Robustness of brain tumor segmentation

Sabine Müller,[a,b,c,d,*] **Joachim Weickert,**[c] **and Norbert Graf**[d]
[a]Fraunhofer ITWM, Competence Center High Performance Computing,
Kaiserslautern, Germany
[b]Fraunhofer Center Machine Learning, Germany
[c]Saarland University, Faculty of Mathematics and Computer Science, Mathematical Image
Analysis Group, Saarbrücken, Germany
[d]Saarland University Medical Center, Department of Pediatric Oncology and Hematology,
Homburg, Germany

Abstract

Purpose: The segmentation of brain tumors is one of the most active areas of medical image analysis. While current methods perform superhuman on benchmark data sets, their applicability in daily clinical practice has not been evaluated. In this work, we investigate the generalization behavior of deep neural networks in this scenario.

Approach: We evaluate the performance of three state-of-the-art methods, a basic U-Net architecture, and a cascadic Mumford–Shah approach. We also propose two simple modifications (which do not change the topology) to improve generalization performance.

Results: In these experiments, we show that a well-trained U-network shows the best generalization behavior and is sufficient to solve this segmentation problem. We illustrate why extensions of this model in a realistic scenario can be not only pointless but even harmful.

Conclusions: We conclude from these experiments that the generalization performance of deep neural networks is severely limited in medical image analysis especially in the area of brain tumor segmentation. In our opinion, current topologies are optimized for the actual benchmark data set but are not directly applicable in daily clinical practice.

© *2020 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.JMI.7.6.064006]

Keywords: deep learning; segmentation; generalization; brain tumors.

Paper 19324RR received Dec. 31, 2019; accepted for publication Dec. 11, 2020; published online Dec. 30, 2020.

1 Introduction

Since AlexNet[1] won the ImageNet Large Scale Visual Recognition Competition challenge,[2] the influence of deep neural networks has increased dramatically in all domains of image processing and pattern recognition. From classification to object tracking and image and video segmentation, new approaches are typically based on deep learning strategies.[3–5] These approaches are also gaining more and more influence in the field of medical image processing. Since Ronneberger et al.[6] proposed the U-Net structure, this model is the de facto standard method in the field of medical image segmentation. While the original approach could be trained with relatively few examples in a short time, current models require a large amount of data and a very time consuming and computationally intense training cycle.[7,8] Since several years, it is common practice to compare the performance of segmentation approaches on benchmark data sets. One of the best-known data sets has been provided in the scope of the "Multimodal Brain Tumor Segmentation Challenge" (BraTS).[9]

Brain tumors account only for a very small fraction of all types of cancer but are also among the most fatal forms of this deadly disease. Gliomas, developing from the glial cells, are the most frequent primary brain tumors. The fast growing and more aggressive types of gliomas called

*Address all correspondence to Sabine Müller, sabine.b.mueller@itwm.fraunhofer.de

2329-4302/2020/$28.00 © 2020 SPIE

high-grade gliomas (HGG), come with an median overall survival rate of up to 15 months.[10] The standard diagnosis technique for brain tumor is magnetic resonance imaging (MRI)[11] providing detailed information about the tumor and the surrounding brain. Tumor segmentation is of crucial importance in surgical and treatment planning, while fully automated segmentation is a challenging task especially for HGG: they usually show diffuse and irregular boundaries and have intensities overlapping with normal brain tissue caused by peritumoral edema. Moreover, acquisition parameters are not standardized, and different parameter settings can have a substantial impact on the visual appearance of the tumor. This makes it difficult to compare the quality of different methods for brain tumor segmentation. As a step toward an unbiased performance evaluation, the BraTS database has been created,[5,9,12] and many recent approaches report benchmark results on either the full data set or parts of it.[7,8,13]

Since this data set has been used for seven years now to compare different approaches with each other, a major drawback has manifested itself over this long time: The main focus of the researchers is not to present the most robust network with best generalization behavior, but to maximize the performance metrics of the BraTS benchmark dataset. Thus, it can happen that the increasingly complicated models are not useful in a real clinical scenario as they heavily overfit the test set: this benchmark data set is saturated.

Nowadays, models do not get better in a general sense but current best approaches overfit the test set more than others. Typically, a test set is meant to be a biased version of a specific problem representation, i.e., all humans with high-grade brain tumors in MRI sequences. To show a statistically significance of one benchmark result being superior to another one, an appropriate sample size is necessary.[14] Unfortunately, the sample size of the BraTS test set is too small to provide a statistical significant difference in the best-performing methods.[5]

In addition, the main strength of deep learning approaches of fitting the underlying data distribution is also their greatest weakness: in a clinical setting, the assumption that training and test data belong to the exact same distribution is typically not correct.

In this work, we evaluate the robustness of different segmentation methods with respect to disturbances in the underlying distribution. We investigate three state-of-the-art methods as well as a simple and intuitive scheme based on the powerful Mumford–Shah functional.

In addition, we suggest two simple and straight forward modifications that allow to increase the generalization performance of the evaluated deep neural networks. Finally, we demonstrate that our semi-supervised segmentation approach is a powerful post-processing step, that allows to robustify the predictions of deep neural networks with respect to disturbances in the test data set. Hence, we combine the best of two worlds: We still can learn the class distributions of the targeted objects while exploiting the robustness of energy formulations to modifications in the data.

1.1 *Contributions*

Our contributions are as follows. First, we show that current state-of-the-art neural network architectures for brain tumor segmentation have a poor generalization behavior and massively overfit the training data. Second, we apply two simple but powerful modifications from the classification community to semantic image segmentation. These alterations allow for higher generalization performance at inference while reducing the network size. Last but not least, we suggest an effective post-processing step that massively improves the segmentation result when disturbances in the data are an issue.

1.2 *Structure of the Paper*

In Sec. 2, we start with the explanation of different state-of-the-art deep neural networks for brain tumor segmentation and a classical approach that does not require training and is based on the well-known Mumford–Shah functional. We then illustrate two new approaches that help to improve the generalization performance of deep neural networks.

In Sec. 3, we analyze the sensitivity of the previously presented deep neural networks to slight changes in the data distributions. We complete this work with our conclusions in Sec. 4.

2 Materials and Methods

The baseline of our evaluation is a segmentation approach that does not require training: a cascadic Mumford–Shah (CMS) cartoon model.[15] Since we can almost eliminate an overfit to the underlying data set, none of the compared deep learning models should score below the performance of this method.

We begin our evaluation with the de-facto standard model for image segmentation with deep neural networks: the U-Net architecture.[6] We continue with an extended version, namely, the No NewNet approach showing high performance on BraTS 2018.[8] Afterward, we take the winner of last years' challenge into account: NVDLMED,[16] using autoencoder regularization to improve the segmentation accuracy.[7] Last but not least, we investigate the third place of the BraTS 2018 challenge, using a cascade of several neural networks for segmentation.[17]

2.1 *Segmentation Approaches*

2.1.1 *Cascadic Mumford–Shah cartoon model*

The segmentation approach based on a CMS cartoon model,[18] does not require training. Its basic idea is to exploit a simple but reliable prior information: Brain tumors are usually brighter than the surrounding brain tissues on T_2-Flair images. Typically, it is difficult to identify a good parameter setting for these kinds of segmentation approaches. Fortunately, our method only depends on a single parameter choice, that can be automatically selected.

