Top 10 BraTS 2020 challenge solution: Brain tumor segmentation with self-ensembled, deeply-supervised 3D-Unet like neural networks.

Théophraste HENRY*[1][0000-0003-0672-415X], Alexandre CARRE*[1] [0000-0002-4793-835X], Marvin LEROUSSEAU[1], Théo ESTIENNE[1], Charlotte ROBERT[1], Nikos PARAGIOS[2], Eric DEUTSCH[1]

[1] Université Paris-Saclay, Gustave Roussy, Inserm, Radiothérapie Moléculaire et Innovation Thérapeutique, 94800, Villejuif, France.
[2] TheraPanacea, Paris, France
* Contributed equally

Abstract. Brain tumor segmentation is a critical task for patient's disease management. To this end, we trained multiple U-net like neural network, mainly with deep supervision and stochastic weight averaging, on the Multimodal Brain Tumor Segmentation Challenge (BraTS) 2020 training dataset, in a cross-validated fashion. Final brain tumor segmentations were produced by first averaging independently two sets of models, and then custom merging the labelmaps to account for individual performance of each set. Our performance on the online validation dataset with test time augmentation were as follows: Dice of 0.81, 0.91 and 0.85; Hausdorff (95%) of 20.6, 4,3, 5.7 mm for the enhancing tumor, whole tumor and tumor core, respectively. Similarly, our ensemble achieved a Dice of 0.79, 0.89 and 0.84, as well as Hausdorff (95%) of 20.4, 6.7 and 19.5mm on the final test dataset. More complicated training schemes and neural network architectures were investigated, without significant performance gain, at the cost of greatly increased training time. While relatively straightforward, our approach yielded good and balanced performance for each tumor subregions. Our solution is open sourced at https://github.com/lescientifik/xxxxx.

Keywords: Deep Learning, Brain Tumor, Semantic Segmentation.

1 Introduction

1.1 Clinical overview

Gliomas are the most frequent primitive brain tumors in adult patients and exhibits various degrees of aggressiveness and prognosis. Magnetic Resonance Imaging (MRI) is required to fully assess tumor heterogeneity, and the following sequences are conventionally used: T1 weighted sequence (T1), T1-weighted contrast enhanced sequence using gadolinium contrast agents (T1Gd), T2 weighted sequence (T2), and fluid attenuated inversion recovery (FLAIR) sequence.

Four distinct tumoral subregions can be defined from MR imaging: the “enhancing tumor” (ET) which corresponds to area of relative hyperintensity in the T1Gd with respect to the T1 sequence; the “non enhancing tumor” (NET) and the “necrotic tumor” (NCR) which are two different histological entities, but exhibits very similar imaging characteristics: both are hypo-intense in T1-Gd when compared to T1; and finally the “peritumoral edema” (ED) which is hyper-intense in FLAIR sequence. These almost homogeneous subregions can be then clustered together to compose three “semantically” meaningful tumor subparts: ET is the first cluster, addition of ET, NET and NCR creates the “tumor core” (TC) region, and addition of ED to TC creates the “whole tumor” (WT). Example of each sequence and tumor subvolumes is provided in Figure 1 using 3D Slicer [1].

