Low-Rank Convolutional Networks for Brain Tumor Segmentation

Pooya Ashtari[1](✉), Frederik Maes[2,3], and Sabine Van Huffel[1]

[1] Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, Belgium
pooya.ashtari@esat.kuleuven.be

[2] Department of Electrical Engineering (ESAT), Processing Speech and Images (PSI), KU Leuven, Leuven, Belgium

[3] Medical Imaging Research Center, UZ Leuven, Leuven, Belgium

Abstract. The automated segmentation of brain tumors is crucial for various clinical purposes from diagnosis to treatment planning to follow-up evaluations. The vast majority of effective models for tumor segmentation are based on convolutional neural networks with millions of parameters being trained. Such complex models can be highly prone to overfitting especially in cases where the amount of training data is insufficient. In this work, we devise a 3D U-Net-style architecture with residual blocks, in which low-rank constraints are imposed on weights of the convolutional layers in order to reduce overfitting. Within the same architecture, this helps to design networks with several times fewer parameters. We investigate the effectiveness of the proposed technique on the BraTS 2020 challenge.

Keywords: Low-rank Representation · U-Net · Glioma Segmentation.

1 Introduction

Gliomas are brain tumors with the highest mortality rate and prevalence. They can be classified into two grades: low-grade glioma (LGG) and high-grade glioma (HGG), with the former being less aggressive than the latter. Multi-modal MRI is widely used to diagnose and assess gliomas in clinical practice. The accurate segmentation of gliomas is crucial for various clinical purposes, including diagnosis, treatment planning, image-guided surgery, and follow-up evaluations. However, manual delineation of tumors is laborious, time-consuming, and expensive especially because experts need to deal with 3D images and several modalities; therefore, accurate computer-assisted methods are needed to automatically perform this task. Despite considerable advances in medical imaging, glioma segmentation is still a challenging task since tumors can vary dramatically in shape, structure, and location across patients and over time within a specific patient. Moreover, the growing tumor mass may displace and deform the surrounding normal brain tissues, as do resection cavities that are present after surgery.

The brain tumor segmentation challenge (BraTS) [1–4, 19] aims to develop effective data-driven algorithms for brain tumor segmentation by providing a large dataset of annotated

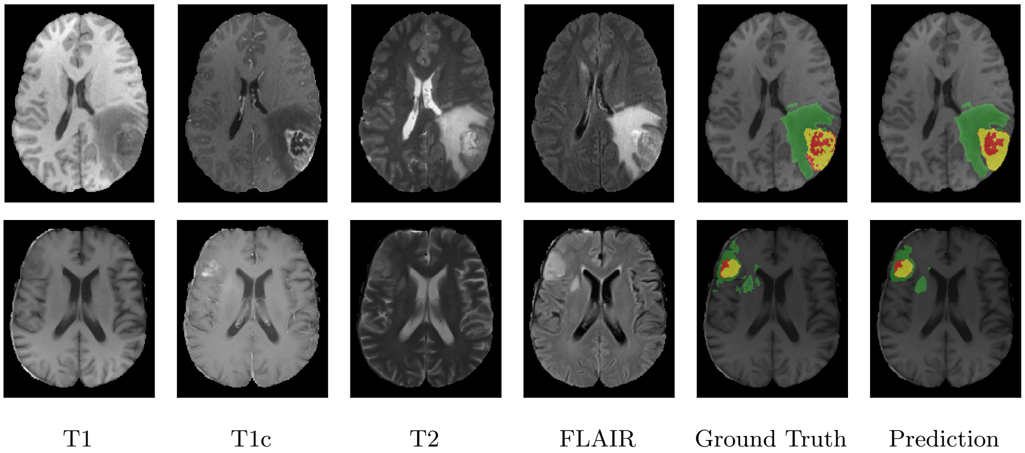


Fig. 1 The multimodal MR images along with the corresponding ground truth and prediction for a HGG (top row) and a LGG (bottom row) representative cases (green: edema, yellow: enhancing tumor, red: necrosis and non-enhancing tumor).