Let us consider a cubic data domain $\Omega \subset \mathbb{R}^3$ and some volumetric data set $\boldsymbol{f}: \Omega \to \mathbb{R}^m$. For our application, its m channels describe different MRI modalities, such as T_1, T_{1c}, T_2, and T_2-Flair. Then a segmentation of $\boldsymbol{f}$ by means of the Mumford–Shah cartoon model[19,20] minimizes the energy functional

$$E(\boldsymbol{u}, C) = \sum_i \int_{\Omega_i} \|\boldsymbol{u} - \boldsymbol{f}\|^2 \mathrm{d}\boldsymbol{x} + \nu \ell(C). \tag{1}$$

Here, the a priori unknown number of segments Ω_i partition the data domain Ω, the function $\boldsymbol{u}$ denotes a piecewise constant approximation of $\boldsymbol{f}$, $\|.\|$ is the Euclidean norm in $\mathbb{R}^m$, and the segment boundaries C have a (Hausdorff) length of $\ell(C)$. The first term of the energy is a data term that penalizes fluctuations within each segment, while the second term favors short segment boundaries. The parameter $\nu > 0$ allows to weight the boundary length in relation to the inhomogeneities within each segment. Obviously, the choice of ν is of crucial importance: The higher the value of this parameter, the less segments are contained in the final result. In Fig. 1, the number of segments decreases with increasing penalization of the boundary length. At the same time, the inhomogeneities within individual segments increases. On MRI T_2-Flair scans, HGG contains areas that are brighter than the brain tissue. We use this prior knowledge and segment for a bright outlier in intensity in the following way: We start with the parameter $\nu = 400{,}000$

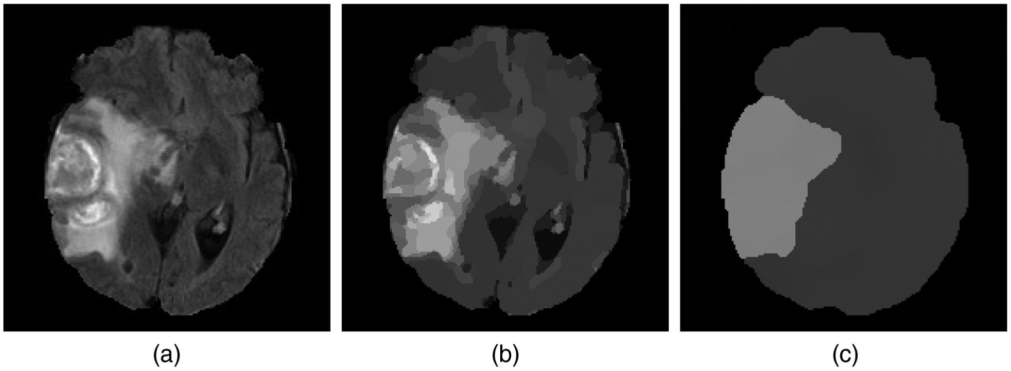

Fig. 1 Exemplary results for different penalizations of the boundary length. (a) T_2-Flair input image. (b) Result for $\nu = 1000$. (c) Result for $\nu = 340000$.

and check if this gives a segmentation into two areas: the tumor and the background. If the area of the thresholded tumor is larger than 50% of the brain volume, this is an indication that ν was too large such that the tumor has been merged with its background. In this case, we reduce ν by 15% and start the procedure again. This approach is repeated recursively until we have a segmentation where the tumor volume is below 50% of the brain volume.

We use this first segmentation to determine further tumor subcomponents: We minimize the Mumford–Shah cartoon model again with a very small boundary penalization ($\nu = 1$), but this time exclusively on the T_{1c} scans and in the previously defined segment. Afterward, we use Otsu's thresholding to identify the active tumor, i.e., enhancing- and non-enhancing tumor core. In this way, we get a splitting of the complete tumor region into active tumor and necrosis/edema. To get the final subcomponents, we apply Otsu's method on both subcomponents and split the first component, i.e., active tumor, into its enhancing and non-enhancing part and the second subcomponent into necrosis and edema.

2.1.2 *U-Net*

The U-Net architecture[6] is probably the most dominant topology in image segmentation. The intuition behind its structure is to re-use already learned feature mappings: This architecture can be split into two components, an encoder and a decoder branch connected by a bottleneck (see Fig. 2). While the first one learns feature mappings and contracts the image to its vector representation in the latent space (i.e., the bottleneck), the decoder part reconstructs an image of the original size using the previously learned feature maps, see Ref. 6 for more details. In this way, the structural integrity is maintained while distortions due to lost locality are reduced. To introduce locality to the massively abstracted feature representations, Ronneberger et al.[6] apply skip-connections, allowing a re-usage of already learned filters.

In its original formulation, the U-Net architecture was developed for 2D cell images. However, its extension to 3D images (necessary for volumetric MRI data) is straightforward—the architecture is identical and only replaces all 2D operators with their corresponding 3D variants.[4]

In the following, we will discuss the No NewNet topology.[8] This recent work shows a very high performance on several datasets. It was developed on the basic assumption that already the original U-Net architecture is very powerful and most extensions of its design are not necessary and too complicated. Since we will follow this assumption in a quite similar way, we explain this work in detail.

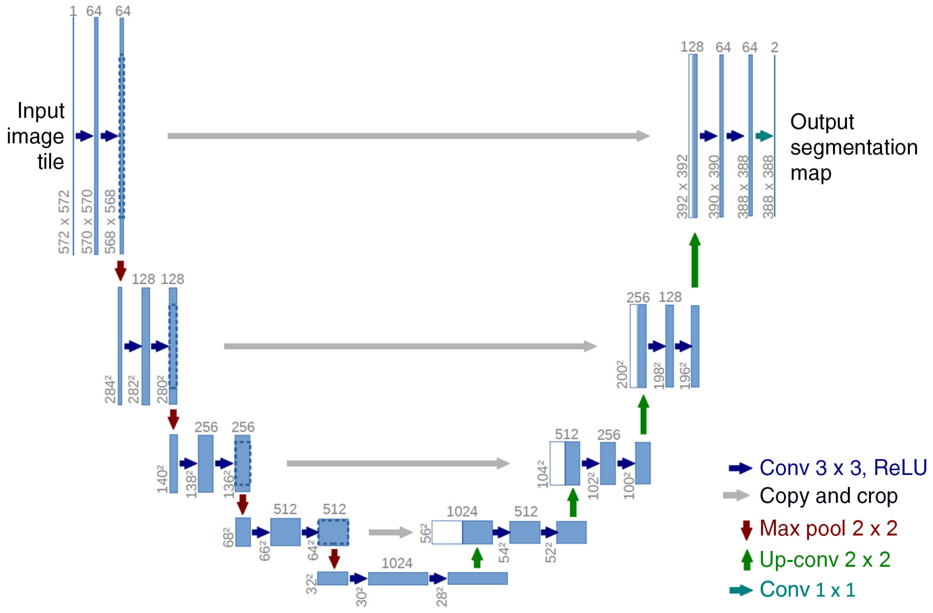

Fig. 2 Basic structure of U-Net approaches. Each blue box indicates a multidimensional feature map, arrows correspond to operations. Image courtesy of Ronneberger et al.[6]

2.1.3 *No NewNet*

Since the publication of the U-Net architecture, the encoder–decoder strategy has become the dominant approach in image segmentation. Nowadays, almost all new developments in this field are based on architectural modifications of this topology.[7,8,21]

In the meantime, it is almost impossible to predict which architecture might be suitable for a problem due to the multitude of possible extensions: Each of these possibilities has been tested on a specific data set. Unfortunately, it is an inherent part of deep learning that there is an architectural overfit to the data set used, making it almost impossible to decide whether an adjustment is appropriate in a different context.

Isensee et al.[8] implemented a number of these variants and evaluated their usefulness. It is not surprising that they found most of these extensions to be pointless in a general context, compared to a well trained U-Net model. Overall, they claim that a generic U-Net architecture with a few minor modifications can be sufficient to provide competitive performance.

Probably, the only significant difference to the original scheme is a normalization after each convolution layer (see Fig. 3). Obviously, this is fully consistent with current findings that normalization leads to wider optima (with higher generalization performance) in the loss surface.[22] To optimize the performance of the model on BraTS benchmark data, the authors suggest a set of additional extensions, see Ref. 8 for more details. All in all, each of those steps contributed some improvement to the overall performance.[8] While most of their adjustments had only minor influence on the segmentation performance, the postprocessing step and the training on additional data noticeably improved the error metric by 0.032 (enhancing core) and 0.013 (complete tumor), respectively. This indicates that their main improvement in performance was caused by the inclusion of more training data, i.e., by reducing the overfit of the model to the training distribution.

