3D MRI Brain Tumor Segmentation Using Autoencoder Regularization

Andriy Myronenko(✉)

NVIDIA, Santa Clara, CA, USA
amyronenko@nvidia.com

Abstract. Automated segmentation of brain tumors from 3D magnetic resonance images (MRIs) is necessary for the diagnosis, monitoring, and treatment planning of the disease. Manual delineation practices require anatomical knowledge, are expensive, time consuming and can be inaccurate due to human error. Here, we describe a semantic segmentation network for tumor subregion segmentation from 3D MRIs based on encoder-decoder architecture. Due to a limited training dataset size, a variational auto-encoder branch is added to reconstruct the input image itself in order to regularize the shared decoder and impose additional constraints on its layers. The current approach won 1st place in the BraTS 2018 challenge.

1 Introduction

Brain tumors are categorized into primary and secondary tumor types. Primary brain tumors originate from brain cells, whereas secondary tumors metastasize into the brain from other organs. The most common type of primary brain tumors are gliomas, which arise from brain glial cells. Gliomas can be of low-grade (LGG) and high-grade (HGG) subtypes. High grade gliomas are an aggressive type of malignant brain tumor that grow rapidly, usually require surgery and radiotherapy and have poor survival prognosis. Magnetic Resonance Imaging (MRI) is a key diagnostic tool for brain tumor analysis, monitoring and surgery planning. Usually, several complimentary 3D MRI modalities are acquired - such as T1, T1 with contrast agent (T1c), T2 and Fluid Attenuation Inversion Recover (FLAIR) - to emphasize different tissue properties and areas of tumor spread. For example the contrast agent, usually gadolinium, emphasizes hyperactive tumor subregions in T1c MRI modality.

Automated segmentation of 3D brain tumors can save physicians time and provide an accurate reproducible solution for further tumor analysis and monitoring. Recently, deep learning based segmentation techniques surpassed traditional computer vision methods for dense semantic segmentation. Convolutional neural networks (CNN) are able to learn from examples and demonstrate state-of-the-art segmentation accuracy both in 2D natural images [6] and in 3D medical image modalities [19].

Multimodal Brain Tumor Segmentation Challenge (BraTS) aims to evaluate state-of-the-art methods for the segmentation of brain tumors by providing

© Springer Nature Switzerland AG 2019
A. Crimi et al. (Eds.): BrainLes 2018, LNCS 11384, pp. 311–320, 2019.
https://doi.org/10.1007/978-3-030-11726-9_28

a 3D MRI dataset with ground truth tumor segmentation labels annotated by physicians [2–5,18]. This year, BraTS 2018 training dataset included 285 cases (210 HGG and 75 LGG), each with four 3D MRI modalities (T1, T1c, T2 and FLAIR) rigidly aligned, resampled to $1 \times 1 \times 1$ mm isotropic resolution and skull-stripped. The input image size is $240 \times 240 \times 155$. The data were collected from 19 institutions, using various MRI scanners. Annotations include 3 tumor subregions: the enhancing tumor, the peritumoral edema, and the necrotic and non-enhancing tumor core. The annotations were combined into 3 nested subregions: whole tumor (WT), tumor core (TC) and enhancing tumor (ET), as shown in Fig. 2. Two additional datasets without the ground truth labels were provided for validation and testing. These datasets required participants to upload the segmentation masks to the organizers' server for evaluations. The validation dataset (66 cases) allowed multiple submissions and was designed for intermediate evaluations. The testing dataset (191 cases) allowed only a single submission, and was used to calculate the final challenge ranking.

In this work, we describe our semantic segmentation approach for volumetric 3D brain tumor segmentation from multimodal 3D MRIs, which won the BraTS 2018 challenge. We follow the encoder-decoder structure of CNN, with asymmetrically large encoder to extract deep image features, and the decoder part reconstructs dense segmentation masks. We also add the variational autoencoder (VAE) branch to the network to reconstruct the input images jointly with segmentation in order to regularize the shared encoder. At inference time, only the main segmentation encode-decoder part is used.

