecture in terms of complexity. The aim is to have the best compromise between performances and number of parameters. This is motivated by the fact that the simpler the model is, the simpler it is also to interpret.

The previous experiments give some insights for designing a compact U-net like architecture. We decided to design and test an architecture designed with the following specifications for our MPU-net:

- Skip-connections with max-pooling and up-sampling;
- Four levels with three convolutions per level;
- Beginning with 6 feature maps;
- Approximately 30,000 parameters, which is far less than 270, 000 but should deliver a good performance with high depth and reduce the risk of collapsing performances.

Level 1	Level 2	Level 3	Level 4
6	7	8	9

Table 4: Number of feature maps at each level.

By setting the number of the feature maps as presented in Table 4 we obtained an architecture with 33,332 learnable parameters and 21 convolutional layers in Fig.21.

The MPU-net was trained by using the entire ISBI 2015 MS segmentation challenge training set with a random leave-one-subject-out-cross-validation, and was tested on the official testing set. The segmentation performed by the first radiologist was used as ground truth since it was observed to be the closest to the predictions in general, and that using both expert segmentations as ground truth would have produced two similar prediction maps. The segmentation of the test set was submitted and evaluated in the ISBI 2015 MS segmentation

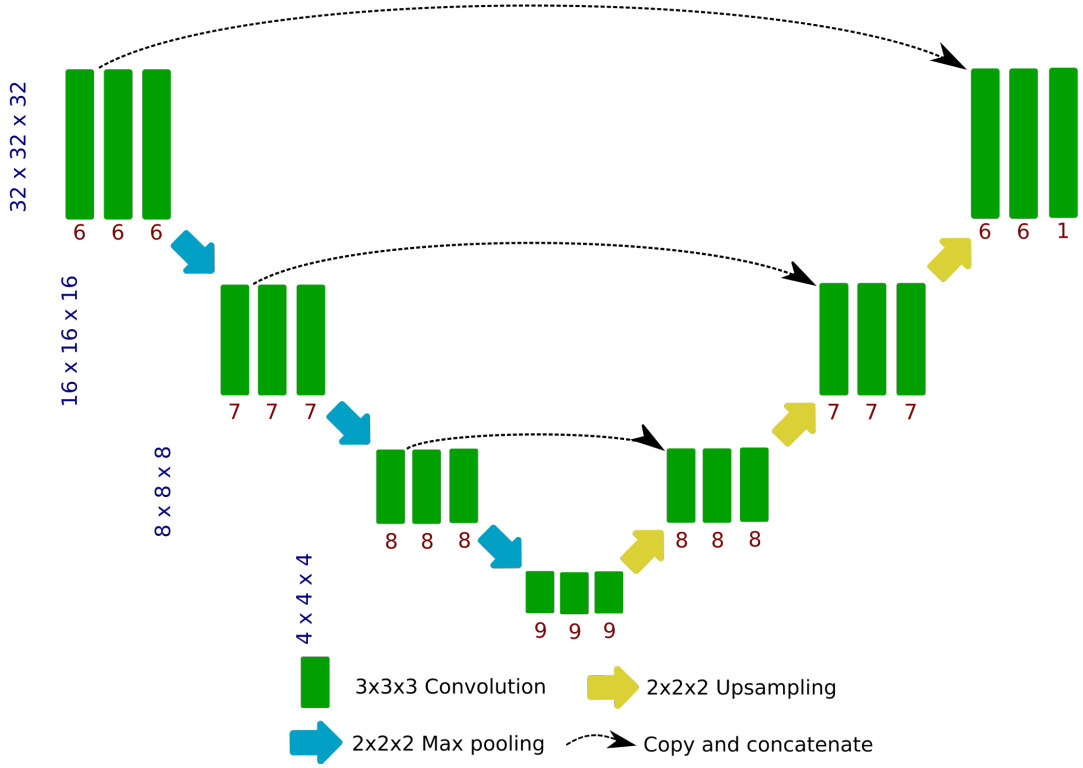


Figure 21: The proposed MPU-net architecture with 33,332 learnable parameters and 21 convolutional layers.

online leader-board [7]. The online submissions are ranked using the website score weighted by different metrics.