2.1.4 *NVDLMED: autoencoder regularization*

The winner of the 2018 BraTS challenges also followed a basic U-Net architecture.[7] While the backbone can still be reduced to an encoder–decoder structure, the author dramatically increased the model size and extended most of the basic topology by additional operations: Although the encoder branch is still similar, its building blocks are massively changed (see Fig. 4). Probably the most important change is an additional variational autoencoder branch reconstructing the input image to itself. This sub-network is then used during the training phase as regularization.

To improve the model performance, NVDLMED is built on an ensemble of 10 different networks. Unfortunately, this setting results in a very large network, that can only be trained on at least NVidia V100 GPUs, or on a CPU cluster.

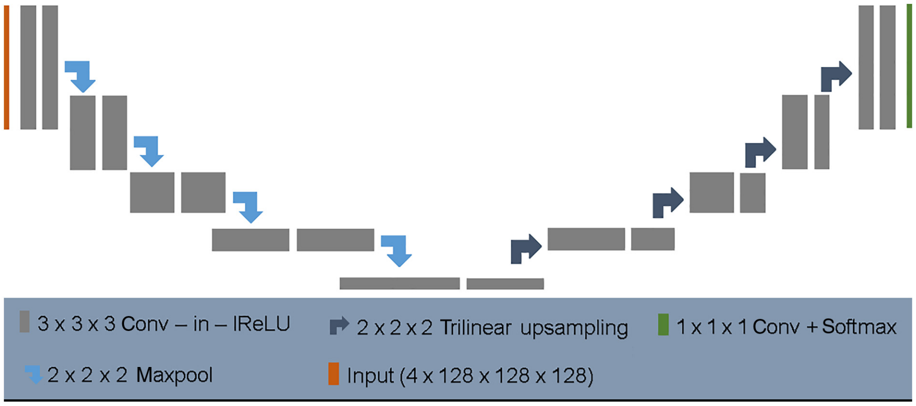
3 x 3 x 3 Conv – in – lReLU
2 x 2 x 2 Trilinear upsampling
1 x 1 x 1 Conv + Softmax
2 x 2 x 2 Maxpool
Input (4 x 128 x 128 x 128)

Fig. 3 Architecture of the No NewNet model. Each gray box corresponds to a series of convolution, instance normalization, and leaky ReLu. Arrows indicate upsampling and downsampling operations, respectively. Image courtesy of Isensee et al.[8]

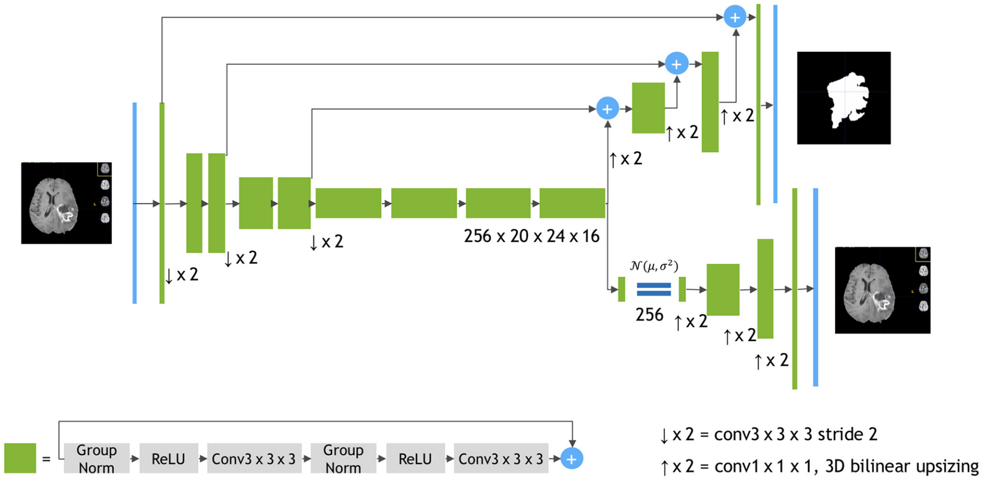

Fig. 4 Architecture of NVDLMED. In contrast to the basic U-Net structure, a second decoder branch is implemented. Image courtesy of Myronenko.[16]

2.1.5 *Cascadic neural networks*

Zhou et al.[17] approach the task of brain tumor segmentation from a slightly different perspective. While most of the state-of-the-art methods consider the identification of the complete tumor and its subcomponents as a single problem, the authors decompose the segmentation challenge into three different sub-tasks. In the first step, their method performs a coarse segmentation to detect the complete tumor. Afterward, the segmentation is refined and intra-tumoral classes are segmented. Finally, this segmentation is again optimized to classify the enhancing tumor core. This cascade of segmentation tasks is realized with two different network topologies. On the one hand, Zhou et al. make use of 3D FusionNets[23] (see Fig. 5) to extract the multi-scale context information. On the other, they apply one-pass multi-task networks.[24] In addition, Zhou et al.[17] perform several modifications, such that the final ensemble contains seven different neural network architectures whose results are averaged for the final model prediction.

2.1.6 *Preprocessing*

Typically MR images are recorded from different hospitals with varying scanners and no standardized parameter settings. This results in strong variations in the MR intensities: Even the same sequence of the same patient (e.g., T_2) acquired at the same machine, can differ dramatically due to inconsistent parameter choices. Deep neural networks learn the data distribution provided by the training set. Hence, it is essential that the value range in the training data corresponds to the range present in the test set. To compensate for these variations, we follow[8] and adjust each modality independently. In a first step, we substract the mean of the brain region and normalize by its standard deviation. Afterward, we remove outliers by clipping and rescale the images to the range [0, 1].

2.1.7 *Postprocessing*

Although deep neural networks proved to produce segmentation results of high quality, postprocessing is a necessary step in a medical context. The BraTS contains HGG and low-grade gliomas (LGG). While the high-grade tumors typically consist of an enhancing tumor core, it is rarely present in low-grade abnormalities.

To compensate for this prior knowledge, we follow[8] and apply a postprocessing step to remove potentially false labels of the enhancing tumor core in LGG.

Our interactive segmentation approach[15] is a recent method with good results on kidney tumors[25] and already proved its ability to correct for false labels. Since it is less well-known, let us discuss it in more detail. We follow[26] and consider a minimal partitioning problem of the cubic image domain $\Omega \subset \mathbb{R}^3$ into $\Omega_1, \ldots, \Omega_n \subset \mathbb{R}^3$ non-overlapping regions

