

Brain Tumor Segmentation Using Deep Fully Convolutional Neural Networks

Geena Kim^(✉) 

College of Computer and Information Sciences,
Regis University, Denver, CO, USA
geenakim@regis.edu

Abstract. In this study, brain tumor substructures are segmented using 2D fully convolutional neural networks. A number of modifications such as double convolution layers, inception modules, and dense modules were added to a U-Net to achieve a deep architecture and test if the increased depth improves the performance. The experiments show that the deep architectures improve the performance. Also, the performance is enhanced from ensembling across the models trained on images in different orientations and ensembling across the models with different architectures. Even without any data augmentation, the ensembled model achieves a competitive performance and generalizes well on a new dataset. The resulting mean 3D Dice scores (ET/WT/TC) on the BRATS17 validation and test sets are 0.75/0.88/0.73 and 0.72/0.86/0.73.

Keywords: Brain Tumor Segmentation
Fully convolutional neural networks
Deep convolutional neural networks

1 Introduction

Though developed long ago, convolutional neural network (CNN) has recently become the most popular approach in the field of computer vision after the AlexNet won by a large margin in the Large Scale Visual Recognition Challenge (LSVRC), a computer vision challenge with a task of classifying objects in natural images [6]. In following studies, deep CNN models such as VGGNet, GoogLeNet (also called InceptionNet) and ResNet improved the classification accuracy further and showed that CNNs with deep-layer architectures can learn more complicated features from images [7–9]. Since then, the CNN approach has improved the accuracy beyond the human-level [10].

A deep CNN model has typically millions of parameters, thus requires a large amount of training images. The success of CNNs in recognizing natural images was largely due to the large public datasets such as the ImageNet, a massive dataset with 14 million images which annotation was crowd-sourced inexpensively [5].

Analyzing medical images is more challenging than natural images by nature and it is also costly to obtain annotated data. Recent efforts organizing challenges on automated medical image analysis and creating annotated datasets such as the Brain Tumor Segmentation (BRATS) challenge and its image collections [1–4] helped improving automated medical image analysis algorithms. Thanks to the considerable size and quality of the annotated data in BRATS, a number of deep CNN architectures have been developed and showed good performances [16–20]. With performances comparable to the top-performing non-neural network algorithms [11–15], the CNN-based algorithms have been steadily increasing their popularity in the challenge leaderboard, yet there are still rooms to improve.

Different from image classification tasks, the final output is 2D (or higher) in image segmentation tasks. Therefore, in segmentation, the classification of one pixel is highly correlated with that of adjacent pixels. One popular method to predict the label is the patch-based method, which a model classifies the center pixel based on the information from surrounding pixels (patch) [23]. Many CNN architectures previously implemented in BRATS use the patch-based approach which has an added benefit that it can help mitigating severe class imbalance by oversampling more patches from the tumor region. Recently, the fully convolutional neural network (FCN) approach showed that it can produce labels more time-efficiently than the patch-based approach [24] and was also implemented in BRATS [18, 20]. Other good examples of FCN-based models in medical images are the U-Net and the V-Net which have symmetric contracting and expanding paths and feature map concatenations between the two paths to preserve local features [25, 26].

In this paper, with an assumption that deep-layer architectures and the modules from the successful deep CNN models such as VGGNet, InceptionNet, ResNet, and DenseNet developed for recognizing natural images are also useful in medical images, modifications such as double convolution, inception modules and dense modules are implemented to the U-Net architecture to segment brain tumor substructures. Also ensembling across models trained on 2D images with axial, coronal and sagittal views as well as across the model architectures are implemented to enhance the performance.

2 Method

2.1 Data Preparation

The BRATS 17 Training data is normalized per MR image volume and per modality using the median pixel value X_m from the histogram.

$$X_n = \frac{X - X_m}{wX_m} \quad (1)$$

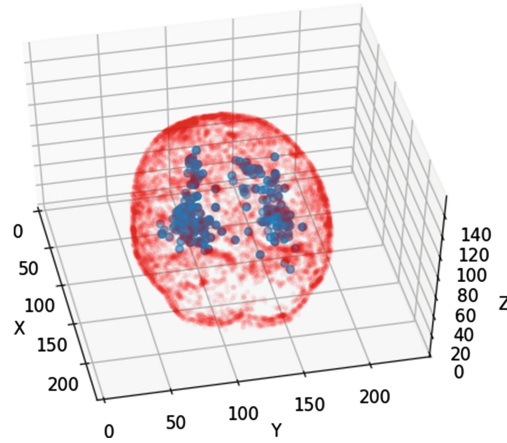


Fig. 1. 3D plot of tumor center locations (aggregated). Blue: Tumor centers, Red: brain contour. (Color figure online)

The constant w can be arbitrary, but was chosen to be 4 to approximate the histogram width. Using the training dataset's segmentation label, each case is categorized based on tumor location-size pair. Tumor location (simple mean x , y , z -coordinates of whole tumor pixels) is categorized to x : Anterior, Center, Posterior, y : Left, Center, Right, z : Top, Center, Bottom, and the size is categorized to Small, Medium and Large based on the tumor volume histograms (Fig. 2). Usually the tumor centers are found mostly in the middle part of the cerebrum and left and right outside to the ventricles (Fig. 1), but there are also less frequent cases found on the top or bottom parts of the brain.

