Brain Tumor Segmentation Based on 3D Residual U-Net

Megh Bhalerao[1](✉) and Siddhesh Thakur[2](✉)

[1] National Institute of Technology, Karnataka, Surathkal, India
megh.bhalerao@gmail.com
[2] Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded, India
sid.cre8er@gmail.com

Abstract. We propose a deep learning based approach for automatic brain tumor segmentation utilizing a three-dimensional U-Net extended by residual connections. In this work, we did not incorporate architectural modifications to the existing 3D U-Net, but rather evaluated different training strategies for potential improvement of performance. Our model was trained on the dataset of the International Brain Tumor Segmentation (BraTS) challenge 2019 that comprise multi-parametric magnetic resonance imaging (mpMRI) scans from 335 patients diagnosed with a glial tumor. Furthermore, our model was evaluated on the BraTS 2019 independent validation data that consisted of another 125 brain tumor mpMRI scans. The results that our 3D Residual U-Net obtained on the BraTS 2019 test data are Mean Dice scores of 0.697, 0.828, 0.772 and Hausdorff$_{95}$ distances of 25.56, 14.64, 26.69 for enhancing tumor, whole tumor, and tumor core, respectively.

Keywords: Brain Tumor Segmentation · CNN · Glioblastoma · Segmentation · BraTS

1 Introduction

Gliomas are the most common type of adult brain tumors arising from glial cells. They are classified into High Grade Glioma (HGG - also referred to as Glioblastoma) and Lower Grade Glioma (LGG). Patients diagnosed with an LGG have better prognosis than an HGG patient. Multi-parametric Magnetic Resonance Imaging (mpMRI) is generally used by radiologists to detect the tumorous region in the brain to plan treatment and surgery, as well as post-operative monitoring of the patient.

The shape and structure of gliomas are highly variable making their detection and classification a tedious task, hence making the need for automatic segmentation algorithms imminent. Several machine learning approaches have been propounded in the past, with the most recent advent of deep learning (i.e., Convolutional Neural Networks) showing state-of-the-art performance in several segmentation tasks [1–5]. The crux of segmentation tasks lies in being able to extract global context as well as local information, which is effectively done by the encoder-decoder architecture of the 3D Residual U-net.

A. Crimi and S. Bakas (Eds.): BrainLes 2019, LNCS 11993, pp. 218–225, 2020.
https://doi.org/10.1007/978-3-030-46643-5_21

2 Materials and Methods

2.1 Data

To create our model we used the publicly available training dataset of the International Brain Tumor Segmentation (BraTS) challenge 2019 comprising mpMRI scans of 259 HGG and 76 LGG subjects (335 in total) [6–10]. For every subject, there are four available co-registered ans skull-stripped mpMRI modalities, namely native T1-weighted (T1), post-contrast T1-weighted (T1CE/T1Gd), T2-weighted (T2), and T2 Fluid Attenuated Inversion Recovery (FLAIR). Every subject was also accompanied by a corresponding ground-truth tumor segmentation label map. These label maps are manually-annotated by expert radiologists. The isometric view of the modalities along with the segmentation label is illustrated below.

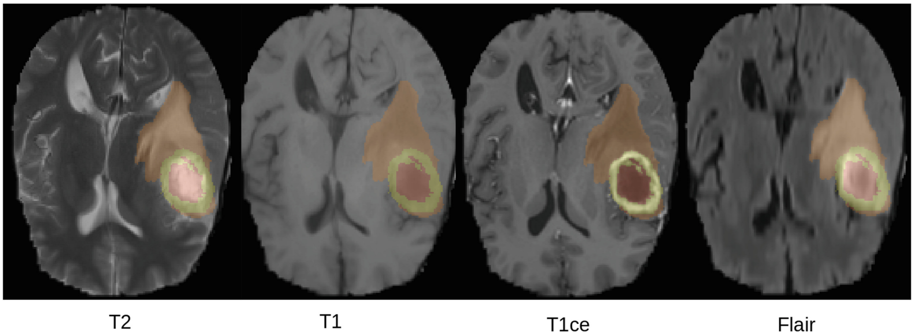

Fig. 1. An example of 4 mpMRI scans with the corresponding label map, comprising the tumor sub-structures. (Color figure online)

