Squeeze-and-Excitation Normalization for Brain Tumor Segmentation

Andrei Iantsen(✉), Vincent Jaouen, Dimitris Visvikis, and Mathieu Hatt

LaTIM, INSERM, UMR 1101, University Brest, Brest, France
andrei.iantsen@inserm.fr

Abstract. In this paper we described our approach for glioma segmentation in multi-sequence magnetic resonance imaging (MRI) in the context of the MICCAI 2020 Brain Tumor Segmentation Challenge (BraTS). We proposed an architecture based on U-Net with a new computational unit termed "SE Norm" that brought significant improvements in segmentation quality. Our approach obtained competitive results on the validation (Dice scores of 0.780, 0.911, 0.863) and test (Dice scores of 0.805, 0.887, 0.843) sets for the enhanced tumor, whole tumor and tumor core sub-regions. The full implementation and trained models are available at https://github.com/iantsen/brats.

Keywords: Medical imaging · Brain tumor segmentation · SE norm · U-Net

1 Introduction

Glioma is a group of malignancies that arises from the glial cells in the brain. Nowadays, gliomas are the most common primary tumors of the central nervous system [1,2]. The symptoms of patients presenting with a glioma depend on the anatomical site of the glioma in the brain and can be too common (e.g. headaches, nausea or vomiting, mood and personality alterations) to give an accurate diagnosis in early stages of the disease. The primary diagnosis is usually confirmed by magnetic resonance imaging (MRI) or computed tomography (CT) that provide additional structural information about the tumor.

Gliomas usually consist of heterogeneous sub-regions (edema, enhancing and non-enhancing tumor core, etc.) with variable histologic and genomic phenotypes [1]. Presently, multimodal MRI scans are used for non-invasive tumor evaluation and treatment planning, due to its ability to depict the tumor sub-regions with different intensities. However, segmentation of brain tumors in multimodal MRI scans is one of the most challenging tasks in medical imaging because of the high heterogenity in tumor appearances and shapes.

The brain tumor segmentation challenge (BraTS) [3–6] is aimed at development of automatic methods for the brain tumor segmentation. All participants of the BraTS are provided with a clinically-acquired training dataset of pre-operative MRI scans (4 sequences per patient) and segmentation masks for

A. Crimi and S. Bakas (Eds.): BrainLes 2020, LNCS 12659, pp. 366–373, 2021.
https://doi.org/10.1007/978-3-030-72087-2_32

three different tumor sub-regions, namely the GD-enhancing tumor, the peritumoral edema, and and the necrotic and non-enhancing tumor core. The MRI scans were acquired with different clinical protocols and various scanners from multiple 19 institutions. Each scan was annotated manually by one to four raters and subsequently approved by expert raters.

The performance of proposed algorithms was evaluated by the Dice score, sensitivity, specificity and the 95th percentile of the Hausdorff distance.

2 Materials and Methods

2.1 SE Normalization

Normalization layers have become an integral part of modern deep neural networks. Existing methods, such as Batch Normalization [7], Instance Normalization [8], Layer Normalization [9], etc., have been shown to be effective for training different types of deep learning models. In essence, any normalization layer performs the following computations. First, for a n-dimensional input $X = (x^{(1)}, x^{(2)}, \ldots, x^{(n)})$, we normalize each dimension

$$x'^{(i)} = \frac{1}{\sigma^{(i)}}(x^{(i)} - \mu^{(i)}) \tag{1}$$

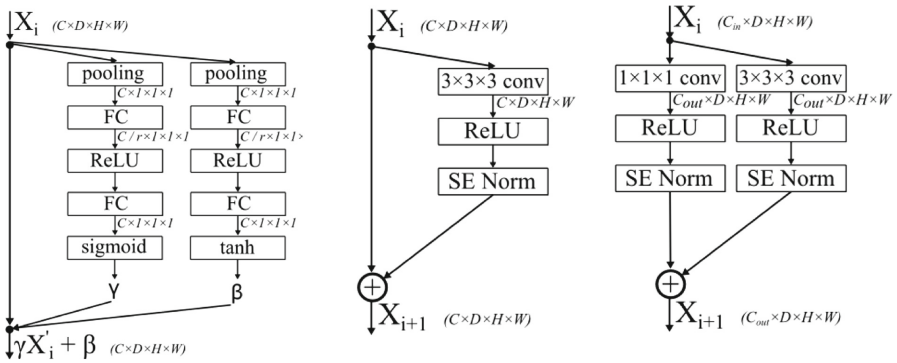
where $\mu^{(i)} = \mathrm{E}[x^{(i)}]$ and $\sigma^{(i)} = \sqrt{\mathrm{Var}[x^{(i)}] + \epsilon}$ with ϵ as a small constant. Normalization layers mainly differ in terms of the dimensions chosen to compute the mean and standard deviation [10]. Batch Normalization, for example, uses the values calculated for each channel within a batch of examples, whereas Instance Normalization - within a single example. Second, a pair of parameters γ_k, β_k are applied to each channel k to scale and shift the normalized values:

$$y_k = \gamma_k x'_k + \beta_k \tag{2}$$

The parameters γ_k, β_k are fitted in the course of training and enable the layer to represent the identity transform, if necessary. During inference, both parameters are *fixed* and *independent* of the input X. In this paper, we propose to apply *instance-wise normalization* and design each parameter γ_k, β_k as *functions* of the input X, i.e.