After each scan is categorized, the training data is split into training, validation, and test subsets such that location-size pair categories are well mixed among those subsets. The 2D image slices are then shuffled within each subset. Though it may be useful, N4 biasfield correction was omitted in the preprocessing as it is time consuming. Also, no data augmentation such as translation, rotation, mirroring and warping was used in data preparation. About 30% of slices without any tumor or brain pixels were randomly discarded to make the training faster. Also, coronal- and sagittal-view 2D images were prepared in the same manner.

2.2 Architecture

Model A: Mini U-Net. Figure 3 shows the base model (mini U-Net). It has total 11 2d-convolution layers. Except the last convolution layer (1×1 convolution layer) for the output, all other convolution layers have 3×3 filters. After each convolution, batch normalization and ReLU activation are applied. Maxpooling is used to contract and upsampling is used to expand spatially. An upsampling layer uses simple resizing by nearest neighbor interpolation. As shown in Fig. 3,

a feature map from the last layer of each downsampling step is concatenated depth-wise with the upsampled layer.

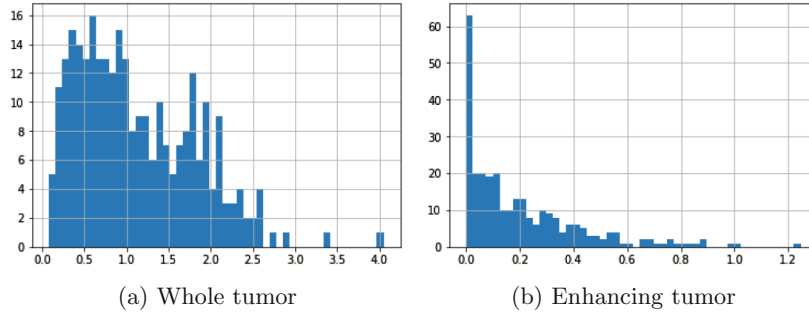


Fig. 2. Tumor volume histogram. Total pixels divided by the total pixels $\times 100$.

Model B: Mini U-Net with Double Convolution Layers. This model has the same structure with the model A except that there are two 3×3 convolution layers before an activation to increase the receptive field (field of view). Having two successive 3×3 convolution layers is equivalent to having one 5×5 convolution but is computationally less expensive by factor of $25/9$.

Model C: Mini U-Net with Inception Modules. Another variant of U-Net tested in this work uses modified inception modules. Typically, an inception module consists of 1×1 , $1 \times 1 - 3 \times 3$, $1 \times 1 - 5 \times 5$ and 3×3 maxpool- 1×1 paths running in parallel then merging depth-wise. In the InceptionNet, 1×1 convolutions before the other convolutions make it computationally less expensive, thus help obtaining a deep layer architecture with less number of parameters. Inception-like modules which replace 5×5 convolutions with two 3×3 convolutions as mentioned above are added to the mini U-Net. Figure 5 shows the overall architecture of the model C. A module A has 1×1 , $1 \times 1 - 3 \times 3$, and $1 \times 1 - 3 \times 3 - 3 \times 3$ convolution paths with stride = 1 and padding. A module B has a 3×3 max pooling layer with stride 2 (denoted as $/2$ by convention), $1 \times 1 - 3 \times 3/2$, and $1 \times 1 - 3 \times 3 - 3 \times 3/2$ convolution layers. A module C is a reverse of a module B and consists of an upsampling layer, $1 \times 1 - 3 \times 3/2$, and $1 \times 1 - 3 \times 3 - 3 \times 3/2$ deconvolution (transpose convolution) layers.

