Improving Boundary Classification for Brain Tumor Segmentation and Longitudinal Disease Progression

Ramandeep S. Randhawa[1(✉)], Ankit Modi[2],
Parag Jain[3], and Prashant Warier[2]

[1] University of Southern California, Los Angeles, USA
ramandeep.randhawa@marshall.usc.edu
[2] Fractal Analytics, Mumbai, India
[3] Dhristi Inc., Palo Alto, USA

Abstract. Tracking the progression of brain tumors is a challenging task, due to the slow growth rate and the combination of different tumor components, such as cysts, enhancing patterns, edema and necrosis. In this paper, we propose a Deep Neural Network based architecture that does automatic segmentation of brain tumor, and focuses on improving accuracy at the edges of these different classes. We show that enhancing the loss function to give more weight to the edge pixels significantly improves the neural network's accuracy at classifying the boundaries. In the BRATS 2016 challenge, our submission placed third on the task of predicting progression for the complete tumor region.

Keywords: Deep neural networks · Segmentation · Loss functions · Glioblastoma

1 Introduction

Accurate quantification of gross tumor volumes of brain tumor is an important factor in the assessment of therapy response in patients — it is also important to quantify the volume of the different tumor components, e.g., cysts, enhancing patterns, edema and necrotic regions. In particular, identifying the edges of these tumor components and observing their evolution over time is critical to an accurate assessment of disease progression. Multi-modal MRI is often used to detect, monitor and quantify this progression [1].

Most automatic segmentation models use traditional machine learning approaches — features are manually defined and fed to a classifier, and the algorithms focus on learning the best weights for the classifier. Over the past couple of years, deep learning models have enabled automatic learning of features in addition to the weights used for classification. Several methods using deep neural networks (DNNs) for brain tumor segmentation have already been proposed [2–6]. Our work builds upon the work by Pereira et al. [4] which uses a

A. Crimi et al. (Eds.): BrainLes 2016, LNCS 10154, pp. 65–74, 2016.
DOI: 10.1007/978-3-319-55524-9_7

DNN that comprises a combination of convolutional and fully connected layers, where the convolutional layers have small 3 × 3 kernels and max-pooling layers.

In addition to standard accuracy measures such as dice scores that determine classification accuracy across the entire segment, we focus our efforts to improve performance at the boundaries between segments. To this end, we propose a pixel-wise weighted cross-entropy loss function, where pixels that are at the boundary of different classes are given more weight and hence the DNN learns to classify them better. We find that incorporating such a weighted function improves the performance of the DNN by 1.4–4.5% measured as an average of out-of-sample dice scores for each of the three regions of interest. It also lowers the standard deviation of the out-of-sample dice scores by 4–18% for each of the three regions. Further, visual inspections of the DNN's predictions also show that our approach leads to much better classification of the tumor at the boundaries between different regions.

2 Dataset

The training dataset of BRATS 2015 [1] comprises brain MRIs for 274 patients — each MRI scan has four modalities: *T1*, *T1c*, *T2*, and *FLAIR*. All images are of dimension 240 × 240 × 155 voxels. All images are already aligned with the *T1c* modality and are skull stripped. Ground truth is provided for each voxel in terms of one of 5 labels: non-tumor, necrosis, edema, non-enhancing tumor and enhancing tumor. Three tumor regions of interest are defined in this problem:

- Complete tumor region that includes all tumor voxels.
- Core tumor region that includes all tumor voxels except *edema*.
- Enhancing tumor region that consists of only the enhancing tumor voxels.