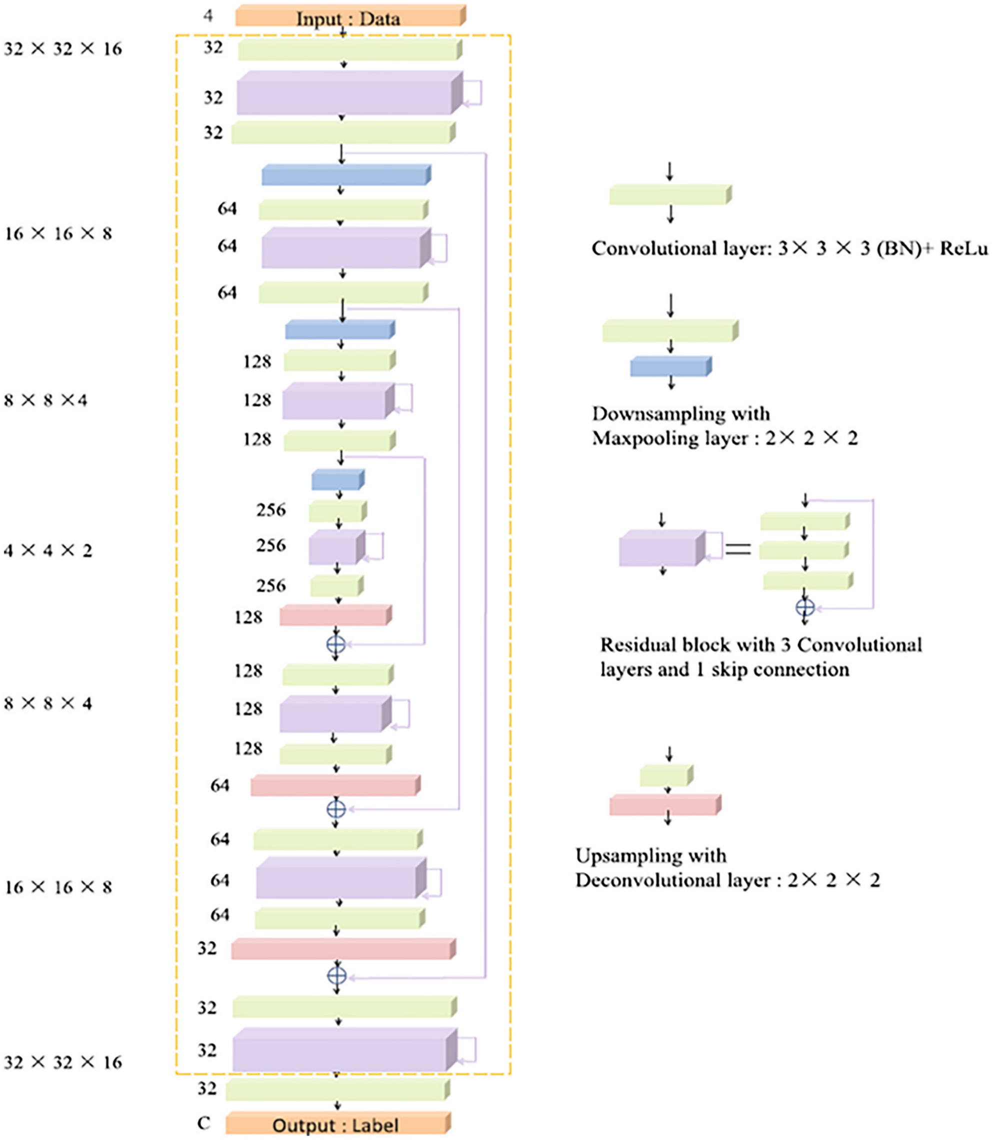

Fig. 5 Topology of FusionNets. Although the basic U-Net structure is still present, the architecture changed dramatically. Image courtesy of L. Vidyaratne et al.[23]

$$\min_{\Omega_1,\ldots,\Omega_n\subset\Omega} \frac{1}{2}\sum_{i=1}^{n} \mathrm{Per}(\Omega_i;\Omega) + \sum_{i=1}^{n} \int_{\Omega_i} h_i(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \quad \text{s.t. } \Omega=\bigcup_{i=1}^{n} \Omega_i, \quad \Omega_i\cap\Omega_j=\varnothing, \quad \forall\; i\neq j, \tag{2}$$

where $\mathrm{Per}(\Omega_i;\Omega)$ denotes the perimeter of region Ω_i inside Ω, and $h_i:\mathbb{R}\to\mathbb{R}_+$ are potential functions reflecting the cost for each pixel being assigned to a certain label $i=1,\ldots,n$. To align image and region boundaries, the perimeter is commonly measured in a metric induced by the underlying image $\boldsymbol{f}:\Omega\to\mathbb{R}^3$. In this application, we weight the perimeter $\mathrm{Per}_g(\Omega_i;\Omega)$ of region boundaries in the metric

$$g(\boldsymbol{x}) = \exp(-\mathcal{E}(\boldsymbol{x})^{\beta}/\overline{\mathcal{E}}), \qquad \overline{\mathcal{E}} \coloneqq \frac{2}{|\Omega|}\int_{\Omega} |\mathcal{E}(\boldsymbol{x})|\mathrm{d}\boldsymbol{x}.$$

Here, $\mathcal{E}:\Omega\to\mathbb{R}$ is the output of the fast structured edge detector of Refs. 27 and 28 and β is a positive parameter. Assume a (measurable) set of user-scribbles $\mathcal{S}_i\subset\Omega$ for each label i is given. We define the potential functions $h_i(\boldsymbol{x})$ in Eq. (2) as the negative logarithm of

$$\tilde{h}_i(\boldsymbol{x}) = \begin{cases} \left\{\frac{1}{|\mathcal{S}_i|}\int_{\mathcal{S}_i} G_\rho G_\sigma \mathrm{d}\boldsymbol{y}\right\}_{\text{scale}}, & \boldsymbol{x} \notin \mathcal{S}_j, \\ 1-\zeta, & \boldsymbol{x} \in \mathcal{S}_j, i = j, \\ \zeta/(n-1), & \boldsymbol{x} \in \mathcal{S}_j, i \neq j, \end{cases}, \tag{3}$$

and

$$G_\rho = k_{\rho_i(\boldsymbol{x})}(\boldsymbol{x}-\boldsymbol{y}), \quad G_\sigma = k_\sigma(\boldsymbol{f}(\boldsymbol{x})-\boldsymbol{f}(\boldsymbol{y})).$$

Here $\{.\}_{\text{scale}}$ denotes linear rescaling to [0,1], $|\mathcal{S}_i|$ is the area occupied by i'th label, ζ is the assumed probability for a scribble being correct, and k_σ and k_{ρ_i} are Gaussians with standard deviation σ in intensity space and adaptive standard deviation $\rho_i(\boldsymbol{x}) = \alpha \inf_{\boldsymbol{y}\in\mathcal{S}_i}|\boldsymbol{x}-\boldsymbol{y}|$ in the spatial domain, respectively. The spatially adaptive standard deviation attenuates the influence of the intensity distribution from scribbles that are far away proportionally to the distance of $\boldsymbol{x}$ to the closest scribble location. Hence, we postprocess the segmentation masks of the deep learning models as follows: In a first step, we sample every eight voxel in the output mask to sparsify the data. Afterward, we incorporate this mask in the cost term of our semi-automatic approach and densify the segmentation.

2.2 *Improving the Generalization Performance*

Overfitting is one of the major problems in training of deep neural networks. Typically, this issue is caused by a lack of training data in combination with complex models. Especially in the situation of medical image segmentation, the amount of data is rather limited. There are several approaches to relax this problem: Obviously, the most straight forward idea is to add more training data. However, this is typically a severe problem. Data augmentation is therefore often used to circumvent the lack of further data. In this approach, for example, additional data are simulated by random rotations, intensity shifts, axis mirror flips, or the addition of noise distributions. While almost all current methods use data augmentation, the simulation of different noise distributions is generally not used. In our opinion, this has two reasons: First, it is not possible to include every distortion that occurs. Although DNNs can handle the exact distortion they were trained on perfectly, they nevertheless show a strong generalization failure toward previously unseen variations.[29] However, the overfit to a specific dataset is reduced, so that although a better generalization can be achieved, the overall performance on the dataset drops slightly. Another possibility is to reduce the capacity of a model by reducing its size. Of course, it is also an option to regularize either the weights or the loss functions of a model. One more strategy is to include normalization layers: Recent work[22] indicates that normalization layers lead to wider optima and therefore better generalization.

In the following, we discuss two approaches: The first one, octave convolutions,[30] addresses the reduction of weights in a neural network while not reducing its capacity. This advanced operator allows to exploit the mixture of frequencies inherent to each image. Second, we illustrate the stochastic weight averaging[22] that enables the optimization algorithm to converge to wider and therefore better generalizing optima in the loss surface.

2.2.1 *Octave convolutions*

The fundamental aspect of convolution layers is their ability to identify local structures in their input data. These characteristics are then assigned to a new filter response, typically the image resolution does not change during this process.