The MPU-net segmentation obtained a score of 90.591 which is comparable to human level segmentation [7].

ISBI score	Submission information	Submission year
93.358	Current best, unknown author from Vanderbilt University	2019
92.637	Hou *et al.* [53]	2019
92.486	Hashemi *et al.* [21]	2018
92.118	Aslani *et al.* [49]	2018
92.076	Andermatt *et al.* [54]	2017
91.44	Valverde *et al.* [55]	2017
90.591	**MPU-net(proposed)**	**2020**
90.283	Maier *et al.* [56]	2016
90.07	Birenbaum *et al.* [57]	2016

Table 5: Known published attempts to ISBI online MS lesion segmentation.

Compared to the found publications also submitting to this challenge (see Table 5), we have achieved acceptable segmentation since we have only used the FLAIR sequence, trained without transfer learning and only on the given ISBI training data set, without artificial data augmentation, dropout [36] or batch normalization [35] and without taking into account temporal data.

The segmentation prediction with the MPU-net runs easily on a laptop with 8 GB of memory and an Nvidia Quadro P1000 GPU and takes 12s per brain. In case of deployment in a clinical environment, the use of this architecture would not necessitate a big investment, thus facilitating the adoption of these type of techniques

to help radiologists.

6 Conclusion

With the goal of finding a compact architecture for the segmentation of MS lesions, we performed an empirical study to determine the relevance of some of the choices assumed in the design of CNN architectures in medical imaging and more precisely in our specific segmentation task. This study focuses on a simple adaptation of the famous U-net and assesses the influence of modifications to give some insights about why and when this architecture is relevant.

It appears that, for this lesion segmentation task, there is a rather wide range of complexity of model in terms of number of parameters and number of convolutional layers showing almost the same performances. In other words, a compact model can be equivalent in performance with a more complex one for our specific task and this should also be the case for other tasks. Furthermore, a predictor that is too complex shows a decrease of performance. The minimum complexity of the model can give some insight on the complexity of a given task. Since the classification of natural images is much more complex than brain lesion segmentation for machines, the minimum complexity of CNNs for our task is far lower that of state-of-the-art classification models, which have many millions of parameters. Also, by adding layers, tested CNNs tend to necessitate fewer parameters to learn.

In given experimental conditions, the use of max-pooling and up-sampling in encoder and decoder was the best method but only with the use of long skip-connections. These connections act as regularizers for avoiding loss of performance due to the complexity of the predictor.

Finally, thanks to all those observations, it has been possible to design a very light architecture for MS lesion segmentation, the MPU-net, that uses FLAIR images only. This architecture appears to deliver a good segmentation quality, comparable to that of the human even if its design and training are in fact very simple as it does not required complex engineering or regularizers and uses only 33,332 parameters.

As the power of a given CNN architecture is often not explained or partly explained in medical imaging, we aimed at defining the reasons for our choices of implementing a U-net like architecture. The present work is an attempt to present, clarify and explain the process of designing a CNN through the example of our specific interest in MS. This study can be extended to many points in this particular case but also to other tasks and domains.

In this work we have presented some of the reasons of the U-net architecture success. We showed that, for this task, with enough convolutional layers, very few parameters and feature maps are needed to reach human level analysis. Designing such efficient light architecture would simplify the use of artificial neural networks in hospitals by not requiring expensive servers. This could also help to explain CNN learned representations [58] in addition to other types of interpretability, and thus could increase trust [59] and lead to the adoption of deep learning methods in medical imaging.

Disclosures

Authors have no conflicts of interest to disclose.

References

[1] M. Pugliatti, S. Sotgiu, and G. Rosati, "The worldwide prevalence of multiple sclerosis," *Clinical neurology and neurosurgery*, vol. 104, no. 3, pp. 182–191, 2002.