LGG and HGG 3D MRI scans, each with four MRI modalities (T1 weighted, post-contrast T1-weighted, T2-weighted, and FLAIR) rigidly co-registered, resampled to the voxel size 1mm^3, and skull-stripped. The BraTS 2020 training dataset consists of 369 cases, each of which is manually annotated by up to 4 raters who label each voxel as enhancing tumor (ET), edema (ED), necrotic and non-enhancing tumor (NCR/NET), or everything else (see Figure 1). However, for evaluation, the 3 nested subregions, namely whole tumor (WT), tumor core (TC–i.e., the union of ED and NCR/NET), and enhancing tumor (ET), are used.

Recently, deep learning models, particularly convolutional neural networks (CNN), surpassed traditional computer vision methods for semantic segmentation. In contrast to the conventional approach based on hand-crafted features, CNNs are able to automatically learn high-level features adapted specifically to the task of brain tumor segmentation. Currently, the vast majority of effective CNNs for medical image segmentation are based on a U-Net [23] architecture with millions of trainable parameters. However, such complex models can be highly prone to overfitting especially in cases where the amount of training data is insufficient, which is usually the case for medical imaging. In this work, we introduce a new layer, called low-rank convolution, in which low-rank constraints are imposed to regularize weights and thus reduce overfitting. We make use of a 3D U-Net [5] architecture with residual modules [10] and further improve it by replacing ordinary convolution layers with low-rank ones, achieving models with several times fewer parameters than the initial ones. This leads to significantly better performance especially because the amount of training data is limited.

The rest of this paper is organized as follows: Section 2 briefly reviews relevant semantic segmentation techniques. Section 3 presents our approach to brain tumor segmentation using low-rank convolutional layer. Experiments are presented in Section 4 . We conclude this paper in Section 5 .

2 Related work

Over the past few years, considerable research efforts have been directed to the development of fully convolutional neural networks for semantic segmentation. Encoder-decoder architectures and their variants, in particular U-Net [23], are probably the most successful ones in the segmentation of medical images.

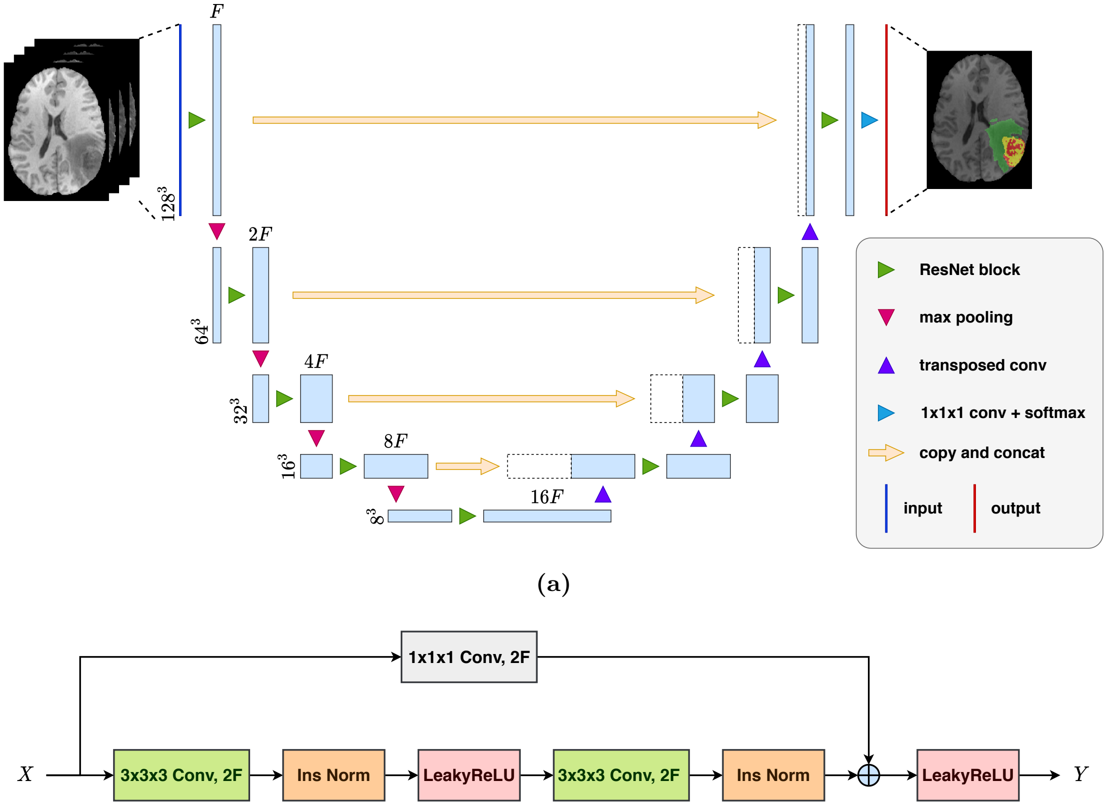
In BraTS 2018 and 2019, all top solutions made use of such architectures in their models one way or another. Isensee et al. [13] focused on the training procedure instead of proposing a new network, winning second place in BraTS 2018 by making only minor modifications to the standard 3D U-Net [5], using additional training data, and applying a simple post-processing technique. McKinley et al. [18] proposed an architecture, in which dense blocks [12] of dilated convolutions are embedded in a shallow U-Net-style network. Following an encoder-decoder CNN architecture, Myronenko [21] won first place in BraTS 2018 by adding a branch to the encoder endpoint and taking a variational auto-encoder (VAE) approach. The winning model [15] in BraTS 2019, was based on a similar architecture but further employed a two-stage cascaded strategy.