Model D: Mini U-Net with Dense Modules. This model incorporates Dense modules from a DenseNet architecture into a U-net. Like ResNet [9], DenseNet [27] has bypass connections where information from early layers can reach to the end without being lost by a series of activations. An important architectural difference is that ResNet achieves the bypass using element-wise addition known as residual connection while DenseNet uses depthwise concatenation after a convolution layer. DenseNet requires less number of parameters due to dense connections each of which adds a small number of feature maps

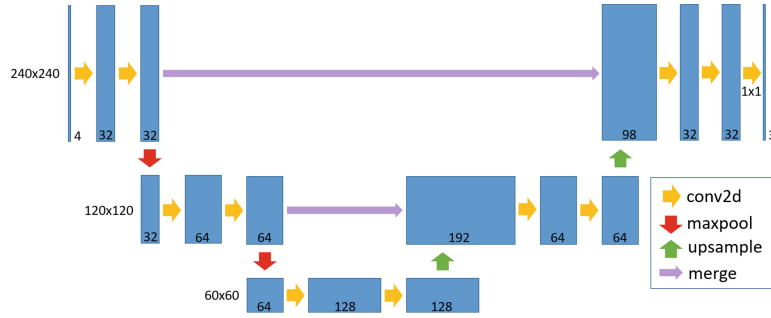


Fig. 3. Model A: mini U-Net architecture.

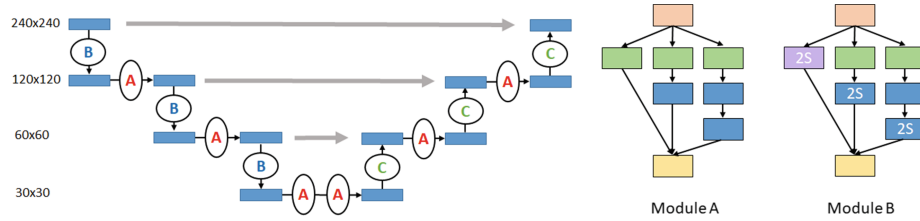


Fig. 4. Model C: mini U-Net architecture with inception-like modules. In the diagrams of Module A and B—orange squares represent the input layers to the modules, green squares represent 1×1 convolution layers, blue squares represent 3×3 convolution layers, and yellow squares represent depth-wise concatenation. (Color figure online)

to the concatenation and uses smaller number of weights rather than changing the weights for all the feature maps. As the depth of a network increases, even a small growth rate k , a hyper parameter (number of filters) of a dense module, can make the feature map size grow quickly. To reduce the number of feature maps, it is useful to use bottle neck and compression by using 1×1 convolutions. In this work, the DenseNet parameters were limited to $N = 4$ and $k = 8$ to fit the model in one GPU (Fig. 4).

2.3 Training

The models are trained using Keras and TensorFlow python libraries on a NVIDIA TitanX GPU. Models A, B, C, and D have roughly 0.5 M, 1 M, 0.5 M, and 0.2 M parameters and the training for 100–150 epochs takes 0.5–2 days on about 20 k images. All models use an Adam optimizer with a learning rate between 0.001 to 0.01, and the categorical cross-entropy for the loss function. Batch normalization is applied before every ReLU activation to reduce overfitting and to use a higher learning rate.

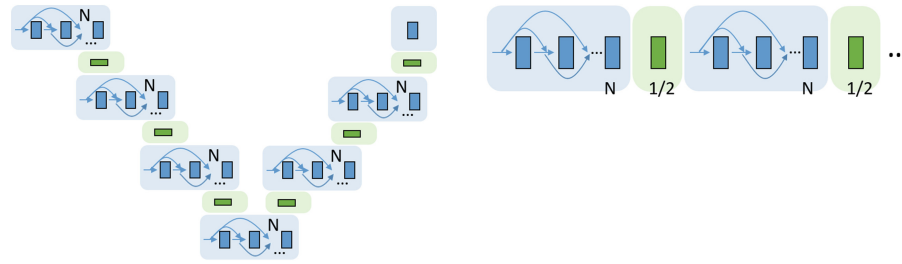


Fig. 5. Model D: mini U-Net architecture with dense modules. Left: the overall model architecture, Right: Zoomed-in view of dense modules. The blue squares represent convolution layers and the green squares are pooling and unpolling layers. Blue arrows represent feature maps concatenation. In a dense module (light blue block), 3×3 convolution layers are repeated N times and at the end of each convolution layer, feature maps from previous layers are concatenated. (Color figure online)

3 Result

3.1 Individual vs. Ensemble

A summary of individual model performances on axial view images is shown in Table 1. It shows that having more layers helps and increases the Dice score on both Training and Validation datasets. The models also generalize well on the Validation set. Training on the axial view images was a natural choice because most of the images in the dataset have been obtained in axial view, as well as the image shape is square (240×240), yet it is beneficial to combine information from side views (coronal and sagittal). Table 2 shows the performance boost when it is ensembled across all views for an individual model. Ensembling is done by averaging the probabilities on each pixel, then rounding up to create labels. There can be other ways to combine all views: for example, instead of averaging probabilities then rounding up, the label can be generated by voting after rounding up individual views. The latter might be a more strict way to eliminate a false positive label with a high probability from one orientation and low probabilities from other orientations, however, was not tested in this study. Table 3 shows the results from ensembling across all views and models. Ensembling across models still contributes to a higher Dice score although the effect is not as big as when ensembling across views. The ensembled model also generalizes well on the validation and the test set.