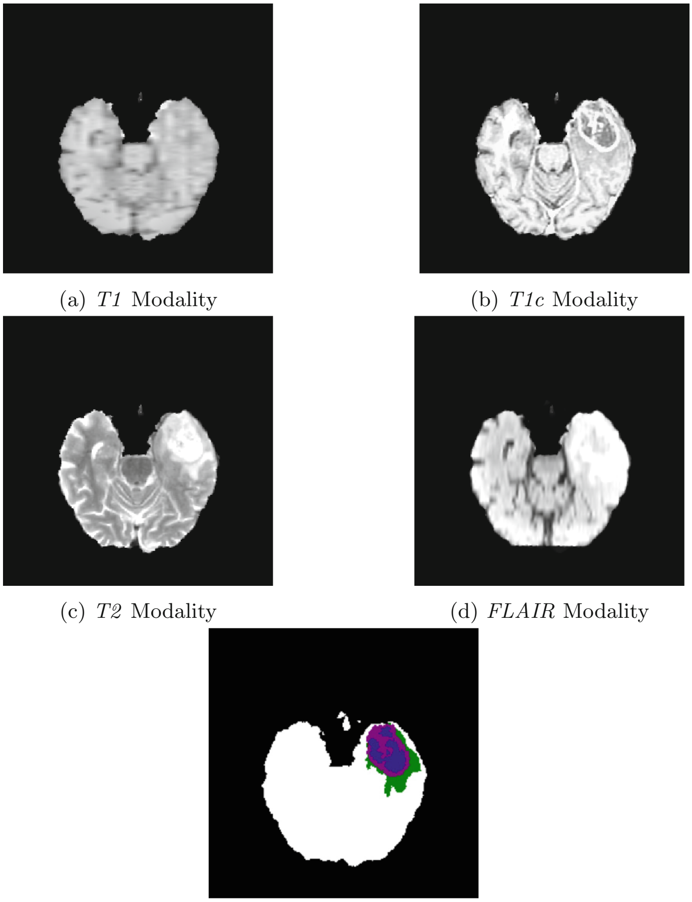
We measure accuracy using the Dice score, which measures the overlap between the ground truth and predictions over each region of interest, and is formally defined as:

$$Dice\ Score = \frac{|P_1 \cap T_1|}{\frac{1}{2}(|P_1| + |T_1|)},$$

where P_1 and T_1 denote the predicted and ground-truth positives, respectively. We compute three dice scores, one for each tumor region: complete, core, and enhancing.

3 DNN Architecture for MRI Segmentation

DNN Architecture. For our submission to the challenge, we used the same DNN proposed by Pereira et al. [4], whose network architecture is illustrated in Fig. 2. This DNN takes a 2-dimensional patch-based approach. That is, each patient's MRI is converted to 155 axial slices, and within each slice, every pixel's class is predicted by taking as input a 33 × 33 patch around the pixel, and combining this across all four modalities. Thus, the input to the DNN has dimensionality



(a) *T1* Modality

(b) *T1c* Modality

(c) *T2* Modality

(d) *FLAIR* Modality

(e) Ground truth

Fig. 1. (a–d) Different modalities of a sample MRI (axial slice) and (e) ground-truth: the tumor region is most clearly identified on the *T1c* modality. The label-color mapping for the ground-truth is: necrosis (blue), edema (green), non-enhancing tumor (red), enhancing tumor (pink). (Color figure online)

$4 \times 33 \times 33$, and the output is the class of the pixel: non-tumor, necrosis, edema, non-enhancing or enhancing tumor. The network uses 8 layers, and has about 28 million parameters.

We also considered two other DNN architectures (U-net [7] and Tri-planar version of the above DNN). However, as we discuss in Sect. 5, the above described DNN (Fig. 2) dominated both these architectures, and so we focused our study on it.

Pixel-Wise Weighted Loss Function. In order to improve accuracy of prediction around the edges, we modify the cross-entropy loss function to weigh pixels based

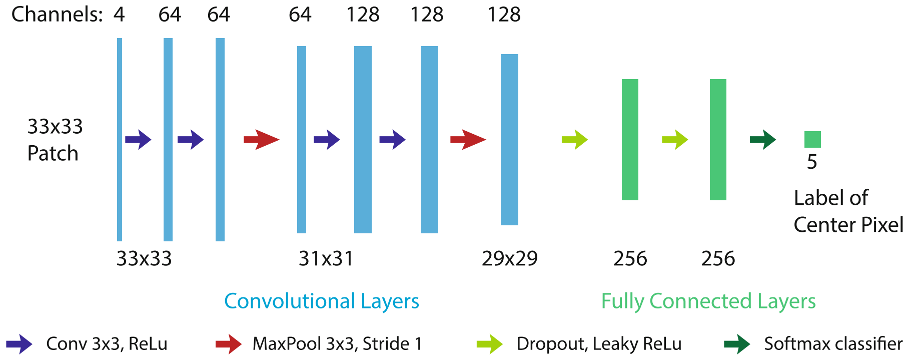


Fig. 2. Network architecture

on their proximity to pixels of other classes. For a mini-batch of M patches, our proposed training loss function is:

$$Loss = -\sum_{i=1}^{M} w_i \log p_i, \tag{1}$$

where p_i is the predicted probability for the correct label of patch i and the weight for each patch w_i is given by:

$$w_i = \frac{N + \#\text{ patch-}i\text{ pixels with label different from center pixel}}{N + \#\text{ patch-}i\text{ pixels with label same as center pixel}} \tag{2}$$