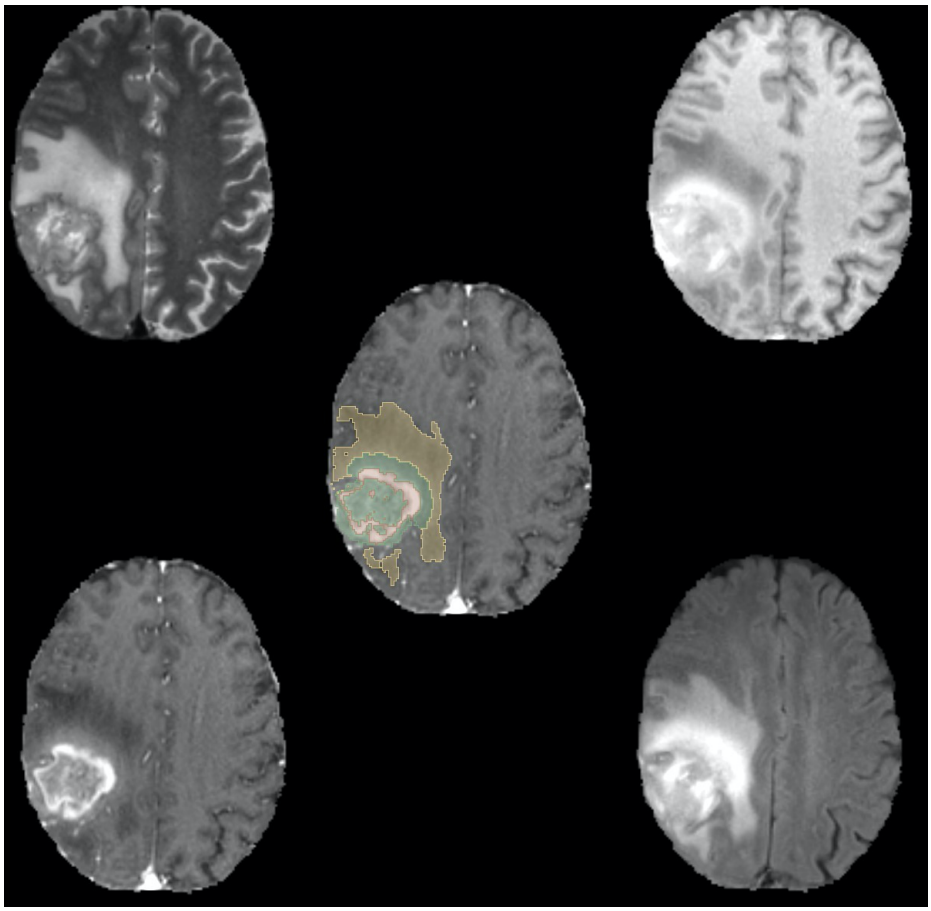

Fig. 1. Example of a brain tumor from the BraTS 2020 training dataset. **Red**: enhancing tumor (ET), **Green**: non enhancing tumor/ necrotic tumor (NET/NCR), **Yellow**: peritumoral edema (ED). **Upper Left:** T2 sequence, **Upper Right**: T1 sequence, **Lower Left**: contrast enhanced T1 sequence, **Lower Right**: FLAIR sequence **Middle:** contrast enhanced T1 sequence with labelmap overlay

Accurate delineation of each tumor subregion is critical to patient's disease management. Indeed, the radiation oncologist is required to segment the tumor, which will be the cornerstone of the radiation treatment plan that will be delivered. Correct segmentation could also unveil prognosis factors through the use of radiomics or deep-learning based approach [2].

1.2 Multimodal Brain Tumor Segmentation challenge 2020

The Multimodal Brain Tumor Segmentation Challenge 2020 [3–7] was split in three different tasks: segmentation of the different tumor sub-regions, prediction of patient overall survival (OS) from pre-operative MRI scans, and evaluation of uncertainty measures in segmentation.

The Segmentation challenge consisted in accurately delineating the ET, TC and WT part of the tumor. Main evaluation metrics were an overlap measure and a distance metric. The commonly used Dice Similarity Coefficient (DSC) measures the overlap between two sets. In the context of ground truth comparison, it can be defined as follows:

$$DSC = \frac{2TP}{2TP+FP+FN} \tag{1}$$

with TP the true positives (number of correctly classified voxels), FP the false positives and FN the false negatives. It is interesting to note that this metric is insensitive to the extent of the background in the image.

The Hausdorff distance [8] is complimentary to the Dice metric, as it measures the maximal distance between the margin of each set. It penalizes greatly outliers, as a prediction could exhibit almost voxel-perfect overlap, but if a single voxel is far away from the reference segmentation, the Hausdorff distance will be high. As such, this metric can seem noisier than the Dice index, but in fact is very handy to evaluate the clinical usefulness of a segmentation. As an example, if a tumor segmentation encompasses distant healthy brain tissue, it would require manual correction from the radiation oncologist to prevent disastrous consequences for the patient, even if the overall overlap as measured by the Dice metric is good enough.