However, each image can be divided into its low-frequency signal, which describes the coarse structure and the global layout, and its high-frequency signal, containing fine details (see Fig. 6). Although this is well known in the classic image processing community, this inherent information cannot be exploited by standard convolutional layers. Recently, there are several attempts to express this structure within layers of deep neural networks.[30,31] The multigrid approach of Ke et al.[31] maps every convolutional layer into a pyramid of operations. In this way, features at different scales can be extracted. However, this type of strategy obviously has

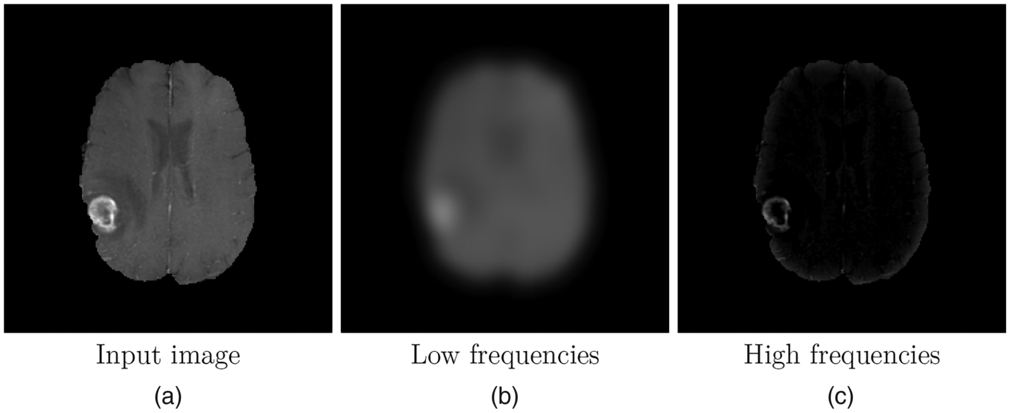

Fig. 6 Illustration of low- and high-frequency parts in an image. The input image of an MRI of the (a) brain is split into its (b) low frequencies and (c) high frequencies.

a massive disadvantage: The amount of required parameters increases with the number of scales in the pyramids.

Octave convolutions use a similar concept but interpret output feature maps as mixtures of information at different frequency scales.[30] Hence, these advanced convolutions factorize the output maps only into two groups: low and high frequencies. The corresponding smoothly changing low-frequency maps are then stored in a low-resolution tensor (half of the original input resolution) to reduce spatial redundancy;[30] (see Fig. 7).

Following this idea, octave convolutions process low-frequency information with corresponding (low frequency) convolutions. This not only increases the receptive field in the original pixel space but also collects more contextual information. Since the resolution for the low-frequency filter responses can be reduced, this saves both computational load and memory consumption.

The effort for such an octave convolution architecture consists of an additional hyperparameter $\alpha \in [0; 1]$ indicating the ratio of low-frequency components. To compute the output feature maps, the convolution kernel is split accordingly (see Fig. 8).

Obviously, filter responses of intra-frequency maps can be computed with regular convolutions. However, upsampling and downsampling (or pooling) operations for inter-frequency computations can also be folded up into the convolutions; see Ref. 30 for more details.

In total, the application of octave convolutions is straight-forward. Due to its inherent design, it is a plug-and-play component, not leading to any architectural changes. In some of our experiments, we replaced all standard convolutions with their octave variants. Although this change

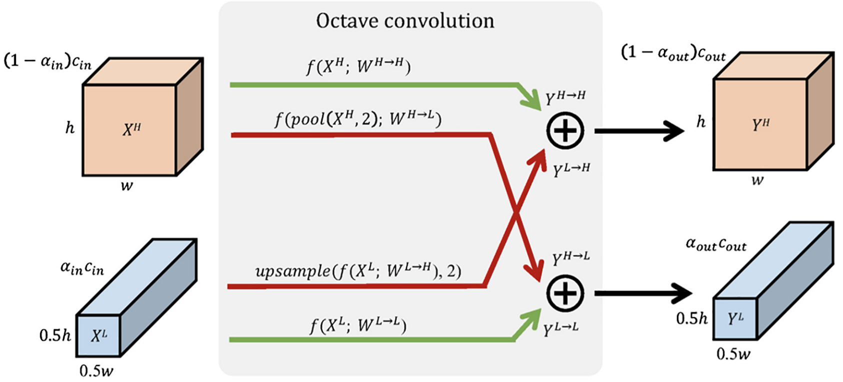

Fig. 7 Detailed design of octave convolutions. Red arrows indicate communication between low- and high-frequency components, green arrows depict regular information updates. Image courtesy of Chen et al.[30]

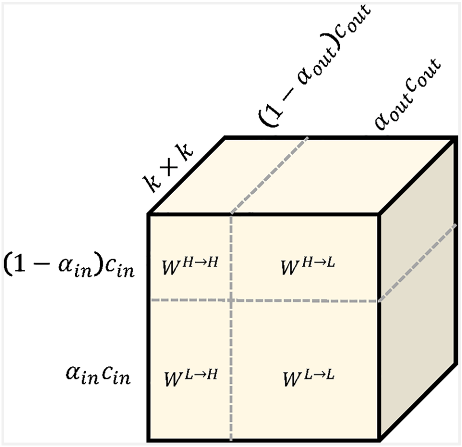

Fig. 8 Illustration of the octave convolution kernel. The kernel is split in intra- and inter-frequency parts. Image courtesy of Chen et al.[30]

had no consequences with respect to network architecture, it dramatically reduced the memory footprint of the models and training time per epoch while improving their generalization behavior (see Sec. 3).

2.2.2 *Stochastic weight averaging*

The training of deep neural networks is a tedious and time-consuming task. While in most cases, the capacity of the model architecture is large enough to solve the depicted problem, finding reasonable hyperparameters (e.g., learning rate and batch size) can be challenging: Especially, the learning rate has massive influence to the training procedure, and an optimal value is of crucial importance. In medical image segmentation, neural network architectures tend to be complicated and can easily overfit due to a limited amount of training data. In this scenario, an appropriate learning rate is even more important.

Typically, deep neural networks do not converge to a global minimum. Therefore, the quality of the model is evaluated with respect to its generalization performance. In general, local optima with flat basins tend to generalize better than those in sharp areas.[3,22,32] Since even small changes of the weights can lead to dramatic changes in the model prediction, these solutions are not stable. If the learning rate is too low, the model converges to the nearest local optimum and may hang in a sharp basin. Once the learning rate is high enough, the inherent random motion of the gradient steps not only prevents the solution from being trapped in one of the sharp regions but can also help the optimizer to escape. Obviously, finding a reasonable learning rate boils down to the trade-off between convergence and generalization.

Probably, the most common strategy to solve this problem is the usage of an cyclic scheme.[32,33] In cosine annealing, the learning rate cyclically decreases from a given maximal value following the cosine function.[33] It turned out that each of the local optima at the end of the cycles had similar performance, but lead to different but not overlapping errors in the model prediction (see Fig. 9). Hence, Huang et al.[3] suggested to combine the local optima of each cycle into an ensemble prediction. Unfortunately, computation time at inference increases dramatically with the number of snapshot models used in the ensemble.