$$\gamma = f_\gamma(X) \tag{3}$$
$$\beta = f_\beta(X) \tag{4}$$

where $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_\mathrm{K})$ and $\beta = (\beta_1, \beta_2, \ldots, \beta_\mathrm{K})$ - the scale and shift parameters for all channels, K is a number of channels. We represent the function f_γ using the original Squeeze-and-Excitation (SE) block with the sigmoid [11], whereas f_β is modeled with the SE block with the tanh activation function to enable the negative shift (see Fig. 1a). This new architectural unit, that we refer to as *SE Normalization* (*SE Norm*), is the major component of our model.



(a) SE Normalization layer

(b) Residual layer with the shortcut connection

(c) Residual layer with the non-linear projection

Fig. 1. Proposed layers. Output dimensions are depicted in brackets.

2.2 Network Architecture

The widely used 3D U-Net [12,13] serves as the basis to design our model. The basic element of the model, a convolutional block comprised of a $3 \times 3 \times 3$ convolution followed by the ReLU activation function and the SE Norm layer, is used to construct the decoder (Fig. 2, blue blocks). In the encoder, we utilize residual layers [14] consist of convolutional blocks with shortcut connections (see Fig. 1b). If numbers of input / output channels in a residual layer are different, we perform a non-linear projection by adding the $1 \times 1 \times 1$ convolutional block to the shortcut in order to match the dimensions (see Fig. 1c).

In the encoder, we perform downsampling applying max pooling with the kernel size of $2 \times 2 \times 2$. To linearly upsample feature maps in the decoder, we use $3 \times 3 \times 3$ transposed convolutions. In addition, we supplement the decoder with three upsampling paths to transfer low-resolution features further in the model by applying the $1 \times 1 \times 1$ convolutional block to reduce the number of channels, and utilizing trilinear interpolation to increase the spatial size of the feature maps (Fig. 2, yellow blocks).

The first residual layer placed after the input is implemented with the kernel size of $7 \times 7 \times 7$ to increase the receptive field of the model without significant computational overhead. The softmax layer is applied to output probabilities for four target classes.

To regularize the model, we add Spatial Dropout layers [15] right after the last residual block at each stage in the encoder and before $1 \times 1 \times 1$ convolution in the decoder tail (Fig. 2, red blocks).

2.3 Data Preprocessing

Intensities of MRI scans are not standardized and typically exhibit a high variability in both intra- and inter-image domains. In order to decrease the intensity