2 Methods

2.1 Generalities

Each of the first author trained multiple neural networks using independant training pipelines. After several iterations of data processing and hyperparameters tuning, we selected a common neural network architecture with minor variations (described below) but kept two separate training approaches, in order to promote network predictions' diversity. The specific details of each pipeline will be described below, and referred to as pipeline A and pipeline B.

2.2 Neural network architecture

The final network used an encoder decoder architecture, heavily inspired by the 3D-Unet architecture from Çiçek et al [9], as can be seen in Figure 2.

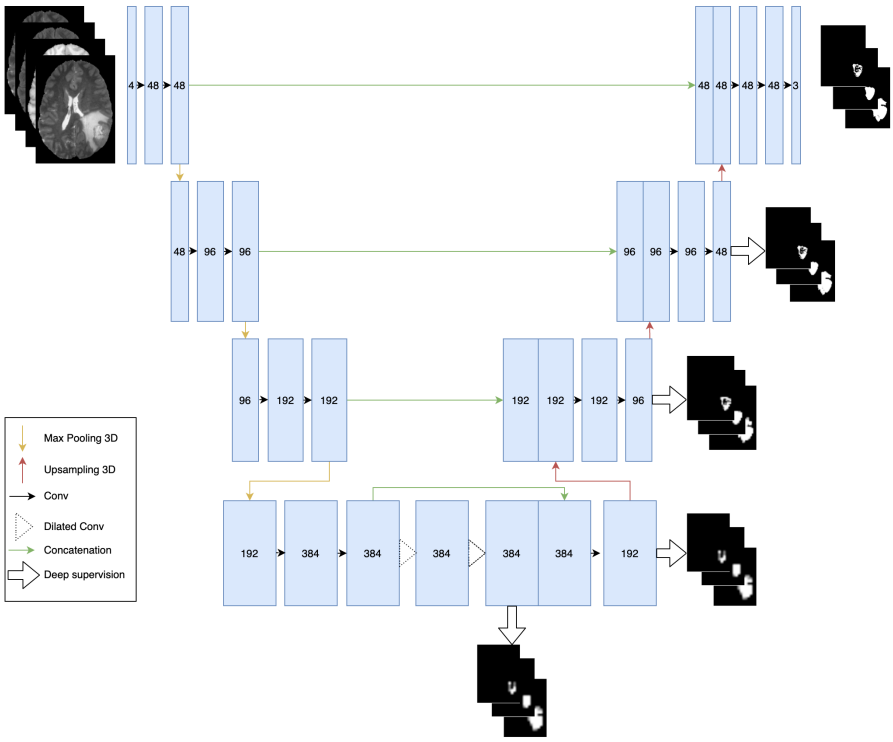

Fig. 2. Neural Network Architecture: 3D-Unet with minor modifications.

In the following description, a stage is defined as an arbitrary number of convolutions that does not change the spatial dimensions of the feature maps. All convolutions were followed by a normalization layer and a nonlinear activation (ReLU layer

[10]). Group normalization [11] (A) and Instance normalization [12] (B) were used as a replacement for Batch Normalization [13] due to a small batch size during training and good theoretical performance on non-medical datasets.

The decoder had four stages. Each stage consisted of two 3x3x3 convolutions. The first convolution increased the number of filters to the predefined value for the stage (48 for stage 1), while the second one keeps the number of output channels unchanged. Between each stage, spatial downsampling was performed by a MaxPool layer with a kernel size of 2x2x2 with stride 2. After each spatial downsampling, the number of filters was doubled. After the last stage, two 3x3x3 dilated convolutions with dilation rate = 2 were performed, and then concatenated with the last stage output.