2 Related Work

Last year, BraTS 2017, top performing submissions included Kamnitsas et al. [13] who proposed to ensemble several models for robust segmentation (EMMA), and Wang et al. [21] who proposed to segment tumor subregions in cascade using anisotropic convolutions. EMMA takes advantage of an ensemble of several independently trained architectures. In particular, EMMA combined DeepMedic [14], FCN [16] and U-net [20] models and ensembled their segmentation predictions. During training they used a batch size of 8, and a crop of $64 \times 64 \times 64$ 3D patch. EMMA's ensemble of different models demonstrated a good generalization performance winning the BraTS 2017 challenge. Wang et al. [21] second place paper took a different approach, by training 3 networks for each tumor subregion in cascade, with each subsequent network taking the output of the previous network (cropped) as its input. Each network was similar in structure and consists of a large encoder part (with dilated convolutions) and a basic decoder. They also decompose the $3 \times 3 \times 3$ convolution kernel into intra-slice ($3 \times 3 \times 1$) and inter-slice ($1 \times 1 \times 3$) kernel to save on both the GPU memory and the computational time.

This year, BraTS 2018 top performing submission (in addition to the current work) included Isensee et al. [12] in the 2nd place, McKinly et al. [17] and Zhou et al. [23], who shared the 3rd place. Isensee et al. [12] demonstrated that a

generic U-net architecture with a few minor modifications is enough to achieve competitive performance. The authors used a batch size of 2 and a crop size of $128 \times 128 \times 128$. Furthermore, the authors used an additional training data from their own institution (which yielded some improvements for the enhancing tumor dice).

McKinly et al. [17] proposed a segmentation CNN in which a DenseNet [11] structure with dilated convolutions was embedded in U-net-like network. The authors also introduce a new loss function, a generalization of binary cross-entropy, to account for label uncertainty. Finally, Zhou et al. [23] proposed to use an ensemble of different networks: taking into account multi-scale context information, segmenting 3 tumor subregions in cascade with a shared backbone weights and adding an attention block.

Compared to the related works, we use the largest crop size of $160 \times 192 \times 128$ but compromise the batch size to be 1 to be able to fit network into the GPU memory limits. We also output all 3 nested tumor subregions directly after the sigmoid (instead of using several networks or the softmax over the number of classes). Finally, we add an additional branch to regularize the shared encoder, used only during training. We did not use any additional training data and used only the provided training set.

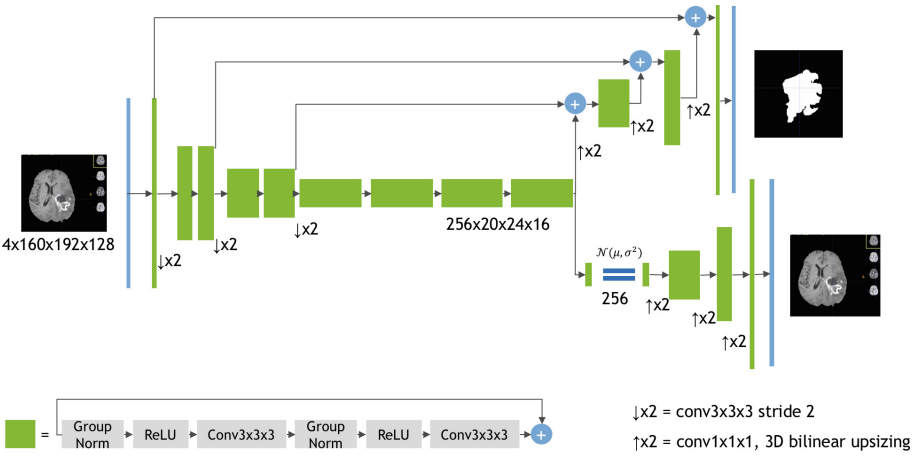

Fig. 1. Schematic visualization of the network architecture. Input is a four channel 3D MRI crop, followed by initial $3 \times 3 \times 3$ 3D convolution with 32 filters. Each green block is a ResNet-like block with the GroupNorm normalization. The output of the segmentation decoder has three channels (with the same spatial size as the input) followed by a sigmoid for segmentation maps of the three tumor subregions (WT, TC, ET). The VAE branch reconstructs the input image into itself, and is used only during training to regularize the shared encoder. (Color figure online)