CNNs are computationally demanding and memory-intensive. Since convolution operations comprise the vast bulk of computations in a deep CNN during both training and inference, methods have been proposed to speed up and compress convolutional layers. MobileNet [11] exploits depthwise separable filters to represent a standard convolution layer more compactly, leading to a substantial reduction in computational complexity at the cost of a small loss of accuracy. Inception [25] uses bottleneck architectures made of cheap 1x1 convolutions to limit the network size. These methods suggest new architectures by factorizing a convolution into smaller blocks.

An alternative approach is based on low-rank approximations [14] and tensor decompositions [16,17], where the weights in a convolution layer are constrained to be low-rank. One advantage of this approach is that for a fixed architecture we can easily control the number of parameters and the computational complexity of the model by adjusting the rank. Furthermore, imposing low-rank constraints can regularize the model and reduce overfitting. In addition to speed-up, Tai et al. [26] achieved significant improvements in some cases using CNNs with low-rank regularization. Note that although all the mentioned techniques are only applied to the task of image classification, they can be also deployed effectively in an encoder-decoder architecture for image segmentation. The impact of low-rank regularization on the performance is expected to be greater when less data is available for training.

3 Method

In this section, we present our approach to brain tumor segmentation. The baseline architecture used is based on a 3D U-Net (Figure 2a) proposed in [5] and customized for BraTS 2018 by [13], except that here convolutional blocks at each level are replaced by ResNet blocks [10]. We also introduce a new layer, called low-rank convolution, as a regularization technique to



(a)

(b)

Fig. 2 The U-Net architecture (a) and the ResNet block (b) used for brain tumor segmentation.

reduce overfitting. By replacing ordinary convolution layers with low-rank ones, we can achieve significantly better performance especially when the amount of training data is insufficient, where the model is more prone to overfitting. In the following, we describe the training procedure and the building blocks of our networks.

3.1 Data Preprocessing and Augmentation

The BraTS data is heterogeneous in the sense that it is multiparametric and acquired with different protocols at multiple institutions using various scanners, making intensity values non-standardized. There is also high between-subject variability in tumors due to the presence of both low- and high-grade gliomas. To alleviate this heterogeneity and insufficiency of data, it is crucial to perform an effective preprocessing workflow before feeding the data into the network.

