Automatic Brain Tumor Segmentation with Domain Adaptation

Lutao Dai[1], Tengfei Li[2], Hai Shu[3], Liming Zhong[3,4], Haipeng Shen[1], and Hongtu Zhu[2]

[1] Faculty of Business and Economics, The University of Hong Kong, Pok Fu Lam, Hong Kong

[2] The Biomedical Research Imaging Center, Department of Radiology and Department of Biostatistics, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA

htzhu@email.unc.edu

[3] Department of Biostatistics, The University of Texas MD Anderson Cancer Center, Houston, TX, USA

[4] Guangdong Provincial Key Laboratory of Medical Image Processing, School of Biomedical Engineering, Southern Medical University, Guangzhou, China

Abstract. Deep convolution neural networks, in particular, the encoder-decoder networks, have been extensively used in image segmentation. We develop a deep learning approach for tumor segmentation by combining a modified U-Net and its domain-adapted version (DAU-Net). We divide training samples into two domains according to preliminary segmentation results, and then equip the modified U-Net with domain adaptation structure to obtain a domain invariant feature representation. Our proposed segmentation approach is applied to the BraTS 2018 challenge for brain tumor segmentation, and achieves the mean dice score of 0.91044, 0.85057 and 0.80536 for whole tumor, tumor core and enhancing tumor, respectively, on the challenge's validation data set.

Keywords: Confusion loss, Domain adaptation, Encoder-decoder network, Brain tumor, Segmentation

1 Introduction

Image segmentation plays an important role in the accurate diagnosis and efficient treatment of brain tumors. However, segmenting brain tumors, such as glioblastomas and gliomas, is difficult, because of poor tissue contrast, irregular shapes and various appearing locations. Moreover, manual segmentation can be very time-consuming and may have large intra/inter-expert variability. This creates a great need to develop reliable automatic approaches for brain tumor segmentation.

The Brain Tumor Segmentation (BraTS) challenge [2–4, 14] is an event to evaluate state-of-the-art methods in automating tumor segmentation on a large data set of annotated, high-grade glioblastomas and lower grade gliomas. To foster accurate segmentation, the BraTS 2018 challenge provides multimodal MRI

scans of each patient, including native T1-weighted, post-contrast T1-weighted, T2-weighted, and T2 Fluid Attenuated Inversion Recovery (FLAIR) volumes.

Modern deep convolutional networks have exhibited exceptional competitiveness in image segmentation, becoming industrial benchmarks [10, 11, 13, 17]. One widely-used is the encoder-decoder networks with U-shaped architectures, such as SegNet [1], U-Net [7] and DeconvNet [15].These networks are composed of a convolutional encoder to extract salient features, and a deconvolutional decoder to recover image details. Such architecture has advantages, including flexible input image sizes, consideration of spatial information, and an end-to-end prediction, leading to lower computational cost and higher representation power.

Despite the excellent performance in the 2017 challenge, the state-of-the-art encoder-decoder network of [10] in our model exploration still loses significant segmentation accuracy for part of the BraTS 2018 training set. This is probably because the network primarily captures the key features of well-segmented samples, but misses those of the others. From the transfer learning perspective, as in [19], if treating the well-segmented samples as samples in the "source" domain and the poor-segmented samples in the "target" domain, then the network fails to learn a domain invariant feature representation. This can hence be viewed as the so-called domain adaptation problem, which aims to match the marginal feature distributions of source and target. Inspired by the domain adaptation technique of [19], we add a domain classifier to the modified U-net of [10], together with a confusion loss to learn a domain invariant feature representation for the brain tumor segmentation task. Our proposed network with domain adaptation significantly enhances the segmentation accuracy on the validation set, with mean Dice scores 0.91044, 0.85057 and 0.80536 for whole tumor, tumor core and enhancing tumor, respectively. The scores on the final test set are 0.871, 0.788 and 0.738, respectively, where detailed comparison with all the other participants in this challenge can be found in [5].