Stochastic weight averaging follows the same idea but at a fraction of computational load. The basic idea is to conduct an equal average of the weights traversed by the optimizer with a learning rate schedule.[22] Intuitively, by taking the average of several local optima in the loss surface, a wider basin can be reached with better generalization performance.[22,34]

In contrast to ensemble approaches, we only need two models: The first one keeps track of the running average of the model weights, while the second one is traversing the weight space. At the end of each learning rate cycle, the state of the second model is used to update the weights of the running average model as

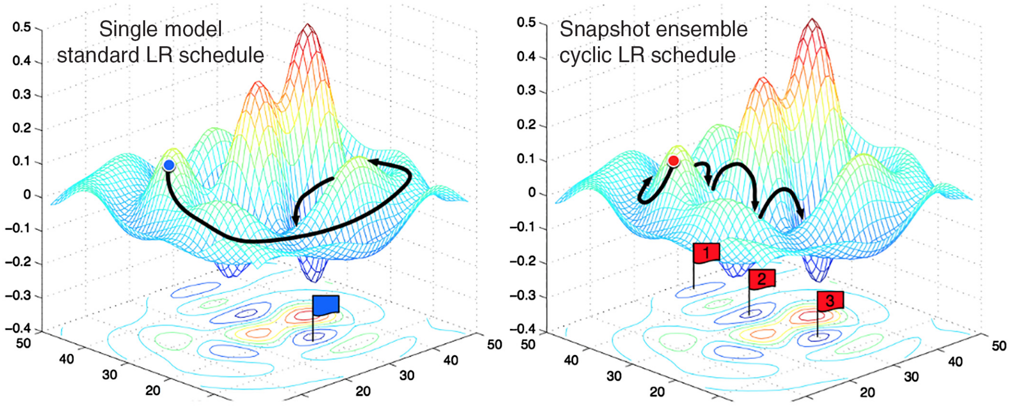

Fig. 9 Illustration of different model snapshots. While the standard learning rate schedule slowly converges to the minimum, snapshot ensembles are a combination of different local optima. Image courtesy of Huang et al.[3]

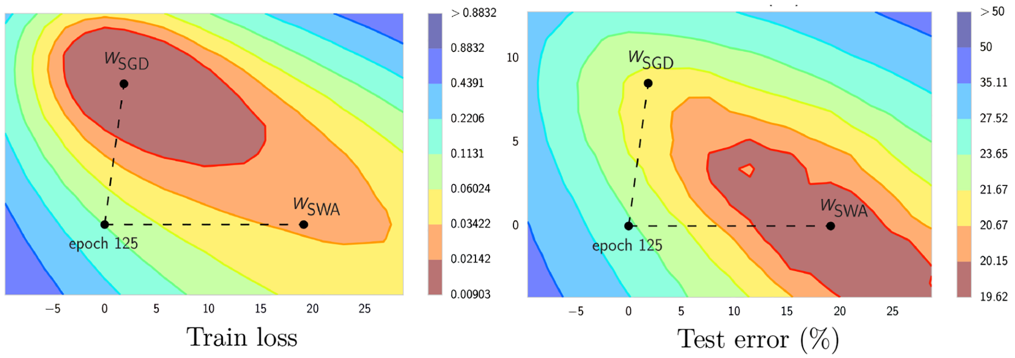

Fig. 10 Illustration of SWA and SGD showing the weights suggested by SGD and SWA at convergence. SWA started the weights of SGD after 125 training epochs. Image courtesy of Izmailov et al.[22]

$$w_{\text{swa}} = \frac{w_{\text{swa}} \cdot n_{\text{models}} + w}{n_{\text{models}} + 1}. \tag{4}$$

Here, w_{swa} are the weights of the running average model, while w are the weights of the model traversing the weight space, respectively. The total number of models to be averaged is given as n_{models}. All in all, stochastic weight averaging significally improves generalization performance,[34] being less prone to the shifts between train and test error loss (see Fig. 10). In general, the strategy can be divided in two phases: In the first phase of 75% of training time, the learning rate schedule follows a standard scheme, e.g., it is fixed to a specific value and decays after several epochs. In the second phase, the learning rate can be set to a constant value or follow a cyclic scheme to encourage the exploration of the loss surface (see Fig. 11).

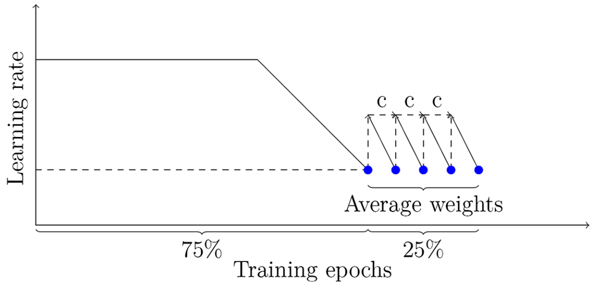

Fig. 11 Sketch of stochastic weight averaging.

3 Results

The BraTS is a widely accepted benchmark data set.[5,9,12] The challenge contains skull-stripped and spatially registered multimodal MR images (T_1, T_{1c}, T_2, and T_2-Flair) with a voxel size of 1 mm in every direction. Tumors are of different shape, size, and location in each data set.

In 2018, the BraTS challenge contained 285 training instances accompanied with 66 validation and 191 test cases. Unfortunately, the testing data set allows only for a single submission, disqualifying this compound for our analysis. However, we found the validation data set to be rather small and therefore not expressive. We decided to rely in our evaluation on five-fold cross validation on the training data set. We follow Isensee's assumption[8] that the conclusions drawn from the training set with cross validation are more general in nature and more robust to changes in the underlying distributions.

We performed nearly all network training on four NVidia Titan V with 12 GB memory and 5120 cuda cores. In case of NVDLMED, we do not have a graphics card with sufficiently large memory: We trained this network for several weeks on Intel Xeon Gold 6132 ("Skylake") with 28 CPU cores and 192GB of main memory.

We set all hyperparameters of the considered networks as described in their publications and used code provided by the authors whenever possible.

To generate a baseline for our experiments, we evaluated all analyzed approaches on the original BraTS2018 training data (see Table 1). Here, "CMS" denotes our cascadic Mumford–Shah method (Sec. 2.1.1) while "CascNN" means the cascadic segmentation approach with multiple neural networks (Sec. 2.1.5), and "No NewNet" refers to the No NewNet approach with region optimization and postprocessing (Sec. 2.1.3). It is noteworthy that "NVDLMED" represents a single NVDLMED network (Sec. 2.1.4) not an ensemble of several models.

The neural networks surpass the cascaded Mumford–Shah approach as expected. Nevertheless, the assumption that a brain tumor has higher average intensities in T_2-Flair images is a reliable prior knowledge: This intuitive method shows a remarkable performance when the entire tumor is considered. However, the most significant difference between the results of the various networks is shown in their accuracy to identify the enhancing tumor core.

Typically, a medical benchmark data set is intended as a biased version of a particular problem, i.e., in the case under consideration, all patients with high-grade brain tumors in MRI sequences. BraTS addresses this issue by providing comprehensive multi-institutional routine examinations of glioblastoma multiforme (HGG) and LGG with pathologically confirmed diagnosis.[5,12] However, care was mostly taken to create a representative visual representation of the brain tumors themselves. In a real clinical scenario, time and cost pressures usually prevail. For this reason, the assumption that voxels have a size of 1 mm in all directions is not realistic. In fact, exactly the opposite is typically true: While in-slice images are taken at high resolution, across-slice images are mostly sampled at lower resolution.

In addition, noise also plays an important role in MRI images. These recordings are very costly and time-consuming: Often, MR sequences differ dramatically in sampling rates and suffer from heavy noise disturbances. All in all, real clinical MR images do not correspond to the scheme of the BraTS benchmark data. To ensure the applicability of segmentation approaches

Table 1 BraTS18 evaluation for different segmentation approaches in terms of Dice score. No additional disturbances.

Method	Enhancing	Complete	Core
CMS	0.70	0.84	0.76
U-Net	0.73	0.89	0.82
CascNN	0.78	0.89	0.84
No NewNet	0.77	0.90	0.84
NVDLMED	0.82	0.91	0.86

Table 2 BraTS18 evaluation for different segmentation approaches. Gaussian noise ($\sigma = 0.02$) is added to the validation data.

Method	Enhancing	Complete	Core
CMS	0.69	0.82	0.74
U-Net	0.71	0.82	0.75
CascNN	0.65	0.76	0.76
No NewNet	0.72	0.83	0.79
NVDLMED	0.68	0.80	0.74

tested on BraTS data in everyday clinical practice, it is necessary for them to show high generalization performance.

For this reason, we analyze the outcomes of the different approaches when the distribution of the validation data set does not exactly match that of the training data. In a first step, we add Gaussian noise with zero mean and standard deviation $\sigma = 0.02$ to the validation data. The results are given in Table 2. The Dice scores indicate that the prior information about tumor appearance used in the cascadic Mumford–Shah approach is highly robust to disturbances. Although this approach performed worse than the considered neural networks in the original setting, it copes relatively well with the noisy data and the Dice score is only marginally reduced ($\approx$0.02 for all categories).