The decoder part of the network was almost symmetrical to the encoder. Between each stage, spatial upsampling was performed using a trilinear interpolation. Shortcut connections between encoder and decoder stage that shared the same spatial sizes were performed by concatenation. The decoder stage performing at the lowest spatial resolution was made up of only one 3x3x3 convolution. Last convolutional layer used a 1x1x1 kernel with 3 output channels and a sigmoid activation.

Previous winner of the Brats challenge [14] limited their downsampling steps to 3. We hypothesized that further downsampling of the features maps, given the limited size of the input (128x128x128), would lead to irreversible loss of spatial information. As the last stage of the encoder takes much less GPU memory than the first, the dilation trick [15] was used to perform a pseudo fifth stage at the same spatial resolution as the fourth stage.

3D attention U-Nets were also trained, using the Convolutional Block Attention Module [16].

2.3 Loss Function

Inspired by the conciseness of the 2019 winning solution [14], the neural network was trained using only the Dice Loss [17] (A).The loss L is computed batch-wise and channel-wise, without weighting:

$$L = \frac{1}{n} \sum_n \frac{S*R+ \varepsilon}{S^2+ R^2+\varepsilon} \quad (2)$$

with n the number of channels, S the output of the neural network after sigmoid activation, R the ground truth label and ε a constant to prevent zero-division (set to 1 in our experiment). Similarly, optimization was made directly on the final tumor regions to predict (ET, TC and WT) and not on their components (ET, NET-NCR, ED). The neural network output was a 3-channel volume, each channel representing the probability map for each tumor regions. Pipeline B used a slighty different formulation of the Dice Loss, without squaring the denominator.

Deep supervision [18] was performed after the dilated convolutions of the bottom of the network, and after each stage of the decoder (except the last) as in [19]. Deep

supervision was achieved by adding an extra 1x1x1 convolution with sigmoid activation and trilinear upsampling. The final loss is the unweighted sum of the main output and the three auxiliary losses.

3 Training pipeline

3.1 Image pre-processing

Since MRI intensities vary depending on manufacturers, acquisition parameters, and sequences, input images needed to be standardized. Min-max scaling of each MRI sequence was performed separately, after clipping all intensity values to the 1 and 99 percentiles of the non-zero voxels distribution of the volume (A). Pipeline B performed a z-score normalization of the non-zero voxels of each IRM sequence independently.

Images were then first cropped to a variable size using the smallest bounding box containing the whole brain, and then randomly re-cropped to a fixed patch size of 128x128x128. This allowed to remove most of the useless background that was present in the original volume, and to learn from an almost complete view of each brain tumor.

3.2 Data augmentation techniques

To prevent overfitting, data augmentation techniques were used.

Pipeline A: first, each channel was randomly rescaled by a value sampled uniformly between 0.9 and 1.1. Then, each channel was noised independently with gaussian noise of zero mean and a standard deviation of 0.1 of the normalized rescaled image standard deviation. Random flip of each spatial dimension was performed. Finally, input channel dropping was done.

These data augmentations steps were performed randomly with a probability of 80%, except for channel dropping which was performed with a smaller probability of 16% (0.2x0.8).

Pipeline B: Voxel intensities were randomly noised using the following data augmentation techniques with a probability of 20% each: Gaussian noise of zero mean and 0.1 standard deviation, gamma correction in the [0.5, 2.5] range, intensity shift between [-0.1, 0.1]. Then, random flip along each spatial axis was performed with a probability of 50%.

3.3 Training details

Models were produced by a five-fold cross-validation. The validation set was only used to monitor the network performance during training, and to benchmark its performance at the end of the training procedure.

Pipeline A: For each fold, the neural network was trained for 200 epochs with an initial learning rate of 1e-4, progressively reduced by a cosine decay after 100 epochs [20]. A batch size of 1 and the Ranger optimizer [21–23] were used. A validation step

was only performed every 3 training steps to reduce total computation time. After 200 epochs, we performed a training scheme inspired from the fast stochastic weight averaging procedure [24]. The initial learning rate was restored to half its initial value (5e-5), and training was done for another 30 epochs with cosine decay. Every 3 epochs, the model weights were saved. This procedure was repeated 5 times for a total of 150 additional epochs. At the end, the saved weights were averaged, effectively creating a new "self-ensembled" model. We expected this method to produce a more generalizable model. For this part of the training, the Adam optimizer [25] was used without weight decay.