If we set the weight $w_i = 1$ for all patches, then (1) reduces to the regular cross-entropy loss function. Intuitively, identifying the label of the center pixel of a patch correctly would be easier if its neighboring pixels are also of the same label. This implies that the outer edge pixels of regions of a particular label would be more difficult to segment than pixels that lie in the interior of such regions. The weights w_i measure this difficulty by computing a ratio of the number of pixels in the patch with labels different from the center pixel to the number of pixels in the patch with the same label as the center pixel. The weights w_i weight the "more difficult" patches that have more pixels in the patch that are different from the center pixel, higher than patches that are "easier." The constant N is a "smoothing" hyper-parameter that is added to ensure the weights remain bounded in a reasonable range so that the DNN learns consistently from all patches, and is only slightly leaned toward the difficult patches.

We also add L_1- and L_2-regularization to the loss function to prevent overfitting.

Pre-processing. We perform very limited pre-processing. In particular, we perform $N4$ bias correction on the *T1* and *T1c* modalities. We transform each input channel on a per image basis by first thresholding intensities lower than 1-percentile and greater than 99-percentile and then normalizing all values to have zero mean and unit standard deviation.

Training and Testing. We train the network using a combination of High-Grade Glioblastoma (HGG) and Low-Grade Glioblastoma (LGG) images. We train using 243 MRIs, and reserve the remaining 27 MRIs for out-of-sample testing (approximately 90/10 split). We maintain the same training and testing datasets throughout the experiments.

Our dataset is highly skewed: with most voxels healthy (approximately 92.4%), and only few with tumors. In particular, we had approximately 0.4% necrosis, 5.1% edema, 0.7% non-enhancing tumor and 1.3% enhancing tumor on average. When dealing with such skewed data, a common practice is to perform two-stage training [2]. In this, the first-stage is an equiprobable stage in which the patches are sampled from the training dataset in a manner so that each label (non-tumor, necrosis, edema, enhancing tumor and non-enhancing tumor) is equally-likely to be chosen. The second-stage is a fine-tuning stage, in which the patches are sampled according to the actual distribution with which they occur in the dataset, but the training is used to train only the fully-connected layers of the DNN. That is, the higher convolutional layers are fixed, or frozen, during the fine-tuning stage. Intuitively, the convolutional layers build a latent representation of the patch, and the fully-connected layers use this latent representation to classify the patch. The equiprobable training phase exposes the convolutional layers to patches of all labels to help build better latent representations, and the fine-tuning phase then helps in refining the classification ability of the fully connected layers.

We train the equiprobable phase for 500 epochs, and the fine-tuning phase for 250 epochs. The overall training takes about 9 h, and segmenting an MRI image takes about 6 min. For our submission to the BRATS 2016 challenge, we used an ensemble of three such trained nets.

When comparing the baseline DNN (with regular cross-entropy loss function) with the (pixel-wise) weighted DNN, we use all the same hyper-parameters except N. We also used data-augmentation by flipping the input patches both horizontally and vertically.

Post-processing. As in [4], we remove small regions of voxels with predicted tumor (of any label) below a cumulative voxel size of certain threshold. We found that a threshold of 3,000 voxels worked best.

4 Results

Table 1 compares the dice scores for the three regions as described in Sect. 2 for the test data for the proposed DNN trained with (pixel-wise) weighted loss function ($N = 10,000$) and the baseline DNN, in which the usual cross-entropy loss function is used. We observe that when using the DNN trained with the weighted loss function, the average dice score improves for all three regions: complete, core and enhancing (tumor). The improvement is the highest for the core region. Further, the standard deviation of the dice scores across all images is lower when using the weighted loss function. For the complete and core regions,

Table 1. Test-set dice scores reported as mean (standard deviation). DNN with weighted loss function ($N = 10,000$) has a higher mean dice score, and lower standard deviation compared with the non-weighted baseline DNN.

DNN	Complete	Core	Enhancing
Baseline	0.85 (0.11)	0.72 (0.22)	0.70 (0.26)
Weighted	**0.87** (0.09)	**0.75** (0.19)	**0.71** (0.25)

the performance improvement was statistically significant with $p = .038$ and $p = .030$, respectively; for the enhancing region, we obtained $p = .080$, which suggests that with a larger test set, the performance improvement could potentially be statistically established.

Table 2 displays the same results separated by the glioblastoma grade (HGG and LGG). We observe that using the weighted loss function leads to consistent improvement in the dice score in all cases. The improvement is in fact higher for the LGG images. We also notice that the dice scores are quite low for both methods for the enhanced tumor in LGG images (some LGG images have no enhanced tumor, which leads to a low dice score for the prediction).