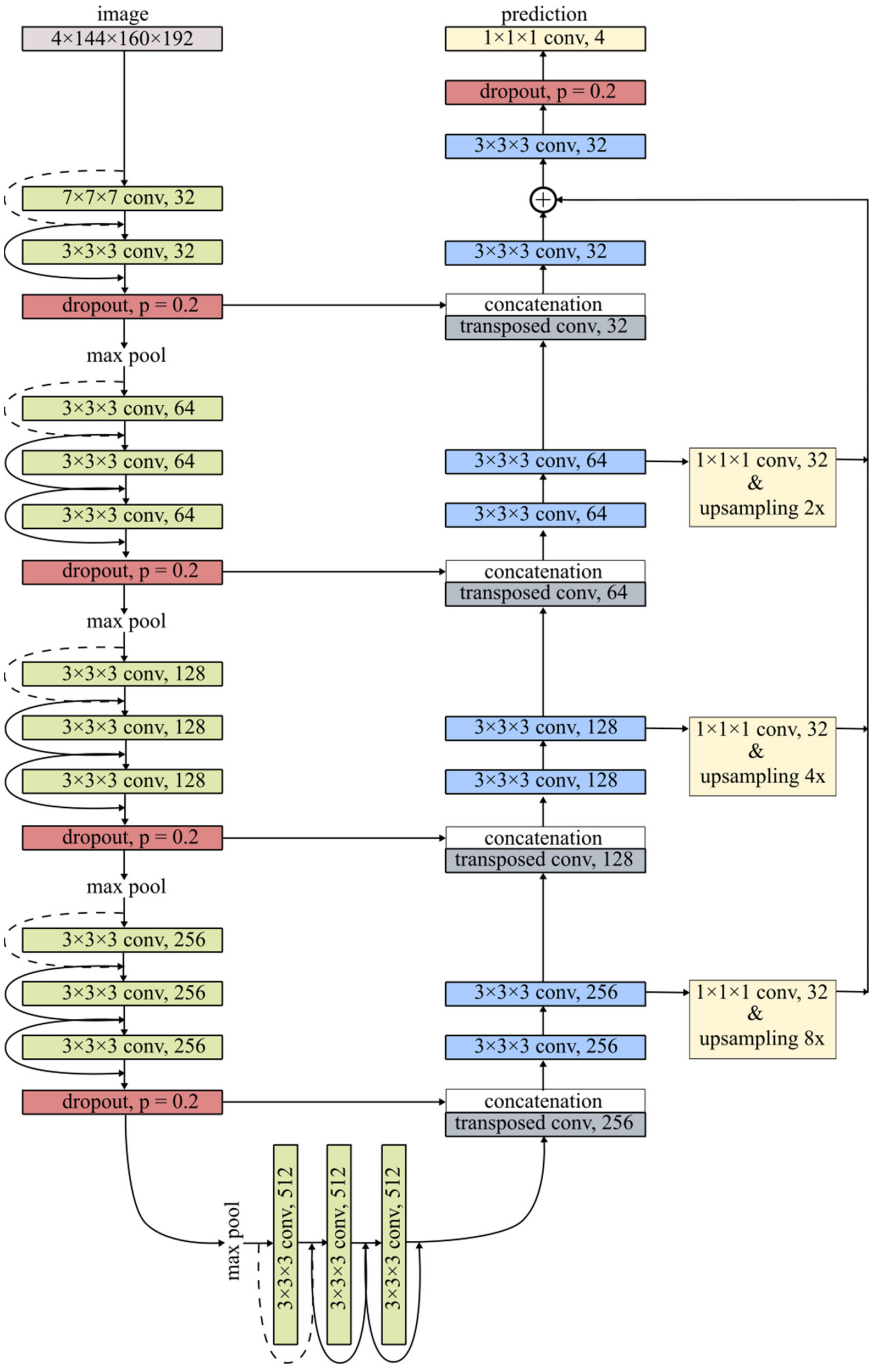


Fig. 2. Proposed network architecture with SE normalization. (Color figure online)

inhomogeneity, we perform Z-score normalization for each MRI sequence and each patient separately. The mean and standard deviation are calculated based on non-zero voxels corresponding to the brain region. All background voxels remain unchanged after the normalization.

2.4 Training Procedure

Due to the large size of provided MRI scans, we perform training on random patches of the size $144 \times 160 \times 192$ voxels (*depth* $\times$ height $\times$ *width*) on two GPUs NVIDIA GeForce GTX 1080 Ti (11 GB) with a batch size of 2 (one sample per worker).

We train the model for 300 epochs using Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ for exponential decay rates for moment estimates, and apply a cosine annealing schedule gradually reducing the learning rate from $lr_{max} = 10^{-4}$ to $lr_{min} = 10^{-6}$ within 25 epochs and performing the learning rate adjustment at each epoch.

2.5 Loss Function

We utilize the unweighted sum of the Soft Dice Loss [16] and the Focal Loss [17] as the loss function in the course of training. The Soft Dice Loss is the differentiable surrogate to optimize the Dice score that is one of the evaluation metrics used in the challenge. The Focal Loss, compared to the Soft Dice Loss, has much smoother optimization surface that ease the model training.

Based on [16], the Soft Dice Loss for one training example can be written as

$$L_{Dice}(y, \hat{y}) = 1 - \frac{1}{\mathrm{C}} \sum_{c=1}^{\mathrm{C}} \frac{2 \sum_i^{\mathrm{N}} y_i^c \hat{y}_i^c + 1}{\sum_i^{\mathrm{N}} y_i^c + \sum_i^{\mathrm{N}} \hat{y}_i^c + 1} \tag{5}$$

The Focal Loss is defined as

$$L_{Focal}(y, \hat{y}) = -\frac{1}{\mathrm{N}} \sum_{i}^{\mathrm{N}} \sum_{c=1}^{\mathrm{C}} y_i^c (1 - \hat{y}_i^c)^{\gamma} \ln(\hat{y}_i^c) \tag{6}$$

In both definitions, $y_i = \left[y_i^1, y_i^2, \ldots, y_i^{\mathrm{C}}\right]^\top$ - the one-hot encoded label for the *i-th* voxel, $\hat{y}_i = \left[\hat{y}_i^1, \hat{y}_i^2, \ldots, \hat{y}_i^{\mathrm{C}}\right]^\top$ - predicted probabilities for the *i-th* voxel. N and C are the total numbers of voxels and classes for the given example, respectively. Additionally we apply Laplacian smoothing by adding +1 to the numerator and denominator in the Soft Dice Loss to avoid the zero division in cases when one or several labels are not represented in the training example. The parameter γ in the Focal Loss is set at 2.