Specifically, the tumor label map is divided into 3 regions; non-enhancing tumor core and necrosis (label 1 - red color), enhancing tumor core (label 4 - yellow color), and peritumoral edema (label 2 - orange color), as shown in Fig. 1. The performance metrics used for the segmentation algorithms are the Dice score, the 95% of the Hausdorff Distance, as well as Sensitivity and Specificity.

2.2 Data Pre-processing

The original MRI scans have lots of background voxels with zero intensity value. For computational efficiency we focus on a region of interest (RoI), *i.e.*, the brain. To achieve this, we obtain a bounding box for all non-zero values across all four mpMRI modalities, and crop each of them according to the largest bounding box (amongst the four modalities), to accommodate the Brain-Region in all the four modalities.

After the bounding-box cropping of all mpMRI scans, we pad the images with zeros along every dimension, with the number of zeros chosen such that to make every dimension divisible by 16, and hence account for 4 downsampling layers in our U-Net architecture. We further normalize each image by $\frac{x-\mu}{\sigma}$, where μ and σ are the mean and standard deviation of that particular image's intensity values (it must be noted here that the normalization is done by calculating the μ & σ values only of the non-zero regions of the image). Furthermore, we apply N3 bias field correction to the BraTS data.

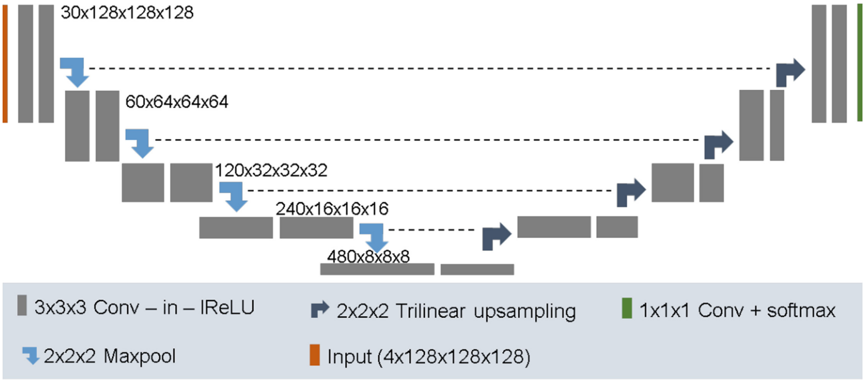

Fig. 2. The architecture followed by our model.

2.3 Network Architecture

We use the 3D U-Net from [2,11] with skip/residual connections as illustrated in Fig. 2.

Our proposed U-Net network is a fully convolutional architecture, meaning that the input image size to the network does not need to be constant (until it meets a certain criteria). The U-Net is an encoder-decoder architecture with 4 downsampling and upsampling modules. The encoder section of the network deciphers semantic information by down-sampling operations, but at the same time loses spatial information due to reduction of image size. This lost spatial information is recovered by the upsampling layers, which perform upsampling using transpose-convolution. A non-linear activation function - Leaky Rectified Linear Unit (leaky ReLU) [12] is applied before every convolution layer, with a leakiness of 10^{-2}. Also since Instance Normalization [13] has been shown to give better results for Image Related Tasks we use Instance Norm instead of Batch-Normalization. In order to avoid the computational bottleneck at the bottom of the "U", skip connections are used which simply concatenate the feature maps of the corresponding upsampling & downsampling layers, so enough contextual information is provided to the network.

Every convolutional module comprises of 2 convolutional layers with the following pipeline: Input → Instance Norm → Leaky ReLU → Convolution →

Dropout → Instance Norm → Leaky ReLU → Convolution → Output (+Input). As we see, we add the input to the output (residual connection), to give something as a shape prior.