For each scan, we first form a 4-channel 3D image as the input, where each channel corresponds to one of the modalities (i.e. T1, post-contrast T1, T2, and FLAIR). We crop each image with a minimal box containing the whole brain region then resize it to the size $128 \times 128 \times 128$. Each channel of each image is then normalized independently using z-score to have intensities with zero mean and unit variance. Three data augmentation techniques are also utilized to reduce overfitting. Firstly, the input image is randomly flipped along the left-right axis. Secondly, we apply a random affine transform (scale $\sim U(0.9, 1.1)$, rotation $\sim U(-10, 10)$). Finally, a Gaussian noise ($\mu = 0, \sigma \sim U(0, 0.25)$) is added to intensities per-channel.

3.2 Network Architecture

As mentioned before, our network, as shown in Figure 2a, follows a U-Net-like architecture made up of encoder and decoder parts. The network takes a 4-channel image of size 128 × 128 × 128 and outputs a *probability map* with the same spatial size and with 4 channels that correspond to the 4 segmentation labels. The network has 4 levels, at each of which in the encoder (decoder) part, the input tensor is downscaled (upscaled) by a factor of two while the number of channels is doubled (halved). Downscaling and upscaling are performed via max-pooling and transposed convolution, respectively. In both the encoder and decoder, we use ResNet blocks [10], where each block is composed of convolution, Instance Normalization [27], and LeakyReLU activation layers (Figure 2b). Two $3 \times 3 \times 3$ convolutions are used in the residual mapping of each ResNet block, and a $1 \times 1 \times 1$ convolution is used in the shortcut connection in order to match the number of input channels with the number of output channels of the residual mapping. At the decoder endpoint, a $1 \times 1 \times 1$ convolution followed by a *softmax layer* is applied to get the segmentation probability map.

3.3 Low-rank Convolution

Convolutions form the backbone of a CNN. A typical U-Net has millions of training parameters, the majority of which are the weights that correspond to convolutional layers (this is true for any CNN with no fully-connected layers). In practice, such a complex model is very likely to overfit in particular when it comes to medical image segmentation applications, where the amount of annotated data is typically limited. Common regularization techniques like *dropout* [24] and *weight decay* can be used to mitigate this problem. However, these methods do not decrease the total number of parameters while modern CNNs are known to be heavily over-parameterized [8], i.e., the number of parameters exceeds the size of training data and what is theoretically sufficient. In this work, by imposing low-rank constraints on weights, we propose a new operation, termed Low-Rank Convolution (LRCONV), enabling the design of deep architectures with much fewer parameters but more robustness to overfitting. It is noteworthy that this idea is unrelated but complementary to other regularization techniques, such as dropout and weight decay.

Let tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{C_{\text{in}} \times H \times W \times D}$ be the input of a 3D convolutional layer (for simplicity, we assume unit stride and dilation, and a zero-padded input) with the kernel tensor $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times H' \times W' \times D'}$ and bias $\mathbf{b} \in \mathbb{R}^{C_{\text{out}}}$. The output tensor $\tilde{\boldsymbol{\mathcal{X}}} \in \mathbb{R}^{C_{\text{out}} \times H \times W \times D}$ is obtained as follows:

$$\tilde{x}_{chwd} = b_c + \sum_{c'=1}^{C_{\text{in}}} \sum_{h'=1}^{H'} \sum_{w'=1}^{W'} \sum_{d'=1}^{D'} x_{c'(h+h')(w'+w)(d+d')} v_{cc'h'w'd'} \tag{1}$$