Table 2. Dice scores for test images separated by tumor grade (HGG and LGG).

DNN	Complete	Core	Enhancing
HGG			
Baseline	0.86	0.73	0.78
Weighted	**0.87**	**0.76**	**0.79**
LGG			
Baseline	0.83	0.65	0.36
Weighted	**0.86**	**0.68**	**0.37**

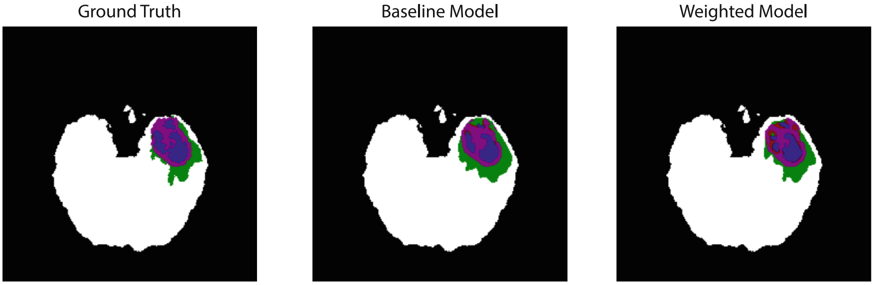


Fig. 3. Groundtruth, prediction without weights and with weights ($N = 10,000$). The images depict 5 classes; the label-color mapping is: necrosis (blue), edema (green), non-enhancing tumor (red), enhancing tumor (pink). (Color figure online)

Figure 3 displays the ground truth, predictions of the baseline and weighted models for the image slice of Fig. 1. Looking at the predictions of the baseline model, we see that there are a large number of misclassifications at the edges of the various regions. The pixel-wise weighted loss function is quite successful at correctly classifying the edge pixels. We do see some misclassifications between the blue (necrosis) and pink (enhancing tumor) regions, which may explain the fact that for the enhancing region, the dice score is only marginally better for the weighted model.

We also compare the specificity and sensitivity of both methods. Tables 3 and 4 displays these results across all images, and separated by grade. We observe that the specificity is marginally higher for the DNN with pixel-wise weighted loss function with the most significant improvement for LGG complete tumor region. For sensitivity, the results are mixed. Across all images, the sensitivity is slightly higher for the DNN with pixel-wise weighted loss function for core and enhancing regions, but slightly lower for complete tumor. When segregating the

Table 3. Specificity scores for test images separated by tumor grade (HGG and LGG).

DNN	Complete	Core	Enhancing
All			
Baseline	0.987 (0.010)	0.997 (0.002)	0.997 (0.002)
Weighted	**0.992** (0.006)	0.997 (0.003)	0.997 (0.002)
HGG			
Baseline	0.988	0.997	0.997
Weighted	**0.991**	0.997	0.997
LGG			
Baseline	0.984	0.996	0.999
Weighted	**0.993**	0.996	0.999

Table 4. Sensitivity scores for test images separated by tumor grade (HGG and LGG).

DNN	Complete	Core	Enhancing
All			
Baseline	**0.86** (0.13)	0.70 (0.24)	0.78 (0.23)
Weighted	0.84 (0.13)	**0.71** (0.22)	**0.80** (0.21)
HGG			
Baseline	0.75	0.71	0.80
Weighted	**0.83**	**0.73**	**0.82**
LGG			
Baseline	**0.89**	**0.71**	**0.73**
Weighted	0.84	0.66	0.69

images by glioblastoma grade, we see that DNN with pixel-wise weighted loss function performs better for all regions for HGG images, but worse for all regions for LGG images. These results suggest that there may be benefits to training separate networks for HGG and LGG images.

5 Discussion

Selecting N. For our results in the previous section, we used $N = 10,000$, which is approximately ten times the number of pixels per patch. With this choice of N, the weights per pixel range from 0.9 to 1.1. It is interesting that such a small change in weights leads to the improved performance in the DNN. We tried various other values for the hyper-parameter N (as reported in Table 5). We also tried $N = 100$, however this performed quite poorly. The table also provides, as a reference, the baseline case which can be considered as setting $N = \infty$.

Table 5. Dice scores for different values of hyper-parameter N.