At the beginning of the network we start off with 30 filters, successively doubling it with every down-sampling module, reaching a maximum of 480 filters at the bottom of the U. To reduce the computational requirement the number of feature maps is reduced to half, just before the first upsampling layer.

2.4 Training Procedure and Hyper-parameters

Feeding the entire 3D mpMRI scan to the network is very computationally expensive, requiring more than 12 GB of GPU memory, hence we experiment with different divisible-by-16 patch sizes and choose the largest one that fits in the memory. We use a batch-size of 1, again due to computational limitations. The largest image patch that can be employed given computational limitations we want to impose (i.e., Tesla P100 12 GB GPU) is $(128 \times 128 \times 128)$. Randomly extracted patches are used to feed our network. 50 Images are randomly taken from the entire training dataset and are used for internal validation of the model during each epoch. Furthermore, we keep a track of which epoch gives us the best validation loss, and that epoch's model is used to generate the segmentations on the actual validation data. We start with a learning rate of 0.001, using a triangular schedule, with the minimum learning rate being 0.000001. Our learning rate varies in a triangular wave like fashion between these two maximum and minimum values, being updated after every iteration (*i.e.*, after every forward pass + back-propagation). This learning rate schedule is used to avoid the local minima spots where the weights might get stuck if we use a monotonically decreasing learning rate. The model is trained for 200 epochs with no-early stopping being done as of now. A stochastic gradient optimizer is used with a momentum of 0.9.

We use a 5-fold cross validation setup to train our models, by dividing our training dataset into 5 folds and training on 4 folds and validating on 1 fold. Hence our training generates 5 models.

In addition to the four modalities (channels - T1, T2, T1ce, Flair), we add an artificially generated fifth modality, with the aim of providing us with some more meaningful information about the tumor. This additional channel is generated by thresholding the flair modality at 0.2 intensity value which is determined heuristically. Thresholding flair provides better information about the whole tumor region and hence has the potential to improve whole tumor performance. Experiments were also carried out with substituting T1 with the artificial modality and also doing the same with T2. The best results were obtained from the experiment which added a fifth modality instead of substituting.

Our model segments the input into the 3 BraTS classes as mentioned above (in addition to 1 background class since we use a soft-max activation in the final layer). Since the most important metric used to quantitatively evaluate the segmentation performance is the Dice score, we continuously optimize towards maximizing the Dice. In the problem statement of BraTS there is an issue of class imbalance (i.e., the background/brain pixels are a lot more in number than the pixels of the segmentation labels). To partially tackle this issue of class imbalance, the Multi-Class dice loss function is used here. This calculates the Dice for individual classes and then averages them. The multi-class dice loss function used is as follows:

$$\ell_{mcd} = 1 - \frac{2}{K} \sum_{K} \frac{\sum_{i} u_i^k v_i^k}{\sum_{i} u_i^k + \sum_{i} v_i^k} \tag{1}$$

where K is the number of classes, k is the k^{th} class, and i is the i^{th} pixel, u is the predicted soft-max probability for the k^{th} class, and v is the one hot encoded ground truth of the corresponding k^{th} class.

Since medical image datasets are generally smaller compared to other semantic segmentation tasks and the U-Net is a fairly deep network, there arises the issue of overfitting. This is tackled by data augmentation techniques - as of now we are using 90-degree rotation, mirroring, and 45-degree rotation. Furthermore, we add Gaussian noise to every training image, with a $\mu = 0$ and a $\sigma = 0.1$ so as to improve the generalization ability of the network on unseen data.