2 Data Description

The BraTS 2018 challenge data are collected from three different resources that are denoted as "2013", "CBICA", and "TCIA", respectively. The training data set includes 20 high-grade glioma subjects (HGGs) from the group 2013, 88 HGGs from CBIC, and 102 HGGs from TCIA, and also includes 10 from 2013 and 65 from TCIA subjects with low-grade gliomas (LGG) that are less aggressive and infiltrative. Each subject has four modalities of MRI scans, including native T1-weighted (T1), post-contrast T1-weighted (T1Gd), T2-weighted (T2), and T2 FLAIR volumes. All MRI images are registered to a common template with the volume size of 240×240×155 voxels resampled to 1 mm isotropic resolution. The tumor regions are annotated into three classes: the GD-enhancing tumor (ET, labeled 4), the peritumoral edema (ED, labeled 2), and the necrotic and non-enhancing tumor (NCR/NET, labeled 1).

A validation data set of 66 subjects is also provided for each participating team, but with no HGG/LGG status or tumor labels. The final evaluation of

the segmentation approach is conducted on an independent test data set of 191 subjects.

3 Segmentation

In this section, we introduce the details of our framework for brain tumor segmentation. Our model is an ensemble of two base models: a modified U-Net and a U-Net with domain adaptation (DAU-Net). In either model, modalities are treated as channels. Domain adaptation is applied to regulate the feature representation learning process so that the extracted features are more invariant to differences between domains. We also discuss our data preprocessing and post-processing procedures that smooth and optimize the segmentation results. Fig. 1 contains the workflow illustration.

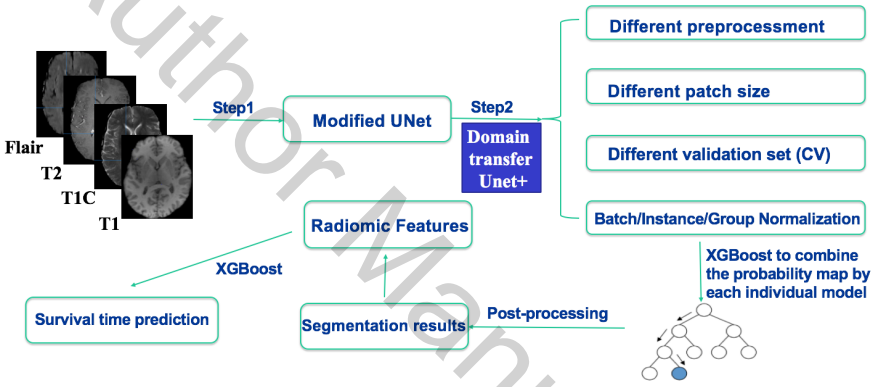

Fig. 1. Segmentation pipeline

3.1 Data Preprocessing

The main purpose of data preprocessing is to bring data to a similar distribution to avoid any initial bias, which is important for data-driven approaches. The provided data has already been skull stripped, co-registered, and resized to uniform resolution. On top of that, we remove the top and bottom 1 percentile of intensity in the brain areas for each image, and normalized the brain intensities by subtracting the mean and dividing the standard deviation. The preprocessing is conducted on brain regions only and independently across modalities and individuals.

3.2 Modified U-Net

Model Description Our modified U-Net is inspired by [10]. In our model, each level of the encoding pathway consists of a residual block with the same structure.

The first convolution layer of each residual block halves the spatial dimension with a stride of 2 (except for the first residual block), and increases the number of channels to 8×2^n, with n being the level counting from 1. As a result, the stack of 5 residual blocks progressively reduces the spatial dimension of the input tensor by a factor of 16 and learns increasingly abstract feature representations. To increase the prediction resolution, the decoding pathway progressively doubles the spatial dimension on each level by an upsampling layer of scale 2, and eventually recovers the spatial dimension of input data. The feature maps generated by the first four residual blocks are concatenated to decoding pathway of the same level to encourage the gradient flow. We apply group normalization [21] to all normalization layers, because it is more stable given a small batch and yields a better result compared to instance normalization [20]. The group number is 16 for level 1 and 32 for the remaining levels. Moreover, we adopt the idea of deep supervision [12], where output maps of different levels are combined sequentially through element-wise addition to constitute the network's final prediction via the softmax function. We integrate the multiclass dice loss function into our framework, since it can effectively mitigate the problem of class imbalance:

$$L_{\text{dice}} = -\frac{2}{|K|} \sum_{k \in K} \frac{\sum_i u_i^k v_i^k}{\sum_i u_i^k + \sum_i v_i^k}$$

where K is the set of prediction classes, u is the probability maps output by the backbone structure, v is the one-hot encoding of the ground truth, and i is the voxel index.

3.3 DAU-Net

Model Description The backbone structure is the same as the model in Subsection 3.2, except all normalization layers are instance normalization [20], because we observe more boost by domain adaptation with the presence of instance normalization. The domain classifier is appended to the end of the encoding pathway, where the feature representation is the most complex. A 1×1×1 convolution layer is first applied to significantly reduce the number of channels from 256 to 32, followed by alternating three fully-connected layers of lengths 256, 32, and 1, and two leaky ReLU rectifiers.

Domain Division We perform two sets of five-fold cross-validation on training data with the backbone structure only. One set follows the data preprocessing procedure described in Subsection 3.1, while the other set has an additional N4 bias correction processed by ANTs [18]. We compare the differences of Dice coefficient between the segmented tumors of the two sets. Although N4 bias correction has a minimal effect on most samples, it does yield significantly different results for some cases. We pick 75 subjects with the most variations and classified them to a different domain from the rest. The full list of those 75 subjects can be found in the Table 3. Based on the two domains of subjects, the network is shown in Fig. 2.