3 Methods

Our segmentation approach follows encoder-decoder based CNN architecture with an asymmetrically larger encoder to extract image features and a smaller decoder to reconstruct the segmentation mask [6,7,9,19,20]. We add an additional branch to the encoder endpoint to reconstruct the original image, similar to auto-encoder architecture. The motivation for using the auto-encoder branch is to add additional guidance and regularization to the encoder part, since the training dataset size is limited. We follow the variational auto-encoder (VAE) approach to better cluster/group the features of the encoder endpoint. We describe the building parts of our networks in the next subsections (see also Fig. 1).

3.1 Encoder Part

The encoder part uses ResNet [10] blocks, where each block consists of two convolutions with normalization and ReLU, followed by additive identity skip connection. For normalization, we use Group Normalization (GN) [22], which shows better than BatchNorm performance when batch size is small (bath size of 1 in our case). We follow a common CNN approach to progressively downsize image dimensions by 2 and simultaneously increase feature size by 2. For downsizing we use strided convolutions. All convolutions are $3 \times 3 \times 3$ with initial number of filters equal to 32. The encoder endpoint has size $256 \times 20 \times 24 \times 16$, and is 8 times spatially smaller than the input image. We decided against further downsizing to preserve more spatial content.

3.2 Decoder Part

The decoder structure is similar to the encoder one, but with a single block per each spatial level. Each decoder level begins with upsizing: reducing the number of features by a factor of 2 (using $1 \times 1 \times 1$ convolutions) and doubling the spatial dimension (using 3D bilinear upsampling), followed by an addition of encoder output of the equivalent spatial level. The end of the decoder has the same spatial size as the original image, and the number of features equal to the initial input feature size, followed by $1 \times 1 \times 1$ convolution into 3 channels and a sigmoid function.

3.3 VAE Part

Starting from the encoder endpoint output, we first reduce the input to a low dimensional space of 256 (128 to represent mean, and 128 to represent std). Then, a sample is drawn from the Gaussian distribution with the given mean and std, and reconstructed into the input image dimensions following the same architecture as the decoder, except we don't use the inter-level skip connections from the encoder here. The VAE part structure is shown in Table 1.

Table 1. VAE decoder branch structure, where GN stands for group normalization (with group size of 8), Conv - $3 \times 3 \times 3$ convolution, Conv1 - $1 \times 1 \times 1$ convolution, AddId - addition of identity/skip connection, UpLinear - 3D linear spatial upsampling, Dense - fully connected layer

Name	Ops	Repeat	Output size
VD	GN, ReLU, Conv (16) stride 2, Dense (256)	1	256×1
VDraw	Sample $\sim \mathcal{N}(\mu(128), \sigma^2(128))$	1	128×1
VU	Dense, ReLU, Conv1, UpLinear	1	$256 \times 20 \times 24 \times 16$
VUp2	Conv1, UpLinear	1	$128 \times 40 \times 48 \times 32$
VBlock2	GN, ReLU, Conv, GN, ReLU, Conv, AddId	1	$128 \times 40 \times 48 \times 32$
VUp1	Conv1, UpLinear	1	$64 \times 80 \times 96 \times 64$
VBlock1	GN, ReLU, Conv, GN, ReLU, Conv, AddId	1	$64 \times 80 \times 96 \times 64$
VUp0	Conv1, UpLinear	1	$32 \times 160 \times 192 \times 128$
VBlock0	GN, ReLU, Conv, GN, ReLU, Conv, AddId	1	$32 \times 160 \times 192 \times 128$
Vend	Conv1	1	$4 \times 160 \times 192 \times 128$