3 Results

We generate the predicted segmentation labels by thresholding our output probability map for the 125 validation cases and 166 test cases with 0.5 as the threshold. All the segmentations are generated using an ensemble of the 5 models generated by the 5 fold cross validation by using the process of voxel-wise majority voting. The results on the validation and test dataset are shown below (Tables 1, 2 and Figs. 3, 4):

Table 1. Results on BraTS 2019 validation data

	Dice			Haus		
	Enh.	Whole.	Core.	Enh.	Whole.	Core.
Mean	0.66677	0.85269	0.70912	7.27002	8.07931	9.57081
Std. Dev	0.29306	0.15486	0.27646	12.50261	13.56434	12.50147
Median	0.77811	0.90147	0.82969	2.44949	3.60555	4.89898

Table 2. Results on BraTS 2019 test data

	Dice			Haus		
	Enh.	Whole.	Core.	Enh.	Whole.	Core.
Mean	0.697289	0.828745	0.772904	25.56391	14.64394	26.69969
Std. Dev	0.239947	0.189408	0.271199	88.55071	57.10659	88.31207
Median	0.77039	0.894235	0.88047	2.23607	3.60555	3

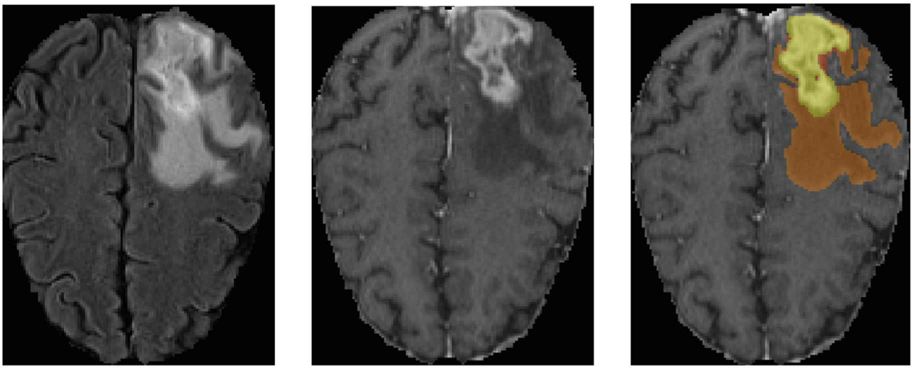

Fig. 3. This is the qualitative result of one of validation cases BraTS19_CBICA_AAM_1 and the corresponding segmentation. Left: Flair, Center: T1ce, Right: Predicted segmentation superimposed over T1ce. Orange is edema, yellow is enhancing tumor and red is necrosis. (Color figure online)

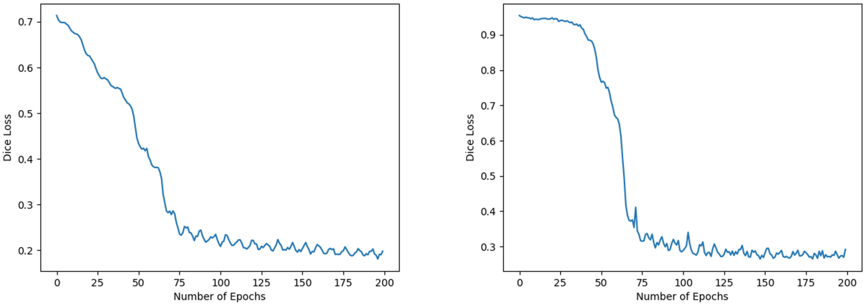

Fig. 4. Left: Training dice loss vs Number of epochs, Right: Validation dice loss vs Number of epochs

4 Discussion

Our model has the potential to provide better results with further improvements in our training process. Below we are listing some of the modifications which we are going to make to the existing process for improvement performance, in the future.

4.1 Post-processing

We further plan to use Isensee et al.'s [4] post-processing technique to improve the dice score of the enhancing tumor. It is known that LGG patients may have no enhancing tumor at all, hence even a single false-positive voxel prediction would make the ET dice score of that patient 0. To overcome this predicament we calculate an experiment-specific voxel threshold for enhancing tumor region. If the number of enhancing tumor voxels are less than this threshold we set all of them as necrosis, and if not, we leave it as it is. This threshold is calculated by mean-dice optimization over the BraTS 2019 training data.