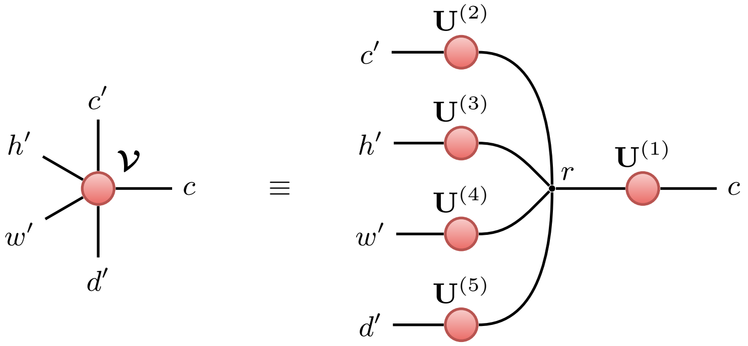


Fig. 3 Canonical Polyadic tensor network. The 5th-order weight tensor (left panel) in a convolutional layer is represented with a Canonical Polyadic tensor network (right panel). In the graphical notation of an Einstein summation, nodes and edges denote the tensors and their corresponding indices, respectively.

where (H, W, D) is the resolution of the input image; C_{in} and C_{out} denote the number of channels in the input and output, respectively; and (H', W', D') is the size of the convolution kernel. To define a low-rank convolution, we can use a Canonical Polyadic [6] form and re-parameterize the weight as a sum of rank-1 tensors:

$$\boldsymbol{\mathcal{V}} = \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \mathbf{u}_r^{(3)} \circ \mathbf{u}_r^{(4)} \circ \mathbf{u}_r^{(5)}, \tag{2}$$

where "$\circ$" denotes the vector *outer product*; $\mathbf{u}_r^{(1)} \in \mathbb{R}^{C_{\text{out}}}$, $\mathbf{u}_r^{(2)} \in \mathbb{R}^{C_{\text{in}}}$, $\mathbf{u}_r^{(3)} \in \mathbb{R}^{H'}$, $\mathbf{u}_r^{(4)} \in \mathbb{R}^{W'}$, and $\mathbf{u}_r^{(5)} \in \mathbb{R}^{D'}$; and R is the rank. Equivalently, the above equation can be re-written elementwise as:

$$v_{cc'h'w'd'} = \sum_{r=1}^{R} u_{cr}^{(1)} u_{c'r}^{(2)} u_{h'r}^{(3)} u_{w'r}^{(4)} u_{d'r}^{(5)}, \tag{3}$$

where $\mathbf{U}^{(j)}$ is a *factor matrix* whose columns are $\{\mathbf{u}_1^{(j)}, \ldots, \mathbf{u}_R^{(j)}\}$. Since the equation (3) can be treated as an Einstein summation, we can illustrate it using a *tensor network* diagram [6] as shown in Figure 3. In this paper, we only use the Canonical Polyadic form, but other sparsely connected tensor network like Tucker and *tensor train* can be also utilized. In a LRCONV layer, the factor matrices and the bias are the parameters to be learned. The rank R is a hyperparameter by which we can control the number of parameters although, throughout the rest of this paper, we tune the ratio $\alpha = R/C_{\text{in}}$ rather than R for controlling the layer complexity. Obviously, the smaller α, the fewer parameters the layer has.

It is worth noting that our approach exploits low-rank representations to regularize a CNN before the training in contrast to methods compressing pre-trained CNNs using low-rank approximations [14] and tensor decompositions [17].

3.4 Loss Function

The loss function used to train the network is the *soft Dice loss* [20], defined as

$$\mathbf{L}_{\text{Dice}} = 1 - \frac{2\langle P, G\rangle + 1}{||P||^2 + ||G||^2 + 1} \tag{4}$$

where $\langle\cdot,\cdot\rangle$ denotes the *dot product* of tensors; $||.||$ denotes the Frobenius norm of a tensor; P is the predicted probability segmentation map (the out of the softmax layer); and G is the *one-hot binary mask* encoding the corresponding ground truth. Both P and G are $4 \times 128 \times 128 \times 128$ tensors, where the 4 channels correspond to the 4 segmentation labels. We add 1 to both the numerator and denominator (sometimes known as *additive smoothing*) to smooth the loss and avoid division by zero. Although we optimize the Dice loss obtained by the labels, e.g. enhancing tumor, edema, necrosis and non-enhancing, we also monitor the Dice for the three overlapping regions, i.e., whole tumor, tumor core, and enhancing tumor.