On the contrary, all of the tested neural networks have a major problem with the different distribution in the validation data. All of them show a significant decline in their segmentation performance. This problem obviously also becomes more serious the more complicated the respective architecture is. While the basic U-Net and the No NewNet model drop by a Dice score of $\approx$0.06 on average, the much more complex CascNN and NVDLMED show a significant decline by a Dice score of $\approx$0.11 and $\approx$0.12, respectively (see Fig. 12). This is consistent with our assumption that the best performing models are not the ones that generalize best on the test data, but only have the strongest overfit. This conclusion unfortunately disqualifies models trained and evaluated on BraTS data to be directly applied in a real clinical scenario.

The two approaches No NewNet and NVDLMED were almost equal in the evaluation of the BraTS18 challenge and our analysis in Table 1. Since the NVDLMED in particular shows a strong overfit on the data set while its training is extremely computationally intensive, we exclude this network in the following from our evaluation.

An obvious remedy to cope with noisy validation data is to add the same noise distribution to the training data as well (see Table 3). In fact, this additional information helps the three deep learning approaches to handle the altered distribution and their performance returns close to the original value. Certainly, it would be possible to add different noise distributions to the training data. However, at training time, it is usually not known how much noise is present in the test set. Another approach would be to include a preprocessing step to denoise the input images.

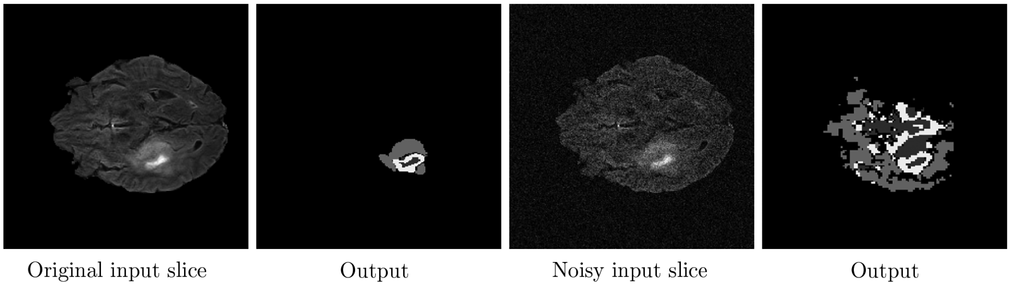

Fig. 12 Exemplary result for the No NewNet model on original, and slightly disturbed ($\sigma = 0.02$) input data.

Table 3 BraTS18 evaluation for different segmentation approaches. Gaussian noise ($\sigma = 0.02$) is added to training and validation data.

Method	Enhancing	Complete	Core
CMS	0.69	0.82	0.74
U-Net	0.72	0.87	0.81
CascNN	0.76	0.89	0.81
No NewNet	0.75	0.90	0.84

Unfortunately, this idea also has a massive disadvantage, small details might be lost. In our opinion, both approaches only lead to disguising the problem, but not to solving it. For this reason, we address the overfitting in the network topology itself.

In the following, we consider the No NewNet (without the adjustments suggested by the authors) as our baseline. Similar to our first experiments, we add Gaussian noise with zero mean and standard deviation $\alpha = 0.02$ and $\alpha = 0.04$ to our validation data (see Table 4). It turns out that the model in its simplest form performs similar to our cascadic Mumford–Shah method when not much noise is present in the data. However, as soon as the noise is seriously altering the data distribution, the model prediction collapses and is outperformed by the classical approach. Obviously, the generalization performance is limited and the network overfits the training data.

In a first step, we apply stochastic weight averaging (see Sec. 2.2.2) with a cycle length of 10 after 75% of the training epochs. This adaptation of the training cycle obviously has an massive influence on the generalization behavior. The averaging of multiple minima in the loss surface

Table 4 BraTS18 evaluation for different adaptations of No NewNet. Gaussian noise is added to the validation data.

Method	Enhancing	Complete	Core
Slight disturbance ($\sigma = 0.02$)			
CMS	0.69	0.82	0.74
Baseline	0.69	0.82	0.76
Baseline + SWA	0.73	0.86	0.81
Baseline + OctConv	0.71	0.84	0.77
Baseline + OctConv + SWA	0.74	0.88	0.83
Baseline + OctConv + SWA + post	0.78	0.89	0.84
Moderate disturbance ($\sigma = 0.04$)			
CMS	0.67	0.79	0.71
Baseline	0.66	0.72	0.70
Baseline + SWA	0.70	0.81	0.76
Baseline + OctConv	0.69	0.77	0.72
Baseline + OctConv + SWA	0.71	0.82	0.79
Baseline + OctConv + SWA + post	0.73	0.85	0.81

allows the model to cope well with the disturbed data while neither the model capacity nor the training time is increased: While the model improves in the first scenario by ≈0.04 on average, the performance gain of ≈0.06 in the second setting with heavier noise disturbances is massive.

Octave convolutions (see Sec. 2.2.1) have already shown in various applications that, in addition to a massive reduction in model size, they also contribute to improving generalization performance.[30] Consequently, we exchange all ordinary convolutions in the model by 3D octave convolutions ($\alpha = 0.75$). Although this minor change does not alter the network topology, in both settings the performance increases by ≈0.02 and 0.03 in Dice score over the baseline approach. This indicates a better generalization at inference to the validation data. The combination of stochastic weight averaging and the inclusion of frequency-aware octave convolutions leads to a improvement of ≈0.06 with slight and ≈0.08 moderate disturbances over the baseline.

The results of both of these modifications let us conclude that overfitting is indeed a serious problem, otherwise our changes would not lead to such drastic improvements.

Afterward, we use the sparsified results as input for our semi-automatic segmentation approach (see Sec. 2.1.7). In the first setting, this postprocessing step mainly corrects for false-positive labels of the enhancing tumor core (see Table 4). However, in the second scenario the robust energy formulation stabilizes the segmentation and increases the overall performance for all classes.

In the end, we evaluated our final model (No NewNet+OctConv+SWA+post) on the original BraTS data without additional noise. We did not observe any drop in its performance: With Dice scores of 0.79 for the enhancing tumor core, 0.90 for the whole tumor, and 0.85 for the tumor core our approach is on par with current state-of-the-art approaches.

It is noteworthy that the intention of this work is not to publish the next neural network trained on BraTS data. We rather want to highlight that generalization is a serious problem when improving on the benchmark metrics is the main goal. Definitely, one might argue that those networks are never meant to be directly applied in a clinical setting. We only partly agree with this opinion. First, BraTS was originally designed to allow for a fair comparison and especially to push research in the direction of brain tumor segmentation. In this context, neural networks that can only be applied to benchmark data sets counteract the goal of a medical image segmentation challenge. Second, networks with a high performance on these data sets should at least perform similar on real data, but in our experiments, all approaches except the No NewNet architecture showed a much lower performance than in the benchmark setting, and even dropped below our method that exclusively rely on reliable prior information. Third, we are deeply convinced that increasing complex models do not lead to a satisfying real-world performance. Similar to Isensee et al.,[8] we implemented several suggested network extensions and found them mostly pointless. Our experiments even indicate that they might be harmful as soon as training and validation data are not generated by the exactly same distribution. Hence, we fully agree that a well trained U-Net architecture is sufficient to solve this segmentation task.

All in all, we improved the generalization performance of the No NewNet architecture by straight forward adjustments in the model and the training procedure itself.

Although we neither changed its topology nor did we need to include the noise distribution in our training data, we could robustify the network while improving its generalization performance. Since this model is actually only a slightly modified version of the original U-Net, our suggested modifications also apply to similar structures.