4.2 Modification of Loss Functions

Though the Dice loss is an excellent metric to measure performance and for optimization purposes, it penalizes the False Positive and False Negative terms equally, but in reality, for medical image segmentation, a false negative term is much more dangerous than a false positive term since it means that the disease has not been detected at a place where it should have been. This issue can be addressed by using the Tversky Loss Function [14], which penalizes the false negative terms more than the false positive ones, and therefore improves performance. The mathematical expression for the Tversky Loss is given below:

$$\ell_{tv} = 1 - \frac{1}{K} \sum_{K} \frac{\sum_{i} u_i^k v_i^k}{\alpha \sum_{i} u_i^k + \beta \sum_{i} v_i^k} \tag{2}$$

All the terms are the same as Multi-class dice loss, except the additional parameters α and β, where $\alpha + \beta = 1$, and $\beta > 1$. The $\beta > 1$ ensures that the FN terms are penalized more than the FP ones. The Multi-class dice loss is a special case of the Tversky loss where $\alpha = \beta = 0.5$. Also, cross entropy loss could be used in conjunction with either the dice loss or the Tversky loss.

4.3 Architectural Modifications

Architectural modifications such as incorporation of inception modules instead of simple convolutional layers have the potential to improve the performance. Inception modules perform convolutions in-parallel with different filter sizes on the input feature map. Paralleled convolutions also help in reduction of bottlenecking of features as compared to convolutions performed sequentially, since multi-scale features can be detected by different filter sizes. The outputs of these parallel paths are then "depth"-concatenated.

5 Conclusions

From the experiments that were done, it can be concluded that, architectural modifications in the traditional 3-D U-Net do not significantly improve its performance on the Brain Tumor Segmentation dataset. Novel training strategies and procedures have a better potential to improve performance.

References

1. Kamnitsas, K., et al.: Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. MIA **36**, 61–78 (2017)
2. Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., Maier-Hein, K.H.: Brain tumor segmentation and radiomics survival prediction: contribution to the BRATS 2017 challenge. In: Crimi, A., Bakas, S., Kuijf, H., Menze, B., Reyes, M. (eds.) BrainLes 2017. LNCS, vol. 10670, pp. 287–297. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75238-9_25
3. Li, X., Chen, H., Qi, X., Dou, Q., Fu, C.-W., Heng, P.A.: H-DenseUNet: hybrid densely connected UNet for liver and liver tumor segmentation from CT volumes. arXiv preprint arXiv:1709.07330 (2017)
4. Isensee, F., Jaeger, P.F., Full, P.M., Wolf, I., Engelhardt, S., Maier-Hein, K.H.: Automatic cardiac disease assessment on cine-MRI via time-series segmentation and domain specific features. In: Pop, M., et al. (eds.) STACOM 2017. LNCS, vol. 10663, pp. 120–129. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75541-0_13
5. 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
6. Bakas, S., et al.: Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Nat. Sci. Data **4**, 170117 (2017)
7. Bakas, S., et al.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-GBM collection. The Cancer Imaging Archive 286 (2017)
8. Bakas, S., et al.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-LGG collection. The Cancer Imaging Archive (2017)
9. Menze, B.H., Jakab, A., Bauer, S., et al.: The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Trans. Med. Imaging **34**(10), 1993–2024 (2015). https://doi.org/10.1109/TMI.2014.2377694
10. 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)
11. 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
12. Arora, R., Basu, A., Mianjy, P., Mukherjee, A.: Understanding deep neural networks with rectified linear units. arXiv preprint arXiv:1611.01491 (2016)
13. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: the missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
14. Salehi, S.S.M., Erdogmus, D., Gholipour, A.: Tversky loss function for image segmentation using 3D fully convolutional deep networks. In: Wang, Q., Shi, Y., Suk, H.-I., Suzuki, K. (eds.) MLMI 2017. LNCS, vol. 10541, pp. 379–387. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67389-9_44