[2] M. T. Wallin, W. J. Culpepper, E. Nichols, Z. A. Bhutta, T. T. Gebrehiwot, S. I. Hay, I. A. Khalil, K. J. Krohn, X. Liang, M. Naghavi, *et al.*, "Global, regional, and national burden of multiple sclerosis 1990–2016: a systematic analysis for the global burden of disease study 2016," *The Lancet Neurology*, vol. 18, no. 3, pp. 269–285, 2019.

[3] A. J. Thompson, B. L. Banwell, F. Barkhof, W. M. Carroll, T. Coetzee, G. Comi, J. Correale, F. Fazekas, M. Filippi, M. S. Freedman, *et al.*, "Diagnosis of multiple sclerosis: 2017 revisions of the mcdonald criteria," *The Lancet Neurology*, vol. 17, no. 2, pp. 162–173, 2018.

[4] D. García-Lorenzo, S. Francis, S. Narayanan, D. L. Arnold, and D. L. Collins, "Review of automatic segmentation methods of multiple sclerosis white matter lesions on conventional magnetic resonance imaging," *Medical image analysis*, vol. 17, no. 1, pp. 1–18, 2013.

[5] M. Styner, J. Lee, B. Chin, M. S Chin, O. Commowick, H.-H. Tran, V. Jewells, and S. Warfield, "3d segmentation in the clinic: A grand challenge ii: Ms lesion segmentation," *MIDAS Journal*, 11 2007.

[6] O. Commowick, F. Cervenansky, and R. Ameli, "Msseg challenge proceedings: Multiple sclerosis lesions segmentation challenge using a data management and processing infrastructure," 2016.

[7] A. Carass, S. Roy, A. Jog, J. L. Cuzzocreo, E. Magrath, A. Gherman, J. Button, J. Nguyen, F. Prados, C. H. Sudre, *et al.*, "Longitudinal multiple sclerosis lesion segmentation: resource and challenge," *NeuroImage*, vol. 148, pp. 77–102, 2017.

[8] A. Danelakis, T. Theoharis, and D. A. Verganelakis, "Survey of automated multiple sclerosis lesion segmentation techniques on magnetic resonance imaging," *Computerized Medical Imaging and Graphics*, vol. 70, pp. 83–100, 2018.

[9] P. Schmidt, C. Gaser, M. Arsic, D. Buck, A. Förschler, A. Berthele, M. Hoshi, R. Ilg, V. J. Schmid, C. Zimmer, *et al.*, "An automated tool for detection of flair-hyperintense white-matter lesions in multiple sclerosis," *Neuroimage*, vol. 59, no. 4, pp. 3774–3783, 2012.

[10] M. J. Fartaria, A. Roche, R. Meuli, C. Granziera, T. Kober, and M. B. Cuadra, "Segmentation of cortical and subcortical multiple sclerosis lesions based on constrained partial volume modeling," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 142–149, Springer, 2017.

[11] F. Vera-Olmos, H. Melero, and N. Malpica, "Random forest for multiple sclerosis lesion segmentation," *Proceedings of the 1st MICCAI Challenge on Multiple Sclerosis Lesions Segmentation Challenge Using a Data Management and Processing Infrastructure-MICCAI-MSSEG*, pp. 81–86, 2016.

[12] D. Goldberg-Zimring, A. Achiron, S. Miron, M. Faibel, and H. Azhari, "Automated detection and characterization of multiple sclerosis lesions in brain mr images," *Magnetic resonance imaging*, vol. 16, no. 3, pp. 311–318, 1998.

[13] S. K. D. Y. T. M. Y. Y. M. O. F. T. Y. K. Jumpei Kuwazuru, Hidetaka Arimura, "Automated detection of multiple sclerosis candidate regions in mr images: false-positive removal with use of an ann-controlled level-set method," *Radiological Physcis and Technology*, 2012.

[14] O. Commowick, A. Istace, M. Kain, B. Laurent, F. Leray, M. Simon, S. C. Pop, P. Girard, R. Ameli, J.-C. Ferré, *et al.*, "Objective evaluation of multiple sclerosis lesion segmentation using a data management and processing infrastructure," *Scientific reports*, vol. 8, no. 1, p. 13650, 2018.