4 Conclusions

With our paper, we have addressed the general problem of model overfitting of deep neural networks in brain tumor segmentation. Although the basic assumption to learn a class distribution from the training data is very powerful, it is also an Achilles heel when training and validation data slightly differ. In a first step, we added noise to the validation data. Unfortunately, our evaluations showed that such small variations lead to a massive drop in network performance for two of the three best-performing methods of BraTS 2018. Afterward, we analyzed the behavior of networks when training and validation data both are disturbed in the same way. It turned out that this additional information allows the network to cope with noisy data. However, since adding noise to the training data can have massive side-effects, we suggested several straightforward modifications to be included in network designs. Last but not least, we showed that these adjustments dramatically improve the generalization performance. Although we did not include the disturbance in the training data, we could reach with the same network topology nearly the same performance as without adding noise. This leads us to the conclusion that, in principle, all extensions of a well trained U-Net architecture for brain tumor segmentation not only fail to improve the result but also worsen the generalization performance.

In our ongoing research, we plan to investigate further simplifications of deep neural network models and the application of our findings to different benchmark data sets. Furthermore, we target the problem of architectural overfit of network topologies with a data-dependent design.

Disclosures

No conflicts of interest, financial, or otherwise, are declared by the authors.

Acknowledgments

N. Graf has received partly funding from the European Union's Seventh Framework Program for research, technological development, and demonstration [Grant agreement No 600841, Computational Horizons in Cancer (CHIC)]. The research of J. Weickert received funding from European Union's Horizon 2020 research and innovation program (Grant agreement No 741215, ERC Advanced Grant INCOVID). The authors state no conflict of interest and have nothing to disclose.

References

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, pp. 1097–1105 (2012).
2. J. Deng et al., "Imagenet: a large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, IEEE, pp. 248–255 (2009).
3. G. Huang et al., "Snapshot ensembles: train 1, get M for free," in *Proc. Int. Conf. Learn. Represent.*, Toulon, France (2017).
4. Ö. Çiçek et al., "3D U-Net: learning dense volumetric segmentation from sparse annotation," *Lect. Notes Comput. Sci.* **9901**, 424–432 (2016).
5. S. Bakas et al., "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BraTS challenge," arXiv:1811.02629 (2018).
6. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," *Lect. Notes Comput. Sci.* **9351**, 234–241 (2015).
7. A. Myronenko, "3D MRI brain tumor segmentation using autoencoder regularization," *Lect. Notes Comput. Sci.* **11384**, 311–320 (2018).
8. F. Isensee et al., "No New-Net," *Lect. Notes Comput. Sci.* **11384**, 234–244 (2018).
9. B. Menze et al., "The multimodal brain tumor image segmentation benchmark (BraTS)," *IEEE Trans. Med. Imaging* **34**(10), 1993–2024 (2015).
10. J. C. Marsh et al., "Current status of immunotherapy and gene therapy for high-grade gliomas," *Cancer Control* **20**, 43–48 (2013).
11. P. Y. Wen et al., "Updated response assessment criteria for high-grade gliomas: response assessment in neuro-oncology working group," *J. Clin. Oncol.* **28**, 1963–1972 (2010).
12. S. Bakas et al., "Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features," *Sci. Data* **4**, 170117 (2017).
13. G. Urban et al., "Multi-modal brain tumor segmentation using deep convolutional neural networks," in *MICCAI BraTS Challenge Proc.*, pp. 31–35 (2014).
14. S.-C. Chow et al., *Sample size calculations in clinical research*, CRC press, Boca Raton, Florida (2017).

15. S. Müller et al., "Robust interactive multi-label segmentation with an advanced edge detector," *Lect. Notes Comput. Sci.* **9796**, 117–128 (2016).
16. A. Myronenko, "3D MRI brain tumor segmentation using autoencoder regularization," in *Int. l MICCAI Brainlesion Workshop*, Springer, Cham, Switzerland, pp. 311–320 (2018).
17. C. Zhou et al., "Learning contextual and attentive information for brain tumor segmentation," *Lect. Notes Comput. Sci.* **11384**, 497–507 (2018).
18. S. Müller, J. Weickert, and N. Graf, "Automatic brain tumor segmentation with a fast Mumford–Shah algorithm," *Proc. SPIE* **9784**, 97842S (2016).
19. D. Mumford and J. Shah, "Boundary detection by minimizing functionals, I," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, IEEE Computer Society Press, San Francisco, CA, pp. 22–26 (1985).
20. D. Mumford and J. Shah, "Optimal approximation of piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.* **42**(5), 577–685 (1989).
21. S. Jégou et al., "The one hundred layers tiramisu: fully convolutional densenets for semantic segmentation," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. Workshops*, IEEE, Honolulu, HI, pp. 11–19 (2017).
22. P. Izmailov et al., "Averaging weights leads to wider optima and better generalization," in *Proc. Conf. Uncertainty Artif. Intell.*, Monterey, CA (2018).
23. L. Vidyaratne et al., "Deep learning and texture-based semantic label fusion for brain tumor segmentation," *Proc. SPIE* **10575**, 105750D (2018).
24. C. Zhou et al., "One-pass multi-task convolutional neural networks for efficient brain tumor segmentation," *Lect. Notes Comput. Sci.* **11072**, 637–645 (2018).
25. S. Müller et al., "Benchmarking Wilms' tumor in multisequence MRI data: why does current clinical practice fail? Which popular segmentation algorithms perform well?" *J. Med. Imaging* **6**(3), 034001 (2019).
26. A. Chambolle, D. Cremers, and T. Pock, "A convex approach to minimal partitions," *SIAM J. Appl. Math.* **5**(4), 1113–1158 (2012).
27. P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proc. IEEE Int. Conf. Comput. Vision*, Washington, DC, pp. 1841–1848 (2013).
28. P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(8), 1558–1570 (2015).
29. R. Geirhos et al., "Generalisation in humans and deep neural networks," in *Adv. Neural Inf. Process. Syst.*, pp. 7538–7550 (2018).
30. Y. Chen et al., "Drop an octave: reducing spatial redundancy in convolutional neural networks with octave convolution," in *IEEE/CVF Int. Conf. Comput. Vision* (2019).
31. T.-W. Ke, M. Maire, and S. X. Yu, "Multigrid neural architectures," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 6665–6673 (2017).
32. L. N. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Conf. Appl. Comput. Vision*, IEEE, pp. 464–472 (2017).
33. I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," in *Proc. Int. Conf. Learn. Represent.*, Toulon, France (2017).
34. B. Athiwaratkun et al., "There are many consistent explanations of unlabeled data: why you should average," in *Proc. Int. Conf. Learn. Represent.*, New Orleans, Louisiana (2019).

Sabine Müller received her BSc degree in computer science from Saarland University, Saarbrücken, Germany, in 2011, and her MSc degree in visual computing in 2014. Currently, she is a PhD candidate in the Mathematical Image Analysis Group at Saarland University, and research assistant at the competence center for high-performance computing at the Fraunhofer institute for industrial mathematics. Her research interests are in the areas of medical image segmentation, computer vision, and pattern recognition and efficient and distributed training of deep neural networks.

Joachim Weickert is a professor of mathematics and computer science at Saarland University (Saarbrücken, Germany), where he heads the Mathematical Image Analysis Group. He graduated and obtained his PhD from the University of Kaiserslautern (Germany) in 1991 and 1996. He worked as post-doctoral researcher at the University Hospital of Utrecht (The Netherlands) and the University of Copenhagen (Denmark), and as assistant professor at the University of Mannheim (Germany). He has developed many models and efficient algorithms for image processing and computer vision using partial differential equations and variational methods. He is editor-in-chief of the *Journal of Mathematical Imaging and Vision*.

Norbert Graf is a professor of pediatrics and director of pediatric oncology and hematology at Saarland University. He is the chairman of the Renal Tumour Study Group of the International Society of Paediatric Oncology, an associate member of the Children's Oncology Group of North America, an external reviewer for the Japan Science and Technology Agency, a member of the board of the VPH-Institute and has more than 25 years of experience in running clinical trials.