3.4 Loss

Our loss function consists of 3 terms:

$$\mathbf{L} = \mathbf{L}_{dice} + 0.1 * \mathbf{L}_{\mathrm{L2}} + 0.1 * \mathbf{L}_{\mathrm{KL}} \tag{1}$$

$\mathbf{L}_{dice}$ is a soft dice loss [19] applied to the decoder output p_{pred} to match the segmentation mask p_{true}:

$$\mathbf{L}_{dice} = \frac{2 * \sum p_{true} * p_{pred}}{\sum p_{true}^2 + \sum p_{pred}^2 + \epsilon} \tag{2}$$

where summation is voxel-wise, and ϵ is a small constant to avoid zero division. Since the output of the segmentation decoder has 3 channels (predictions for each tumor subregion), we simply add the three dice loss functions together.

$\mathbf{L}_{\mathrm{L2}}$ is an L2 loss on the VAE branch output I_{pred} to match the input image I_{input}:

$$\mathbf{L}_{\mathrm{L2}} = ||I_{input} - I_{pred}||_2^2 \tag{3}$$

$\mathbf{L}_{\mathrm{KL}}$ is standard VAE penalty term [8,15], a KL divergence between the estimated normal distribution $\mathcal{N}(\mu, \sigma^2)$ and a prior distribution $\mathcal{N}(0, 1)$, which has a closed form representation:

$$\mathbf{L}_{\mathrm{KL}} = \frac{1}{N} \sum \mu^2 + \sigma^2 - \log \sigma^2 - 1 \tag{4}$$

where N is total number of image voxels. We empirically found a hyper-parameter weight of 0.1 to provide a good balance between dice and VAE loss terms in Eq. 1.

3.5 Optimization

We use Adam optimizer with initial learning rate of $\alpha_0 = 1e{-}4$ and progressively decrease it according to:

$$\alpha = \alpha_0 * \left(1 - \frac{e}{N_e}\right)^{0.9} \tag{5}$$

where e is an epoch counter, and N_e is a total number of epochs (300 in our case). We use batch size of 1, and draw input images in random order (ensuring that each training image is drawn once per epoch).

3.6 Regularization

We use L2 norm regularization on the convolutional kernel parameters with a weight of $1e-5$. We also use the spatial dropout with a rate of 0.2 after the initial encoder convolution. We have experimented with other placements of the dropout (including placing dropout layer after each convolution), but did not find any additional accuracy improvements.

3.7 Data Preprocessing and Augmentation

We normalize all input images to have zero mean and unit std (based on non-zero voxels only). We apply a random (per channel) intensity shift ($-0.1..0.1$ of image std) and scale (0.9..1.1) on input image channels. We also apply a random axis mirror flip (for all 3 axes) with a probability 0.5.

4 Results

We implemented our network in Tensorflow [1] and trained it on NVIDIA Tesla V100 32 GB GPU using BraTS 2018 training dataset (285 cases) without any additional in-house data. During training we used a random crop of size $160 \times 192 \times 128$, which ensures that most image content remains within the crop area. We concatenated 4 available 3D MRI modalities into the 4 channel image as an input. The output of the network is 3 nested tumor subregions (after the sigmoid).

We report the results of our approach on BraTS 2018 validation (66 cases) and the testing sets (191 cases). These datasets were provided with unknown glioma grade and unknown segmentation. We uploaded our segmentation results to the BraTS 2018 server for evaluation of per class dice, sensitivity, specificity and Hausdorff distances.

Aside from evaluating a single model, we also applied test time augmentation (TTA) by mirror flipping the input 3D image axes, and averaged the output of the resulting 8 flipped segmentation probability maps. Finally, we ensembled a set of 10 models (trained from scratch) to further improve the performance.