Pipeline B: The maximum number of training iterations was set to 400. The best model kept was the one with the lowest loss value on the validation set. A batch size of 3 and Adam optimizer with the following hyperparameter was used: initial learning of 1e-4, betas (0.9, 0.999), eps (1e-4) and weight decay (0.0). Cosine annealing scheduler were used to decay the learning rate.

Common: In order to train a bigger neural network, float 16 precision (FP16) was used, which reduced memory consumption, accelerate the training procedure, and may lead to extra performance [20].

The neural network was built and trained using Pytorch v1.6 (which has native FP16 training capability) on Python 3.7. The model could fit on one graphic card (GPU).

3.4 Inference

Inference was performed in two pass. The first pass created two labelmaps per case; one for each pipeline. 3 different models per fold (except one fold due to time constraint) were available for pipeline A: a 3D attention U-net version, a U-net version trained on an unfiltered version of the training dataset, and a U-net version trained on a subset of the training dataset where cases with high dice loss, even after full training from a previous run, were removed. The top two performing models per fold were chosen for ensembling. For Pipeline B, the five cross-validated models (one per fold) were ensembled. Ensembling was performed by simple predictions averaging.

First pass

For each pipeline, the initial volume is preprocessed like the training data, then cropped to the minimal brain extent, and finally zero-padded to have each of the spatial dimension divisible by 8. Test time augmentation (TTA) was done using 16 different augmentations for each of the five models generated by the cross-validation, for a total of 80 predictions per sample. We used flips, and 90-180-270 rotations only in the axial plane, as rotation in other planes led to worse performance on the local validation set. Final prediction was made by averaging the 80 predictions, using a threshold of 0.5 to binarize the prediction.

Labelmap reconstruction was then performed in a straightforward manner: ET prediction is left untouched, the NET-NEC region of the tumor was deduced from a

boolean operation between the ET label and the TC label, and similarly for the edema between the TC and the WT label (NET-NEC = TC – ET; edema = WT – TC).

Second pass

The first pass gave two labelmaps per case. Based on the online validation dataset, ensemble of models from the pipeline B were consistently above ensemble of models from the pipeline A on the whole tumour dice metric. We hypothesized that models from pipeline B were better for predicting edema. To keep the score intact on ET and TC from models A, ET and NET/NCR predicted labels had to be left untouched. Accordingly, if models A predicted background or edema and models B predicted edema or background respectively, models B predicted labels were kept.

4 Results

4.1 Online Validation dataset

Table 1 displays the results for the online validation data. Our models produced Dice metric above 0.8. for each tumor region. Our two-pass merging strategy had no impact on the ET and TC segmentation performance, while greatly improving WT segmentation. Single pass strategy already yielded good performance for all three tumor regions. Larger value of Hausdorff distance for ET is explained by the absence of this label on some cases. Consequently, predicting even one voxel of ET would lead to a major penalty for this metric.

Example of segmented tumor from the online validation set is displayed in Figure 2. While it is hard to visually discriminate best from average result, our worst generated mask showed obvious error: contrast enhanced arteries were mislabeled as enhancing tumor.

Table 1. Performance on the complete BraTs'20 Online Validation Data.