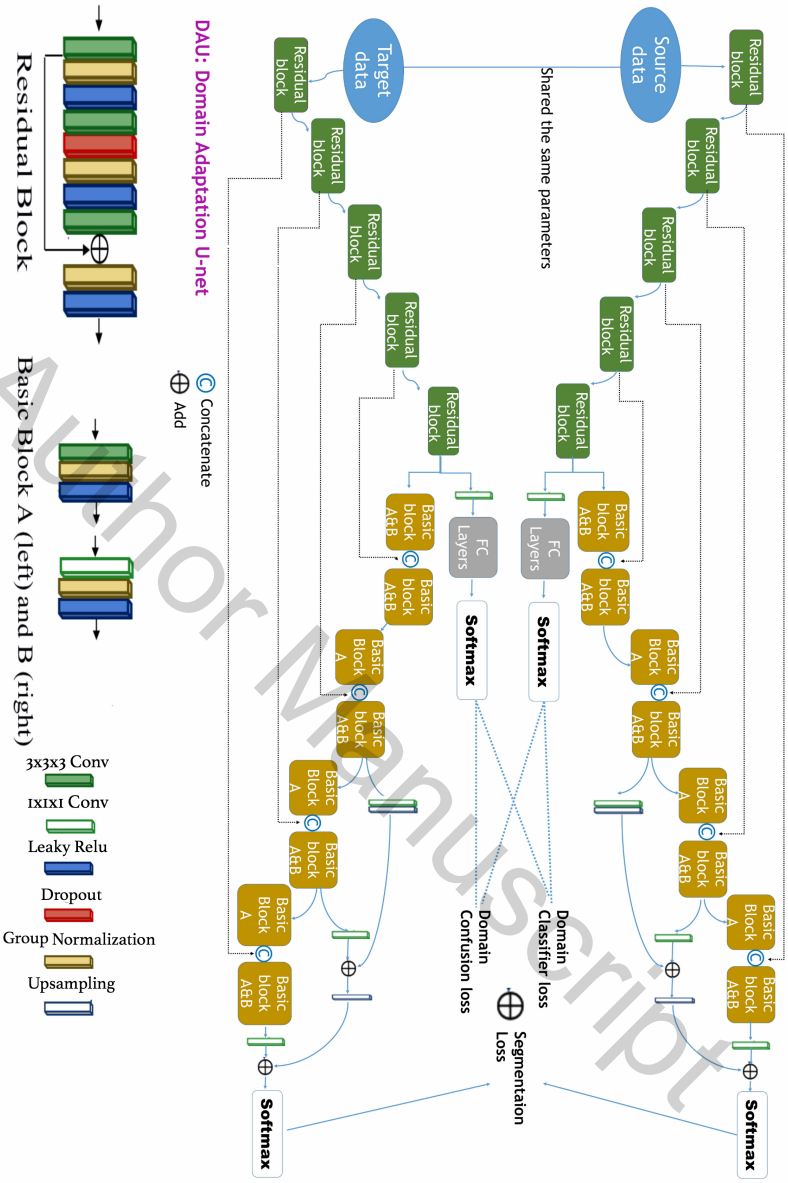

Fig. 2. Domain adaptation network architecture

Training Procedure The backbone structure and the domain classifier are trained alternatively. In the $2n$-th epoch, the objective function adds an additional confusion loss onto the dice loss function L_{dice}, which is the cross entropy between

the predicted domain label and a uniform distribution:

$$L_{2n} = -\frac{2}{|K|}\sum_{k\in K}\frac{\sum_i u_i^k v_i^k}{\sum_i u_i^k + \sum_i v_i^k} - \lambda\sum_d \frac{1}{|D|}\log q_d,$$

where D is the set of domain categories, and q_d is the estimated probability for the d-th domain from the domain classifier. q_d is modeled by a softmax function of the classifier activations $q_d = \text{softmax}(\theta_1^T f(\theta_2))$, where θ_1 includes activation parameters in the fully connected layer and θ_2 includes representation parameters in the modified U-net (Fig. 2). In this step, θ_1 is kept unchanged and parameter of the backbone structure θ_2 is updated. The hyperparameter λ controls the degree of the domain confusion relative to the backbone structure.

In the $(2n+1)$-th epoch, θ_2 is frozen so that only θ_1 is updated. The domain classifier aims to discriminate samples according to the feature representation output by the encoding pathway. The cross-entropy loss is computed with domain labels as follows:

$$L_{2n+1} = -\sum_d \mathrm{I}[y_D = d]\log q_d.$$

In summary, the two steps update different parts of parameters. By training the model iteratively, both the backbone structure and the domain classifier are optimized. The best domain classifier learned by minimizing L_{2n+1} is expected to still perform poorly on the final domain prediction, due to the confusion loss in L_{2n}. With such a domain classifier, the encoding pathway has incentives to capture the domain-invariant features. This helps to improve the generalizability of the model, since differences in MRI data representation are usually significant. The training was carried out on 4 NVIDIA Titan Xp GPU cards for about 2 days.

3.4 Experiment Configuration

The input tensors of size 128×128×128 are randomly sampled from brain areas and augmented by random flipping and transpose during each epoch. The training is implemented by PyTorch using the Adam optimizer with the learning rate initially set to be 8×10^{-4} and exponentially decaying at a rate of 0.98 every epoch for the modified U-Net, and every two epochs for the DAU-Net. All networks are trained for about 600 epochs, and the ones with the lowest Dice loss of whole tumor are selected as candidates for ensemble.

At the test time, the domain classifier is dropped. The whole brain regions whose dimensions are padded to the nearest multiple of 16 are served as inputs to the modified U-Net, and the returned segmentation maps are subsequently padded with zeros to reach the original dimension.

3.5 Model ensemble and XGBoost

XGBoost [6], short for extreme gradient boosting, is an implementation of gradient-boosted decision trees designed for speed and performance. The term gradient

boosting was proposed by Friedman[8]. It is a tree-based machine learning algorithm that usually dominates structured or tabular datasets on classification and regression predictive modeling problems, and boosting is an ensemble technique where new models are added to correct the errors made by existing models.