3.5 Optimization

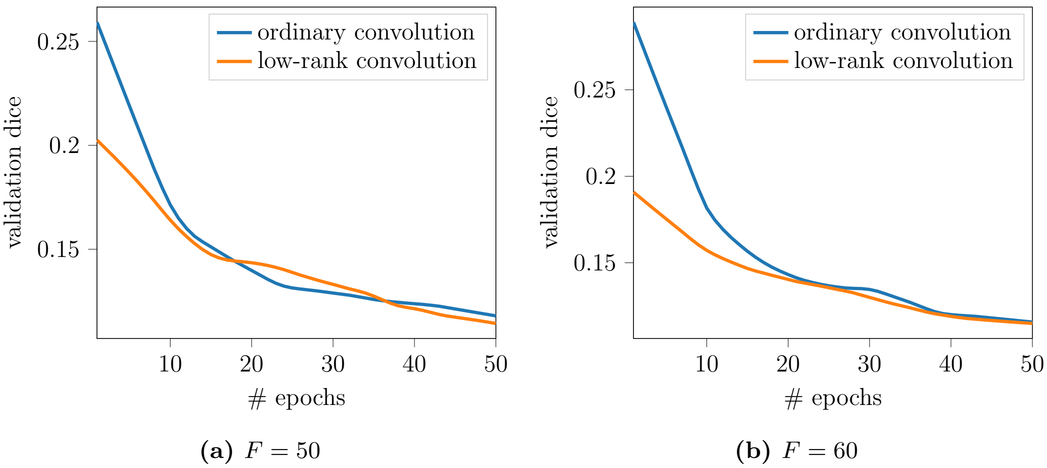
All networks are trained for 50 epochs, with a batch size of one. We use Adam optimizer with initial learning rate of 10^{-4} and regularize models with ℓ_2 weight decay of 10^{-5}. The learning rate is scheduled to decrease by a factor of 5 if the validation metric sees no improvement within 5 epochs. The training set is randomly split into 80% (298 cases) used for training and the rest 20% (73 cases) used for validation.

3.6 Postprocessing

For a given test image in the inference phase, the probability map from the network output is resized to its original size. The map is then padded to have the same size as it had before cropping in the preprocessing step. We need to process the resulting probability map to obtain the final binary segmentation mask. The most trivial way is to select the label with the highest probability for each voxel. However, this does not exploit the fact that the tumor subregions, i.e., necrosis, tumor core, and whole tumor, are nested within each other. To overcome this shortcoming, we perform a hierarchical scheme, where the whole tumor is first extracted by thresholding the probability map. Having restricted to the voxels of the whole tumor, the edema channel of the probability map is then thresholded to extract the tumor core. The final threshold is applied to separate the necrosis and enhanced tumor within the tumor core. As shown in the next section, this postprocessing improves the performance significantly, with the Hausdorff distance of enhancing tumor decreasing by $\approx 15\%$.

4 Experiments and Results

All the models were implemented using PyTorch [22] and PyTorch Lighting [9] frameworks and trained on a NVIDIA P100 SXM2 GPU. We experimented with different values for the initial



(a) $F = 50$

(b) $F = 60$

Fig. 4 Validation metric plotted against epochs for two different values of F (the initial number of feature maps). The low-rank convolutions with $\alpha = 0.5$ are used. For visualization purposes, the curves are smooth via LOESS [7].

number of feature maps (denoted by F in Figure 2) and found the larger values to perform better although the GPU memory limitation did not allow us to try values greater than 60. We noticed that low-rank convolution can improve the results particularly for larger networks, roughly those with $F > 40$. We obtain the low-rank version of a U-Net by replacing $3 \times 3 \times 3$ ordinary convolutions with low-rank ones in all except the first level (the number of parameters at the first level is already relatively small). As seen in Figure 4, the networks with low-rank convolutions converge faster, attributed to the fact that the number of training parameters of the initial networks is dramatically reduced (see Table 1), resulting in much less complex models and optimization problems. The impact of low-rank regularization on convergence and performance is more substantial for the case $F = 60$ (with 92 million parameters) compared to the case $F = 50$ (with 64 million parameters), which is somewhat expected since the former has far more parameters.