Metric (mean)	ET	WT	TC
Dice	0.80585	0.91148	0.85416
Dice (without second pass)	0.80585	0.89518	0.85415
Sensitivity	0.81488	0.91938	0.84485
Specificity	0.99970	0.99915	0.99963

Hausdorff (95%)	20.55756	4.30103	5.69298

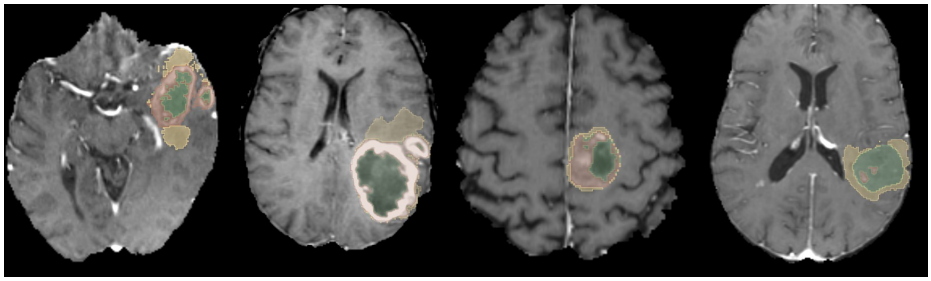

Fig. 2. From left to right: ground truth example from training set, and generated segmentations from our solution for three patients among the online validation score (respectively: best mean dice score, average mean dice score, and worst mean dice score).. **Red**: enhancing tumor (ET), **Green**: non enhancing tumor/ necrotic tumor (NET/NCR), **Yellow**: peritumoral edema (ED). It is interesting to note that both exhibit the same pattern: central non enhancing tumor core with surrounding enhancing ring and diffuse peritumoral edema.

4.2 Testing dataset

Our final results on the testing dataset can be found in table 2. This results allowed us to be in the top 10 teams for the segmentation challenge. Note however a significant discrepancy between TC Hausdorff between testing and validation dataset, while all other metrics showed small but limited overfit.

Table 2. Performance on the BraTs'20 Testing Data.

Metric (mean)	ET	WT	TC
Dice	0.78507	0.88595	0.84273
Sensitivity	0.81308	0.91690	0.85934
Specificity	0.99967	0.99905	0.99964
Hausdorff (95%)	20.36071	6.66665	19.54915

4.3 Ablation Study

Experiments with and without patients removal, and with and without attention block were produced for pipeline A. Cross-validated results based on 4 of the 5 fold can be found in Table 3 (one fold not available due to time constraint). There was no clear benefit of neither strategy, hence we decided to keep the two best available models for each fold for this pipeline.

Table 3. Ablation study. Results from cross-validation on the training set.

Dice: mean (std)	ET	WT	TC
U-Net like	0.8077 (0.011)	**0.9070** (0.006)	**0.8705** (0.013)
+ Patients removal	0.8126 (0.019)	0.9043 (0.005)	0.8686 (0.012)
+ Attention block	**0.8144** (0.022)	0.9037 (0.008)	0.8701 (0.018)

5 Discussion

Our solution to this challenge yielded good results, without much of the modern "bells and whistles" that can be used: short additive residual connections [26], dense blocks [27], more recent neural networks backbone based on inverted residual bottleneck [28], newer decoder structure like biFPN layer [29], or semi-supervised setting using brain dataset from the Medical Decathlon [30].

We tried all of these refinements, without significant improvement on the local validation set. We hypothesize that this was probably due to GPU memory constraints. Indeed, while these layers improve the model accuracy at a relatively small parameter cost, it increases significantly the size of the activation maps of the model, forcing us to use smaller networks (reduction of the number of output channels per convolutional layer). Reducing the crop size of the patch was not an option as this would have most probably reduced the network performance due to the lack of context. Moreover, all these additions yielded significant increases of the training time, reducing the searchable space in the limited timeframe of the challenge.

Stochastic weight averaging at the end of the training was the most notable refinement we used. This training scheme was a remnant from the mean teacher semi-supervised training [31]. We did not benchmark its real potential but expect it to prevent from overfitting on the training set and remembering the noisy labels. Indeed, it has been shown that a high learning rate could prevent such behavior, and we expect that our training benefit from the multiple learning rate restarts [32].