We aim to ensemble multiple models trained from deep learning for brain tumor segmentation using XGBoost. Most of the previous literature focused on majority voting and averaging; [9] compared the three ensembling approaches including majority voting, averaging and expectation-maximization; [22] demonstrated that the XGBoost approach can outperform the majority voting label fusion. We here train different models, which differ in preprocessing methods (with or without bias correction), different patch sizes, different splits of training dataset into training and validation parts, and different normalization (instance/group) of the network (modified U-net with group normalization and the DAU-net with instance normalization). We choose the 9 models with top Dice coefficients on the validation set; for each subject in the training dataset, we make predictions using the 9 models; next we calculate the set S of voxels where there exists disagreement for the 9 models; then, for each tumor class and each subject, we randomly choose 1000 voxels without replacement from the set S, and we use the predicted probability of each model for the four tumor classes at each voxel as covariates to predict the true label using XGBoost; to determine the hyperparameters, we split the training dataset into five parts and used a 5-fold cross-validation to optimize the maximal depth, minimum child weight, penalization parameters and learning rate, etc., to minimize the softmax loss function.

3.6 Post-processing

We employ the following post-processing techniques [22] to fill in the holes and delete the small, isolated clusters:

1. Segment the tumor mask into all connected components/clusters. Voxels in clusters whose volume is less than 0.2 times the largest connected cluster volume will be reclassified as non-tumor.
2. Segment the enhancing core mask into all connected components/clusters. Voxels in clusters whose volume is less than 0.01 times the largest connected cluster volume will be reclassified as the necrosis.
3. Fill in the holes within the tumor mask and assign voxels within the holes to necrosis area.

We find that the performance can be improved by applying post-processing on the existing results.

3.7 Survival Prediction

Based on the previous segmentation results, we crop out the tumor core region and extract various radiomic features to predict the survival time for the three

modalities Flair, T1 post-contrast, and T2. For each modality, features include: 10 intensity statistics features (such as maximum, minimum, median, and quantiles, skewness, kurtosis, and entropy), 51 shape features (24 Zernike moment based shape descriptors, 21 Hu Moment based shape descriptors, and 6 statistics of local binary patterns), 112 texture features (13 gray-level co-occurrence matrix features, 27 threshold adjacency statistics, and 72 wavelet transform features). We also add the ratio of all tumor class volumes to the whole brain volume as additional features. Similar features are used in [22].

We use XGBoost to predict survival time based on the above radiomic features, together with age. The difference between the prediction here and that in Subsection 3.5 is the dimension adopted in the survival prediction task is much higher, which will bring about overfitting and high computational complexity. We use a 5-fold cross-validation to select important features and optimize the maximal depth, minimum child weight, penalization parameters and learning rate, and other tuning parameters.

4 Results

4.1 Selection of λ

To investigate the optimum value of λ introduced in L_{2n}, we conduct multiple trials with λ as the only varying parameter (Fig. 3). Within a certain range of λ, there is a clear enhancement of the average dice coefficient for whole tumor and tumor core, whose optimum values are achieved at $\lambda = 0.1$ and $\lambda = 0.075$, respectively. Passing over the optimum point, we can see a clear decline in the average dice coefficient for both. The average dice coefficient of enhancing tumor fluctuates with λ, but its highest peak is at $\lambda = 0.1$, which is very close to the optimal λ's of whole tumor and tumor core. We hence choose $\lambda = 0.1$ for our proposed network. Detailed segmentation results for the validation set, with and without domain adaptation, are shown in Table 1, Phase 1 and Phase 2a, respectively.

4.2 Results of mean Dice score and survival prediction

The prediction results of the validation set after XGBoost and postprocessing are the best, as shown in the final row (i.e., XGBoost+Postprocessing of Table 1). According to the Wilcoxon signed-rank test, its Dice scores are significantly larger than those in phases 1 (in the 1st row), whose p-values are less than $0.048, 4 \times 10^{-5}$, and 3×10^{-4}, for ET, WT and TC, respectively. We use an example from the validation dataset to illustrate the improvement of ensemble in Fig. 4. The prediction results of the patients' survival time for the validation set are shown in Table 2.