The results on the BraTS 2020 validation set (125 cases) obtained by the BraTS online evaluation framework are reported in Table 1. Our base model follows the Res-U-Net architecture (with $F = 60$ initial feature maps) described in Section 3.2. By postprocessing the probability maps using the hierarchical scheme described in Section 3.6, we achieved an improvement, particularly on the validation Dice of enhancing tumor. The results were further improved by training the low-rank version (with $\alpha = 0.5$) of the network, with the Hausdorff distance of enhancing tumor falling from 32.22 to 25.01. The low-rank network with 20.8 million parameters is far more memory efficient than the initial network with 92.2 million parameters. Finally, an ensemble of 8 models, including networks with $F \in \{30, 40, 50, 60\}$ and their low-rank versions (with $\alpha = 0.5$), was used to reduce the variance in predictions. To build the ensemble model, the probability maps were first averaged and then thresholded via the hierarchical scheme. This further improved all the scores, leading to the Dice score of over 90% for the whole tumor.

The greatest improvement was observed for the tumor core, with the Dice score increasing from 77.84% to 79.1% and the Hausdorff distance dropping from 16.96 mm to 7.76 mm. Figure 1 shows the results of this model on representative HGG and LGG cases.

Table 1 The average scores on the BraTS 2020 validation set (125 cases). They are computed by the online evaluation platform (WT: whole tumor, TC: tumor core, and ET: enhancing tumor).

Method	# Params (M)	Dice (%)			Hausdorff		
		ET	WT	TC	ET	WT	TC
Res-U-Net	92.2	71.99	89.68	78.79	38.18	5.74	14.49
Res-U-Net + post	92.2	73.32	89.84	78.77	32.22	5.75	14.46
Low-rank Res-U-Net + post	20.8	74.82	89.12	77.84	25.01	7.61	16.94
Ensemble of 8 models	—	75.06	90.37	79.1	25.61	5.23	7.76

Table 2 presents the performance of the method on the BraTS 2020 test set. The average Dice scores for enhancing tumor core, whole tumor, and tumor core are 77.73%, 87.37%, and 81.24%, respectively. The corresponding values of Hausdorff distance are 16.21 mm, 6.28 mm, and 20.52 mm, respectively. Overall, the test results are consistent with those of the validation set, which reflects our model is neither biased towards the validation set nor has a high variance.

Table 2 Summary statistics of the scores on the BraTS 2020 test set (166 cases). They are computed by the online evaluation platform (WT: whole tumor, TC: tumor core, and ET: enhancing tumor).

	Dice			Hausdorff		
	ET	WT	TC	ET	WT	TC
Mean	77.73	87.37	81.24	16.21	6.28	20.52
StdDev	21.6	13.93	25.45	69.39	11.76	74.75
Median	83.97	91.58	91.05	2.0	3.25	2.83
25th quantile	75.08	86.60	82.82	1.41	2.0	1.49
75th quantile	89.59	94.54	94.92	3.0	5.72	5.34

5 Conclusion

This paper proposes a regularization technique for CNNs by re-parameterizing convolutional layers as a low-rank structure, particularly canonical polyadic form. We devised a U-Net architecture with ResNet blocks consisting of low-rank convolutions. We examined the impact of this low-rank regularization on performance, verifying its effectiveness for brain tumor segmentation in multimodal MRI scans. The results on the BraTS 2020 data show that despite having much fewer parameters, the low-rank networks can outperform the unregularized versions especially in terms of Dice coefficients and Hausdorff distances on the enhancing tumor.