[15] S. Valverde, M. Cabezas, E. Roura, S. González-Villa, J. Salvi, A. Oliver, and X. Lladó, "Multiple sclerosis lesion detection and segmentation using a convolutional neural network of 3d patches," *MSSEG Challenge Proceedings: Multiple Sclerosis Lesions Segmentation Challenge Using a Data Management and Processing Infrastructure*, p. 75, 2016.

[16] R. McKinley, R. Wepfer, T. Gundersen, F. Wagner, A. Chan, R. Wiest, and M. Reyes, "Nabla-net: A deep dag-like convolutional architecture for biomedical image segmentation," in *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 119–128, Springer, 2016.

[17] T. Nair, D. Precup, D. L. Arnold, and T. Arbel, "Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 655–663, Springer, 2018.

[18] T. Brosch, L. Y. W. Tang, Y. Yoo, D. K. B. Li, A. Traboulsee, and R. Tam, "Deep 3d convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation," *IEEE Transactions on Medical Imaging*, vol. 35, pp. 1229–1239, May 2016.

[19] A. Fenneteau, P. Bourdon, D. Helbert, C. Fernandez-Maloigne, C. Habas, and R. Guillevin, "Learning a CNN on multiple sclerosis lesion segmentation with self-supervision," in *3D Measurement and Data Processing, IS&T Electronic Imaging 2020 Symposium*, (San Francisco, United States), #Jan# 2020. Best paper award 3D Measurement and Data Processing.

[20] R. McKinley, R. Wepfer, F. Aschwanden, L. Grunder, R. Muri, C. Rummel, R. Verma, C. Weisstanner, M. Reyes, A. Salmen, *et al.*, "Simultaneous lesion and neuroanatomy segmentation in multiple sclerosis using deep neural networks," *arXiv preprint arXiv:1901.07419*, 2019.

[21] S. R. Hashemi, S. S. M. Salehi, D. Erdogmus, S. P. Prabhu, S. K. Warfield, and A. Gholipour, "Asymmetric loss functions and deep densely-connected networks for highly-imbalanced medical image segmentation: Application to multiple sclerosis lesion detection," *IEEE Access*, vol. 7, pp. 1721–1735, 2019.

[22] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[23] F. Eitel, E. Soehler, J. Bellmann-Strobl, A. U. Brandt, K. Ruprecht, R. M. Giess, J. Kuchling, S. Asseyer, M. Weygandt, J.-D. Haynes, *et al.*, "Uncovering convolutional neural network decisions for diagnosing multiple sclerosis on conventional mri using layer-wise relevance propagation," *NeuroImage: Clinical*, vol. 24, p. 102003, 2019.

[24] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller, "Toward interpretable machine learning: Transparent deep neural networks and beyond," *arXiv preprint arXiv:2003.07631*, 2020.

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, vol. 9351 of *LNCS*, pp. 234–241, Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).

[26] F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-Hein, "Brain tumor segmentation and radiomics survival prediction: Contribution to the brats 2017 challenge," in *International MICCAI Brainlesion Workshop*, pp. 287–297, Springer, 2017.

[27] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[28] Y. Feng, H. Pan, C. H. Meyer, and X. Feng, "A self-adaptive network for multiple sclerosis lesion segmentation from multi-contrast mri with various imaging protocols.," *CoRR*, vol. abs/1811.07491, 2018.

[29] S. Valverde, M. Salem, M. Cabezas, D. Pareto, J. C. Vilanova, L. Ramió-Torrentà, A. Rovira, J. Salvi, A. Oliver, and X. Lladó, "One-shot domain adaptation in multiple sclerosis lesion segmentation using convolutional neural networks," *NeuroImage. Clinical*, p. 101638, 2018.

[30] F. La Rosa, M. J. Fartaria, T. Kober, J. Richiardi, C. Granziera, J.-P. Thiran, and M. B. Cuadra, "Shallow vs deep learning architectures for white matter lesion segmentation in the early stages of multiple sclerosis," in *International MICCAI Brainlesion Workshop*, pp. 142–151, Springer, 2018.