5 Conclusion

We presented our contribution to the BraTS 2018 challenge in this paper. We developed a deep learning approach for the tumor segmentation by combining

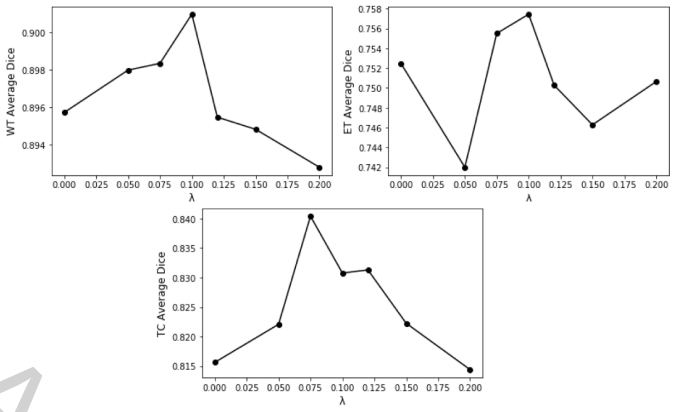

Fig. 3. Dice coefficients with varying λ.

Table 1. Segmentation results on validation data

Method	Dice_ET	Dice_WT	Dice_TC
Phase1	0.75245	0.89571	0.81561
Phase2 Model a	0.75983	0.90397	0.82489
Phase2 Model b	0.76091	0.90616	0.83622
Phase2 Model c	0.74669	0.90349	0.8278
Phase2 Model d	0.74187	0.90435	0.83211
Phase2 Model e	0.75779	0.90733	0.83824
Phase2 Model f	0.76091	0.90420	0.83713
Phase2 Model g	0.76814	0.90574	0.84704
Phase2 Model h	0.75440	0.90594	0.83826
Phase2 Model i	0.78582	0.90491	0.83689
XGBoost+Postprocessing	0.80536	0.91044	0.85057

Table 2. The prediction result on validation data

Accuracy	MSE	MedianSE	stdSE	SpearmanR
0.5	99409.107	33754.5	210658.725	0.332

a modified U-Net and the DAU-Net. Both models were trained with extensive data augmentation. We applied the XGBoost procedure to ensemble our image segmentation predictions. The ensemble of the 9 top-performing models outperformed each individual model on validation data. Due to time constraints, we did not explore the effect of domain adaptation on feature learning, which can explain the improvement on performance. How domain adaptation regulates feature learning will be a promising research topic that sheds lights on a better design of data augmentation as well as a preprocessing pipeline. Moreover, we

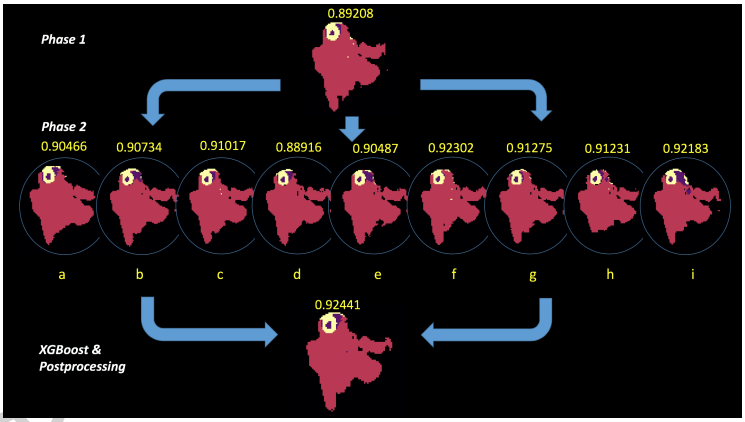

Fig. 4. Ensemble of models. Examples are from the patient "Brats18_CBICA_ALV_1". The numbers above each sub-figure are the dice coefficients of whole tumor for that patient.

tried a few ways of defining domains, including dividing the data by gliomas' grades and data source, but have not yet developed a common standard for automatic domain split with good interpretability. Keeping in mind that it is hard to train an effective domain classifier by dividing a small dataset into groups, we also expect the model to have a better result with the introduction of external data. In this case, the data naturally come from different domains. Besides, we tried different hyperparameters in our postprocessing to make improvement of Dice performance on validation dataset, which may cause overfitting.