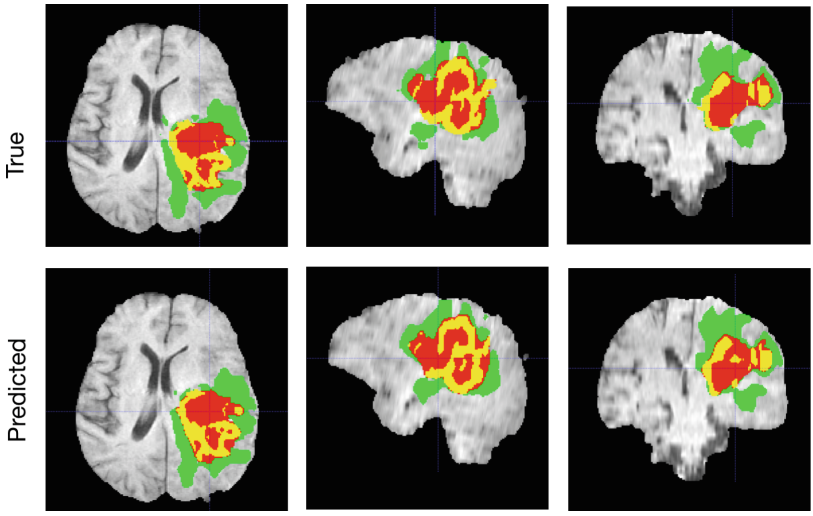

Fig. 2. A typical segmentation example with true and predicted labels overlaid over T1c MRI axial, sagittal and coronal slices. The whole tumor (WT) class includes all visible labels (a union of green, yellow and red labels), the tumor core (TC) class is a union of red and yellow, and the enhancing tumor core (ET) class is shown in yellow (a hyperactive tumor part). The predicted segmentation results match the ground truth well. (Color figure online)

Table 2 shows the results of our model on the BraTS 2018 validation dataset. At the time of initial short paper submission (Jul 13, 2018), our dice accuracy performance was second best (team name NVDLMED[1]) for all of the 3 segmentation labels (ET, WT, TC). The TTA only marginally improved the performance, but the ensemble of 10 models resulted in 1% improvement, which is consistent with the literature results of using ensembles.

For the testing dataset, only a single submission was allowed. Our results are shown in Table 3, which won the 1st place at BraTS 2018 challenge.

Table 2. BraTS 2018 validation dataset results. Mean Dice and Hausdorff measurements of the proposed segmentation method. EN - enhancing tumor core, WT - whole tumor, TC - tumor core.

Validation dataset	Dice			Hausdorff (mm)		
	ET	WT	TC	ET	WT	TC
Single model	0.8145	0.9042	0.8596	3.8048	4.4834	8.2777
Single model + TTA	0.8173	0.9068	0.8602	3.8241	4.4117	6.8413
Ensemble of 10 models	0.8233	0.9100	0.8668	3.9257	4.5160	6.8545

[1] https://www.cbica.upenn.edu/BraTS18/lboardValidation.html.

Table 3. BraTS 2018 testing dataset results. Mean Dice and Hausdorff measurements of the proposed segmentation method. EN - enhancing tumor core, WT - whole tumor, TC - tumor core.

Testing dataset	Dice			Hausdorff (mm)		
	ET	WT	TC	ET	WT	TC
Ensemble of 10 models	0.7664	0.8839	0.8154	3.7731	5.9044	4.8091

Time-wise, each training epoch (285 cases) on a single GPU (NVIDIA Tesla V100 32 GB) takes 9 min. Training the model for 300 epochs takes 2 days. We've also trained the model on NVIDIA DGX-1 server (that includes 8 V100 GPUs interconnected with NVLink); this allowed to train the model in 6 h (7.8x speed up). The inference time is 0.4 s for a single model on a single V100 GPU.