N	Complete	Core	Enhancing
1,000	0.85 (0.13)	**0.75** (0.19)	**0.72** (0.25)
10,000	**0.87** (0.09)	**0.75** (0.19)	0.71 (0.25)
20,000	0.86 (0.10)	**0.75** (0.18)	0.71 (0.25)
100,000	0.85 (0.11)	0.73 (0.19)	0.70 (0.25)
Baseline	0.85 (0.11)	0.72 (0.22)	0.70 (0.26)

Value of Two-Stage Training. As mentioned earlier, we perform training of the DNN in two stages. To understand the value of the fine-tuning phase, we computed the dice scores for the DNN (weighted) after only completing the equiprobable training phase. In this case, we obtained dice scores of: 0.77 (complete), 0.65 (core) and 0.60 (enhanced). Comparing this with the dice scores in Table 1, we see that there is significant benefit to the fine-tuning phase of training: about 10% for complete region, 15% for core region, and 20% for the enhanced region.

Comparison with Other Architectures. Before embarking on our study we compared three different DNN architectures (with the regular cross-entropy loss function), to pick the best candidate for studying the effect of introducing the pixel-wise weighted loss function. In particular, we considered a tri-planar version of the DNN displayed in Fig. 2, in which for each patch, the sagittal and coronal slices containing the center pixel of interest were also used as input (so the input was 12 channels instead of 4); such an architecture has been referred to as $2.5D$ [8,9]. We also considered the popular U-net architecture [7]. For this we did not obtain good results for direct segmentation, but instead we trained three different networks, one for each region of interest, in a one-versus-rest fashion. Table 6 displays the results. Our baseline DNN clearly dominates the tri-planar

Table 6. Comparison of different unweighted architectures.

DNN	Complete	Core	Enhancing
Baseline	0.85 (0.11)	**0.72** (0.22)	**0.71** (0.26)
U-net	**0.87** (0.08)	0.63 (0.19)	0.60 (0.28)
Tri-planar	0.77 (0.22)	0.65 (0.27)	0.60 (0.29)

architecture. Turning to the U-net, it performs slightly better than our baseline architecture for the complete tumor region, but performed worse for the core and enhancing regions. Thus, put together with the fact that the U-net was trained in a one-versus-rest fashion for each region, we chose to proceed our study with the baseline architecture. The lower performance of the U-net was surprising, and would make for an interesting future study.

6 Conclusions

In this paper, we propose a pixel-wise weighted loss function that focuses on improving classification accuracy at edges of regions of different labels. This loss function is a modification of the traditional cross-entropy loss function that gives more weight to pixels that are surrounded by a large number of pixels of different labels. Our out-of-sample results show that a small such modification (with weights ranging from 0.9–1.1) improves the performance of the DNN by 1.5–4.5% on average. In the BRATS 2016 challenge, our submission placed third on the task of predicting progression for the complete tumor region.

References

1. 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 Trans. Med. Imaging **34**(10), 1993–2024 (2015)
2. Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.-M., Larochelle, H.: Brain tumor segmentation with deep neural networks. Med. Image Anal. **35**, 18–31 (2016)
3. Havaei, M., Dutil, F., Pal, C., Larochelle, H., Jodoin, P.-M.: A convolutional neural network approach to brain tumor segmentation. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Handels, H. (eds.) BrainLes 2015. LNCS, vol. 9556, pp. 195–208. Springer, Cham (2016). doi:10.1007/978-3-319-30858-6_17
4. Pereira, S., Pinto, A., Alves, V., Silva, C.A.: Deep convolutional neural networks for the segmentation of gliomas in multi-sequence MRI. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Handels, H. (eds.) BrainLes 2015. LNCS, vol. 9556, pp. 131–143. Springer, Cham (2016). doi:10.1007/978-3-319-30858-6_12
5. Urban, G., Bendszus, M., Hamprecht, F., Kleesiek, J.: Multi-modal brain tumor segmentation using deep convolutional neural networks. In: proceedings of the BRATS-MICCAI (2014)

6. Zikic, D., Ioannou, Y., Brown, M., Criminisi, A.: Segmentation of brain tumor tissues with convolutional neural networks. In: Proceedings of the MICCAI-BRATS, pp. 36–39 (2014)
7. 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). doi:10.1007/978-3-319-24574-4_28
8. Rao, V., Sarabi, M.S., Jaiswal, A.: Brain tumor segmentation with deep learning. In: MICCAI BraTS (Brain Tumor Segmentation) Challenge, pp. 31–35 (2014)
9. Shin, H.-C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D.J., Summers, R.M.: Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. CoRR abs/1602.03409 (2016). http://dblp.uni-trier.de/rec/bib/journals/corr/ShinRGLXNYMS16