The training data in the challenge has labels for three tumor sub-regions, namely the necrotic and non-enhancing tumor core (NCR & NET), the peritumoral edema (ED) and the GD-enhancing tumor (ET). However, the evaluation is done for the GD-enhancing tumor (ET), the tumor core (TC), which is comprised of NCR & NET along with ET, and the whole tumor (WT) that combines

all provided sub-regions. Hence, during training we optimize the loss directly on these nested tumor sub-regions.

2.6 Ensembling

To reduce the variance of the model predictions, we build an ensemble of models that are trained on different splits of the train set and use the average as the ensemble prediction. At each iteration, the model is built on 90%/10% splits of the train set and subsequently evaluated on the online validation set. Having repeated this procedure multiple times, we choose 20 models with the highest performance on the online validation set and combine them into the ensemble. Predictions on the test set are produced by averaging predictions of the individual models and applying a threshold operation with a value equal to 0.5.

2.7 Post-processing

The Dice score used for the performance evaluation in the challenge is highly sensitive to cases wherein the model predicts classes that are not presented in the ground truth. Therefore, a false positive prediction for a single voxel leads to the lowest value of the Dice score and might significantly affect the average model performance on the whole evaluation dataset. This primarily refers to patients without ET sub-regions. To address this issue, we add a post-processing step to remove small ET regions from the model outcome if their area is less than a certain threshold. We set its value at 32 voxels since it is the smallest ET area among all patients in the train set.

3 Results and Discussion

The results of the BraTS 2020 segmentation challenge are presented in Table 1 and Table 2. The Dice score, Sensitivity and Hausdorff distance (HD) were utilized for the evaluation. Results in Table 1 were obtained on the online validation set with 125 patients without publicly available segmentation masks. The U-Net model was used as a baseline for comparison purposes. Final results on the test set consisted of 166 patients are shown in Table 2.

For all cases, the lowest average Dice score was obtained for the ET sub-region. This can be partially explained by the relatively small size of the ET class compared to the other tumor sub-regions that made segmentation of this class more challenging. The proposed model outperformed U-Net in all evaluation metrics except for the Dice score for the ET class. It is mainly caused by cases wherein the ET sub-regions were not presented. Combining multiple models into the ensemble allowed to address this issue since it reduced the chance to receive false positive predictions for the ET class as well as led to the better performance in terms of HD.

Table 1. Performance on the online validation set ($n = 125$). Average results are provided for each evaluation metrics.

Metrics	Dice score			Sensitivity			HD		
Class	ET	WT	TC	ET	WT	TC	ET	WT	TC
U-Net	0.772	0.899	0.825	0.794	0.896	0.813	5.813	5.973	6.576
Best Model	0.740	0.908	0.862	0.816	0.909	0.854	3.841	4.602	5.339
Ensemble	0.761	0.911	0.863	0.814	0.908	0.850	3.695	4.475	4.816
Ensemble + pp	0.780	0.911	0.863	0.815	0.908	0.850	3.717	4.475	4.816

Table 2. Performance on the test set ($n = 166$).

Metrics	Dice score			Sensitivity			HD		
Class	ET	WT	TC	ET	WT	TC	ET	WT	TC
Ensemble + pp	0.805	0.887	0.843	0.854	0.909	0.866	15.429	4.535	19.589

References

1. Bakas, S.: Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Nature Sci. Data **4**(1), 1–13 (2017)
2. Upadhyay, N., Waldman, A.: Conventional MRI evaluation of gliomas. British J. Radiol. **84**(2), Special Issue 2 S107–S111 (2011)
3. Menze, B.H., et al.: The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Trans. Med. Imaging **34**(10), 1993–2024 (2015)
4. Bakas, S. et al: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-GBM collection. TCIA (2017)
5. Bakas, S., et al.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-LGG collection. TCIA (2017)
6. Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A. 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)
7. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv preprint arXiv:1502.03167 (2015)
8. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
9. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer Normalization. arXiv preprint arXiv:1607.06450 (2016)
10. Wu, Y., He, K.: Group normalization. In: European Conference on Computer Vision (ECCV) (2018)
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks, CoRR, vol. abs/1709.01507 (2017). http://arxiv.org/abs/1709.01507
12. 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

13. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-net: learning dense volumetric segmentation from sparse annotation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 424–432. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_49
14. 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
15. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient Object Localization Using Convolutional Networks. arXiv preprint arXiv:1411.4280 (2014)
16. Milletari, F., Navab, N., Ahmadi, S.-A.: V-net: fully convolutional neural networks for volumetric medical image segmentation. In: International Conference on 3D Vision, pp. 565–571. IEEE (2016)
17. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal Loss for Dense Object Detection. arXiv preprint arXiv:1708.02002 (2017)