5 Discussion and Conclusion

In this work, we described a semantic segmentation network for brain tumor segmentation from multimodal 3D MRIs, which won the BraTS 2018 challenge. While experimenting with network architectures, we have tried several alternative approaches. For instance, we have tried a larger batch size of 8 to be able to use BatchNorm (and take advantage of batch statistics), however due to the GPU memory limits this modification required to use a smaller image crop size, and resulted in worse performance. We have also experimented with more sophisticated data augmentation techniques, including random histogram matching, affine image transforms, and random image filtering, which did not demonstrate any additional improvements. We have tried several data post-processing techniques to fine tune the segmentation predictions with CRF [14], but did not find it beneficial (it helped for some images, but made some other image segmentation results worse). Increasing the network depth further did not improve the performance, but increasing the network width (the number of features/filters) consistently improved the results. Using the NVIDIA Volta V100 32 GB GPU we were able to double the number of features compared to V100 16 GB version. Finally, the additional VAE branch helped to regularize the shared encoder (in presence of limited data), which not only improved the performance, but helped to consistently achieve good training accuracy for any random initialization. Our BraTS 2018 testing dataset results are 0.7664, 0.8839 and 0.8154 average dice for enhanced tumor core, whole tumor and tumor core, respectively.

References

1. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). https://www.tensorflow.org/, software available from tensorflow.org
2. Bakas, S., et al.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-GBM collection. Cancer Imaging Arch. (2017). https://doi.org/10.7937/K9/TCIA.2017.KLXWJJ1Q

3. Bakas, S., et al.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-LGG collection. Cancer Imaging Arch. (2017). https://doi.org/10.7937/K9/TCIA.2017.GJQ7R0EF
4. Bakas, S., et al.: Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Sci. Data **4**, 170117 (2017)
5. Bakas, S., Reyes, M., et al.: Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge. In: arXiv:1811.02629 (2018)
6. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. arXiv:1802.02611 (2018)
7. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1800–1807 (2017)
8. Doersch, C.: Tutorial on variational autoencoders. arxiv e-print (2016). http://arxiv.org/abs/1606.05908
9. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. In: Proceedings of the International Conference on Computer Vision (ICCV) (2017)
10. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38
11. 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. 2261–2269 (2017)
12. Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., Maier-Hein, K.H.: No new-net. In: Crimi, A., Bakas, S., Kuijf, H., Keyvan, F., Reyes, M., van Walsum, T. (eds.): BrainLes 2018. LNCS, vol. 11384, pp. 234–244. Springer, Cham (2019)
13. Kamnitsas, K., et al.: Ensembles of multiple models and architectures for robust brain tumour segmentation. In: Crimi, A., Bakas, S., Kuijf, H., Menze, B., Reyes, M. (eds.) BrainLes 2017. LNCS, vol. 10670, pp. 450–462. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75238-9_38
14. Kamnitsas, K., et al.: Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Med. Image Anal. **36**, 61–78 (2016)
15. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In: The International Conference on Learning Representations (ICLR) (2014)
16. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440 (2015)
17. McKinley, R., Meier, R., Wiest, R.: Ensembles of densely-connected CNNs with label-uncertainty for brain tumor segmentation. In: Crimi, A., Bakas, S., Kuijf, H., Keyvan, F., Reyes, M., van Walsum, T. (eds.): BrainLes 2018. LNCS, vol. 11384, pp. 456–465. Springer, Cham (2019)
18. Menze, B.H., et al.: The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Trans. Med. Imaging **34**(10), 1993–2024 (2015)
19. Milletari, F., Navab, N., Ahmadi, S.A.: V-Net: fully convolutional neural networks for volumetric medical image segmentation. In: Fourth International Conference on 3D Vision (3DV) (2016)
20. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28

21. Wang, G., Li, W., Ourselin, S., Vercauteren, T.: Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In: Crimi, A., Bakas, S., Kuijf, H., Menze, B., Reyes, M. (eds.) BrainLes 2017. LNCS, vol. 10670, pp. 178–190. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75238-9_16
22. Wu, Y., He, K.: Group normalization. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11217, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01261-8_1
23. Zhou, C., Chen, S., Ding, C., Tao, D.: Learning contextual and attentive information for brain tumor segmentation. In: Crimi, A., Bakas, S., Kuijf, H., Keyvan, F., Reyes, M., van Walsum, T. (eds.): BrainLes 2018. LNCS, vol. 11384, pp. 497–507. Springer, Cham (2019)