[31] L. G. Nyúl, J. K. Udupa, and X. Zhang, "New variants of a method of mri scale standardization," *IEEE transactions on medical imaging*, vol. 19, no. 2, pp. 143–150, 2000.

[32] M. M. McCormick, X. Liu, L. Ibanez, J. Jomier, and C. Marion, "Itk: enabling reproducible research and open science," *Frontiers in neuroinformatics*, vol. 8, p. 13, 2014.

[33] T. S. Yoo, M. J. Ackerman, W. E. Lorensen, W. Schroeder, V. Chalana, S. Aylward, D. Metaxas, and R. Whitaker, "Engineering and algorithm design for an image processing api: a technical report on itk-the insight toolkit," *Studies in health technology and informatics*, pp. 586–592, 2002.

[34] S. Roy, J. A. Butman, D. S. Reich, P. A. Calabresi, and D. L. Pham, "Multiple sclerosis lesion segmentation from brain mri via fully convolutional neural networks," *arXiv preprint arXiv:1803.09172*, 2018.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[37] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, p. 3, 2013.

[38] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[39] F. Chollet *et al.*, "Keras." https://keras.io, 2015.

[40] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[42] B. Efron, "Bootstrap methods: another look at the jackknife," in *Breakthroughs in statistics*, pp. 569–593, Springer, 1992.

[43] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *arXiv preprint arXiv:1301.3557*, 2013.

[44] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 342–347, IEEE, 2011.

[45] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Deep Learning and Data Labeling for Medical Applications*, pp. 179–187, Springer, 2016.

[46] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[47] B. Graham, "Fractional max-pooling," *arXiv preprint arXiv:1412.6071*, 2014.

[48] C. Zhang, Y. Song, S. Liu, S. Lill, C. Wang, Z. Tang, Y. You, Y. Gao, A. Klistorner, M. Barnett, *et al.*, "Ms-gan: Gan-based semantic segmentation of multiple sclerosis lesions in brain magnetic resonance imaging," in *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, IEEE, 2018.

[49] S. Aslani, M. Dayan, L. Storelli, M. Filippi, V. Murino, M. A. Rocca, and D. Sona, "Multi-branch convolutional neural network for multiple sclerosis lesion segmentation," *NeuroImage*, vol. 196, pp. 1–15, 2019.

[50] M. Bianchini and F. Scarselli, "On the complexity of neural network classifiers: A comparison between shallow and deep architectures," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 8, pp. 1553–1565, 2014.

[51] Y. Bengio, Y. LeCun, *et al.*, "Scaling learning algorithms towards ai," *Large-scale kernel machines*, vol. 34, no. 5, pp. 1–41, 2007.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[53] B. Hou, G. Kang, X. Xu, and C. Hu, "Cross attention densely connected networks for multiple sclerosis lesion segmentation," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 2356–2361, IEEE, 2019.

[54] S. Andermatt, S. Pezold, and P. C. Cattin, "Automated segmentation of multiple sclerosis lesions using multi-dimensional gated recurrent units," in *International MICCAI Brainlesion Workshop*, pp. 31–42, Springer, 2017.

[55] S. Valverde, M. Cabezas, E. Roura, S. González-Villà, D. Pareto, J. C. Vilanova, L. Ramió-Torrentà, À. Rovira, A. Oliver, and X. Lladó, "Improving automated multiple sclerosis lesion segmentation with a cascaded 3d convolutional neural network approach," *NeuroImage*, vol. 155, pp. 159–168, 2017.

[56] O. Maier and H. Handels, "Ms lesion segmentation in mri with random forests," *Proc. 2015 Longitudinal Multiple Sclerosis Lesion Segmentation Challenge*, pp. 1–2, 2015.

[57] A. Birenbaum and H. Greenspan, "Longitudinal multiple sclerosis lesion segmentation using multi-view convolutional neural networks," in *Deep Learning and Data Labeling for Medical Applications*, pp. 58–67, Springer, 2016.

[58] W. Samek and K.-R. Müller, "Towards explainable artificial intelligence," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 5–22, Springer, 2019.

[59] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.