Notably, while our results were not state of the art for the BraTS 2020 challenge, the segmentation performance of our method is in the usual range of inter-rater agreement for lesion segmentation [33, 34] and could already be valuable for clinical use. As an example, Figure 3 zooms in the tumor segmentation of the first two segmentations of Figure 2 (respectively manual ground truth annotations and best validation case). The predicted label is smoother than the manually created ground truth. We let the reader make his own mind and decide which segmentation is more appealing for clinical use.

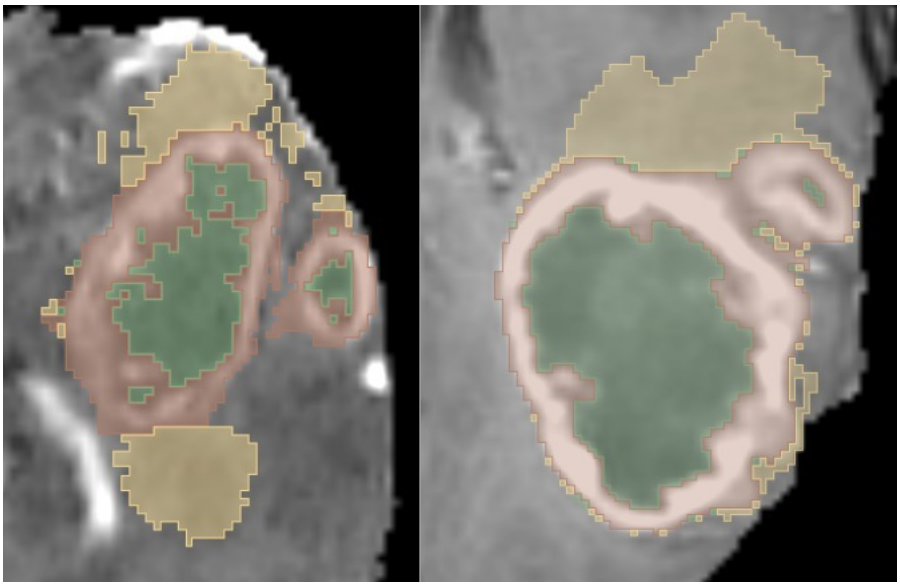

Fig. 3. Zoomed version of the first two vignettes of Figure 2. Left: ground truth example from training set. Right: generated segmentations from our solution for the best mean dice score patient on the validation set. **Red**: enhancing tumor (ET), **Green**: non enhancing tumor/ necrotic tumor (NET/NCR), **Yellow**: peritumoral edema (ED). It is interesting to note that both exhibit the same pattern: central non enhancing tumor core with surrounding enhancing ring and diffuse peritumoral edema.

6 Conclusion

The task of brain tumor segmentation, while challenging, can be solved with good accuracy using 3D-Unet like neural network architecture, with a carefully crafted preprocessing, training and inference procedure. We open-sourced our training pipeline, allowing future researchers to build upon our findings, and improve our segmentation performance.

References

1. Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J.-C., Pujol, S., et al.: 3D Slicer as an Image Computing Platform for the Quantitative Imaging Network. Magn. Reson. Imaging. 30, 1323–1341 (2012). https://doi.org/10.1016/j.mri.2012.05.001.

2. Dercle, L., Henry, T., Carré, A., Paragios, N., Deutsch, E., Robert, C.: Reinventing radiation therapy with machine learning and imaging bio-markers (radiomics): State-of-the-art, challenges and perspectives. Methods. (2020). https://doi.org/10.1016/j.ymeth.2020.07.003.

3. Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby,

J., et al.: The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). IEEE Trans. Med. Imaging. 34, 1993–2024 (2015). https://doi.org/10.1109/TMI.2014.2377694.

4. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.S., et al..: Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features. Sci. Data. 4, 170117 (2017). https://doi.org/10.1038/sdata.2017.117.

5. 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. ArXiv181102629 Cs Stat. (2019).

6. Bakas S, Akbari H, Sotiras A, Bilello M, Rozycki M, Kirby J, 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.KLXWJJ1Q.

7. Bakas S, Akbari H, Sotiras A, Bilello M, Rozycki M, Kirby J, 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.GJQ7R0EF.

8. Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J.: Comparing images using the Hausdorff distance. IEEE Trans. Pattern Anal. Mach. Intell. 15, 850–863 (1993). https://doi.org/10.1109/34.232073.

9. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. ArXiv160606650 Cs. (2016).

10. Nair, V., Hinton, G.E.: Rectified Linear Units Improve Restricted Boltzmann Machines. 8.

11. Wu, Y., He, K.: Group Normalization. ArXiv180308494 Cs. (2018).

12. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance Normalization: The Missing Ingredient for Fast Stylization. ArXiv160708022 Cs. (2017).

13. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. (2015).

14. Two-Stage Cascaded U-Net: 1st Place Solution to BraTS Challenge 2019 Segmentation Task | SpringerLink, https://link.springer.com/chapter/10.1007/978-3-030-46640-4_22, last accessed 2020/07/16.

15. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. ArXiv160600915 Cs. (2017).

16. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: Convolutional Block Attention Module. ArXiv180706521 Cs. (2018).

17. Milletari, F., Navab, N., Ahmadi, S.-A.: V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. ArXiv160604797 Cs. (2016).

18. Wang, L., Lee, C.-Y., Tu, Z., Lazebnik, S.: Training Deeper Convolutional Networks with Deep Supervision. ArXiv150502496 Cs. (2015).

19. Qin, X., Zhang, Z., Huang, C., Gao, C., Dehghan, M., Jagersand, M.: BASNet:

Boundary-Aware Salient Object Detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7471–7481. IEEE, Long Beach, CA, USA (2019). https://doi.org/10.1109/CVPR.2019.00766.

20. He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., Li, M.: Bag of Tricks for Image Classification with Convolutional Neural Networks. ArXiv181201187 Cs. (2018).

21. Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., et al.: On the Variance of the Adaptive Learning Rate and Beyond. ArXiv190803265 Cs Stat. (2020).

22. Zhang, M.R., Lucas, J., Hinton, G., Ba, J.: Lookahead Optimizer: k steps forward, 1 step back. ArXiv190708610 Cs Stat. (2019).

23. Yong, H., Huang, J., Hua, X., Zhang, L.: Gradient Centralization: A New Optimization Technique for Deep Neural Networks. ArXiv200401461 Cs. (2020).

24. Athiwaratkun, B., Finzi, M., Izmailov, P., Wilson, A.G.: There are many consistent explanations of unlabeled data: why you should average. 22 (2019).

25. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. ArXiv14126980 Cs. (2017).

26. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. ArXiv151203385 Cs. (2015).

27. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely Connected Convolutional Networks. ArXiv160806993 Cs. (2018).

28. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv170404861 Cs. (2017).

29. Tan, M., Pang, R., Le, Q.V.: EfficientDet: Scalable and Efficient Object Detection. ArXiv191109070 Cs Eess. (2020).

30. Simpson, A.L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., van Ginneken, B., et al.: A large annotated medical image dataset for the development and evaluation of segmentation algorithms. ArXiv190209063 Cs Eess. (2019).

31. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. ArXiv170301780 Cs Stat. (2018).

32. Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint Optimization Framework for Learning with Noisy Labels. ArXiv180311364 Cs Stat. (2018).

33. Chassagnon, G., Vakalopoulou, M., Battistella, E., Christodoulidis, S., Hoang-Thi, T.-N., Dangeard, S., et al.: AI-Driven quantification, staging and outcome prediction of COVID-19 pneumonia. Med. Image Anal. 101860 (2020). https://doi.org/10.1016/j.media.2020.101860.

34. Tacher, V., Lin, M., Chao, M., Gjesteby, L., Bhagat, N., Mahammedi, A., et al.: Semiautomatic Volumetric Tumor Segmentation for Hepatocellular Carcinoma: Comparison between C-arm Cone Beam Computed Tomography and MRI. Acad. Radiol. 20, 446–452 (2013). https://doi.org/10.1016/j.acra.2012.11.009.