For survival prediction, we extracted radiomic features based on the segmentation results, but did not include deep-learning features together, due to time constraints. Neural networks for predictions can usually outperform the traditional prediction methods [16], which is worth further exploration.

Acknowlegement

This research was partially supported by Ministry of Science and Technology Major Project of China 2017YFC1310903, University of Hong Kong Stanley Ho Alumni Challenge Fund, University Research Committee Seed Funding Award 104004215, US National Science Foundation grants DMS-1407655, and NIH grants MH086633 and MH116527.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)

2. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J., Farahani, K., Davatzikos, C.: Segmentation labels and radiomic features for the pre-operative scans of the tcga-gbm collection. The Cancer Imaging Archive (2017)
3. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J., Farahani, K., Davatzikos, C.: Segmentation labels and radiomic features for the pre-operative scans of the tcga-lgg collection. The Cancer Imaging Archive (2017)
4. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.S., Freymann, J.B., Farahani, K., Davatzikos, C.: Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. Nature Scientific Data **4**, 170117 (2017)
5. Bakas, S., Reyes, M., et Int, Menze, B.: 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)
6. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785–794. ACM (2016)
7. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 424–432. Springer (2016)
8. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
9. Huo, J., Okada, K., Pope, W., Brown, M.: Sampling-based ensemble segmentation against inter-operator variability. In: Medical Imaging 2011: Computer-Aided Diagnosis. vol. 7963, p. 796315. International Society for Optics and Photonics (2011)
10. 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: International MICCAI Brainlesion Workshop. pp. 287–297. Springer (2017)
11. Kamnitsas, K., Bai, W., Ferrante, E., McDonagh, S., Sinclair, M., Pawlowski, N., Rajchl, M., Lee, M., Kainz, B., Rueckert, D., et al.: Ensembles of multiple models and architectures for robust brain tumour segmentation. In: International MICCAI Brainlesion Workshop. pp. 450–462. Springer (2017)
12. Kayalibay, B., Jensen, G., van der Smagt, P.: Cnn-based segmentation of medical imaging data. arXiv preprint arXiv:1701.03056 (2017)
13. Li, T., Fan, Z., Ziliang, Z., Hai, S., Hongtu, Z.: A label-fusion-aided convolutional neural network for isointense infant brain tissue segmentation. In: 2018 IEEE 15th International Symposium on Biomedical Imaging. pp. 692–695 (2018)
14. 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 Transactions on Medical Imaging **34**(10), 1993 (2015)
15. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE international conference on computer vision. pp. 1520–1528 (2015)
16. Notley, S., Magdon-Ismail, M.: Examining the use of neural networks for feature extraction: A comparative analysis using deep learning, support vector machines, and k-nearest neighbor classifiers. arXiv preprint arXiv:1805.02294 (2018)

17. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 234–241. Springer (2015)
18. Tustison, N.J., Cook, P.A., Klein, A., Song, G., Das, S.R., Duda, J.T., Kandel, B.M., van Strien, N., Stone, J.R., Gee, J.C., et al.: Large-scale evaluation of ants and freesurfer cortical thickness measurements. Neuroimage **99**, 166–179 (2014)
19. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4068–4076 (2015)
20. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance Normalization: The Missing Ingredient for Fast Stylization. arXiv.org (Nov 2017)
21. Wu, Y., He, K.: Group Normalization. arXiv.org (Jun 2018)
22. Zhou, F., Li, T., Li, H., Zhu, H.: Tpcnn: Two-phase patch-based convolutional neural network for automatic brain tumor segmentation and survival prediction. In: International MICCAI Brainlesion Workshop. pp. 274–286. Springer (2017)