Acknowledgements. The research leading to these results has received funding from EU H2020 MSCA-ITN-2018: INtegrating Magnetic Resonance SPectroscopy and Multimodal Imaging for Research and Education in MEDicine (INSPiRE-MED), funded by the European Commission under Grant Agreement #813120. This research also received funding from the Flemish Government (AI Research Program). Sabine Van Huffel and Pooya Ashtari are affiliated to Leuven.AI - KU Leuven institute for AI, B-3000, Leuven, Belgium.

References

1. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J., Farahani, K., Davatzikos, C.: Segmentation labels and radiomic features for the pre-operative scans of the tcga-gbm collection. the cancer imaging archive. Nat Sci Data **4**, 170117 (2017)
2. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J., Farahani, K., Davatzikos, C.: Segmentation labels and radiomic features for the pre-operative scans of the tcga-lgg collection. The cancer imaging archive **286** (2017)
3. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.S., Freymann, J.B., Farahani, K., Davatzikos, C.: Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. Scientific data **4**, 170117 (2017)
4. Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R.T., Berger, C., Ha, S.M., Rozycki, M., et al.: Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. arXiv preprint arXiv:1811.02629 (2018)
5. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: International conference on medical image computing and computer-assisted intervention. pp. 424–432. Springer (2016)
6. Cichocki, A., Lee, N., Oseledets, I., Phan, A.H., Zhao, Q., Mandic, D.P.: Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. Foundations and Trends® in Machine Learning **9**(4-5), 249–429 (2016)
7. Cleveland, W.S., Devlin, S.J.: Locally weighted regression: an approach to regression analysis by local fitting. Journal of the American statistical association **83**(403), 596–610 (1988)
8. Denil, M., Shakibi, B., Dinh, L., Ranzato, M., De Freitas, N.: Predicting parameters in deep learning. In: Advances in neural information processing systems. pp. 2148–2156 (2013)
9. Falcon, W.: Pytorch lightning. GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning Cited by **3** (2019)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
12. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)

13. Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., Maier-Hein, K.H.: No new-net. In: International MICCAI Brainlesion Workshop. pp. 234–244. Springer (2018)
14. Jaderberg, M., Vedaldi, A., Zisserman, A.: Speeding up convolutional neural networks with low rank expansions. arXiv preprint arXiv:1405.3866 (2014)
15. Jiang, Z., Ding, C., Liu, M., Tao, D.: Two-stage cascaded u-net: 1st place solution to brats challenge 2019 segmentation task. In: International MICCAI Brainlesion Workshop. pp. 231–241. Springer (2019)
16. Kim, Y.D., Park, E., Yoo, S., Choi, T., Yang, L., Shin, D.: Compression of deep convolutional neural networks for fast and low power mobile applications. arXiv preprint arXiv:1511.06530 (2015)
17. Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I., Lempitsky, V.: Speeding-up convolutional neural networks using fine-tuned cp-decomposition. arXiv preprint arXiv:1412.6553 (2014)
18. McKinley, R., Meier, R., Wiest, R.: Ensembles of densely-connected cnns with label-uncertainty for brain tumor segmentation. In: International MICCAI Brainlesion Workshop. pp. 456–465. Springer (2018)
19. Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al.: The multimodal brain tumor image segmentation benchmark (brats). IEEE transactions on medical imaging **34**(10), 1993–2024 (2014)
20. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV). pp. 565–571. IEEE (2016)
21. Myronenko, A.: 3d mri brain tumor segmentation using autoencoder regularization. In: International MICCAI Brainlesion Workshop. pp. 311–320. Springer (2018)
22. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in neural information processing systems. pp. 8026–8037 (2019)
23. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
24. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
25. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
26. Tai, C., Xiao, T., Zhang, Y., Wang, X., et al.: Convolutional neural networks with low-rank regularization. arXiv preprint arXiv:1511.06067 (2015)
27. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)