Appendix

Table 3. Selected 75 cases to be another domain

	Name	Type		Name	Type
1	Brats18_2013_11_1	HGG	46	Brats18_2013_0_1	LGG
2	Brats18_2013_19_1	HGG	47	Brats18_2013_15_1	LGG
3	Brats18_2013_22_1	HGG	48	Brats18_2013_16_1	LGG
4	Brats18_2013_25_1	HGG	49	Brats18_2013_1_1	LGG
5	Brats18_2013_4_1	HGG	50	Brats18_2013_9_1	LGG
6	Brats18_CBICA_ABB_1	HGG	51	Brats18_TCIA09_141_1	LGG
7	Brats18_CBICA_ABO_1	HGG	52	Brats18_TCIA09_177_1	LGG
8	Brats18_CBICA_ANG_1	HGG	53	Brats18_TCIA09_255_1	LGG
9	Brats18_CBICA_ANP_1	HGG	54	Brats18_TCIA09_402_1	LGG
10	Brats18_CBICA_AOD_1	HGG	55	Brats18_TCIA09_462_1	LGG
11	Brats18_CBICA_AOH_1	HGG	56	Brats18_TCIA09_493_1	LGG
12	Brats18_CBICA_AOZ_1	HGG	57	Brats18_TCIA09_620_1	LGG
13	Brats18_CBICA_AQA_1	HGG	58	Brats18_TCIA10_130_1	LGG
14	Brats18_CBICA_AQQ_1	HGG	59	Brats18_TCIA10_261_1	LGG
15	Brats18_CBICA_AQR_1	HGG	60	Brats18_TCIA10_266_1	LGG
16	Brats18_CBICA_AQU_1	HGG	61	Brats18_TCIA10_276_1	LGG
17	Brats18_CBICA_ARW_1	HGG	62	Brats18_TCIA10_282_1	LGG
18	Brats18_CBICA_ARZ_1	HGG	63	Brats18_TCIA10_413_1	LGG
19	Brats18_CBICA_ASE_1	HGG	64	Brats18_TCIA10_420_1	LGG
20	Brats18_CBICA_ASH_1	HGG	65	Brats18_TCIA10_442_1	LGG
21	Brats18_CBICA_ATF_1	HGG	66	Brats18_TCIA10_490_1	LGG
22	Brats18_CBICA_AUQ_1	HGG	67	Brats18_TCIA10_628_1	LGG
23	Brats18_CBICA_AWI_1	HGG	68	Brats18_TCIA10_629_1	LGG
24	Brats18_CBICA_AXN_1	HGG	69	Brats18_TCIA10_637_1	LGG
25	Brats18_CBICA_AXQ_1	HGG	70	Brats18_TCIA10_644_1	LGG
26	Brats18_CBICA_AYI_1	HGG	71	Brats18_TCIA13_618_1	LGG
27	Brats18_CBICA_BFP_1	HGG	72	Brats18_TCIA13_621_1	LGG
28	Brats18_CBICA_BHB_1	HGG	73	Brats18_TCIA13_633_1	LGG
29	Brats18_CBICA_BHK_1	HGG	74	Brats18_TCIA13_645_1	LGG
30	Brats18_TCIA01_180_1	HGG	75	Brats18_TCIA13_650_1	LGG
31	Brats18_TCIA01_190_1	HGG			
32	Brats18_TCIA01_411_1	HGG			
33	Brats18_TCIA01_425_1	HGG			
34	Brats18_TCIA02_168_1	HGG			
35	Brats18_TCIA02_226_1	HGG			
36	Brats18_TCIA03_257_1	HGG			
37	Brats18_TCIA04_328_1	HGG			
38	Brats18_TCIA04_343_1	HGG			
39	Brats18_TCIA04_437_1	HGG			
40	Brats18_TCIA05_277_1	HGG			
41	Brats18_TCIA06_165_1	HGG			
42	Brats18_TCIA06_211_1	HGG			
43	Brats18_TCIA06_409_1	HGG			
44	Brats18_TCIA08_278_1	HGG			
45	Brats18_TCIA08_406_1	HGG			