3.2 Dice Score by Labels

Figure 6 shows individual model Dice scores by segmentation labels and tumor substructures tested on the test subset of the training dataset. Dice score for necrosis and non-enhancing tumor has the largest spread and lowest mean, which may attribute to the small size and low frequency of necrosis in general. Each model performs similar on enhancing tumor with mean Dice scores between

Table 1. Individual model performances on axial images: mean 3D Dice score

Model	Train hours	Dice score (ET/WT/TC)	
		Test data	
		Train17	Val17
A. Baseline	20	63/83/75	64/83/68
B. Double-conv	27	64/85/77	68/86/70
C. Inception	54	66/86/77	66/87/72
D. Dense	20	66/85/78	68/86/72

Table 2. Ensembling across views: mean 3D Dice score

Model	View	Dice score (ET/WT/TC)	
		Tested on Train17	
		Individual	Ensemble
B. Double-conv	Axial	64/85/77	69/87/80
B. Double-conv	Coronal	65/81/76	
B. Double-conv	Sagittal	65/85/78	
D. Dense	Axial	66/85/78	70/85/78
D. Dense	Coronal	64/84/75	
D. Dense	Sagittal	64/82/77	

Table 3. Ensembling across views and models: mean 3D Dice score.

Model	View	Dice score (ET/WT/TC)				
		Train17		Val17		Test17
		Individual	Ensemble	Individual	Ensemble	Ensemble
B. Double-conv	Axial	64/85/77	71/87/81	68/86/70	75/88/73	72/86/73
C. Inception	Axial	66/86/77		66/87/72		
D. Dense	Axial	66/85/78		68/86/72		
B. Double-conv	Coronal	65/81/76		—		
C. Inception	Coronal	63/85/76		—		
D. Dense	Coronal	64/84/75		—		
B. Double-conv	Sagittal	65/85/78		—		
C. Inception	Sagittal	65/84/78		—		
D. Dense	Sagittal	64/82/77		—		

0.64–0.66. The outliers with low ET Dice scores are usually from the ones with little to no enhancing tumor region while the whole tumor size is big, or from the rare cases.

Table 4. Model performances on various data subsets in axial view: mean 3D Dice score. In the Val17 column, the [†] symbol means that the Dice scores are obtained using a subset (HGG) of the Val17 dataset.

Experiment	Model	Trained on	Dice score (ET/WT/TC)				
			Test data				
			TR15HGG	TR17HGG	TR17LGG	TR17ALL	Val17
1	A	TR15HGG	83/89/82	77/80/78	26/77/40	63/80/68	–
2	A	TR17HGG	–	79/84/83	20/66/32	64/81/72	80/83.5/81 [†]
3	A	TR17ALL	–	79/83/81	–	63/83/75	64/83/68
4	A	TR17HGG	–	79/84/85	22/72/32	64/82/71	–
5	B	TR17HGG	–	80/85/86	32/76/31	65/83/71	–
6	A	TR17LGG	–	62/72/50	24/82/66	52/75/54	–
7	B	TR17LGG	–	67/76/52	20/82/63	54/78/56	–
8	C	TR17LGG	–	66/77/49	24/82/66	55/79/54	–
9	B	TR17ALL	–	79/85/82	22/83/65	64/85/77	68/86/70
10	C	TR17ALL	–	80/87/82	25/83/60	65/86/77	66/87/72

3.3 Performance Comparison on HGG, LGG and HGG+LGG

Table 4 shows the individual model performance on various datasets in axial view. The performance of the model A (Baseline model: mini U-Net with a single convolution per layer) trained on the BRATS15 training dataset versus on the BRATS17 training dataset is shown in the experiments 1 and 2. The model trained on BRATS15 data has decreased Dice scores on BRATS17 data but is still comparable to when it was trained on BRATS17 data.

Also, models A, B, and C were trained and tested on different subsets of BRATS17 training data (HGG, LGG, and ALL). In general, the performances of models trained on HGG vs. HGG+LGG (ALL) data make little difference on ET and WT, but there is an improvement for TC Dice score when trained with HGG+LGG data. Also, usually models perform better when inferencing over HGG data than HGG+LGG or LGG alone. Inferencing enhancing tumor on LGG seems more challenging no matter which dataset a model was trained on, which might be because there are cases with a large whole tumor region without any enhancing tumor pixels (Fig. 7).

Experiments 1–3 and 4–10 use different normalization methods for preprocessing the images. The method used in experiments 4–10 uses the median pixel value in the brain pixels histogram instead of the 1–99 percentile pixel values to normalize. The previous method (1–99) used in experiments 1–3 was considered mainly to mitigate an effect of outlier pixel values. However, it has a drawback of

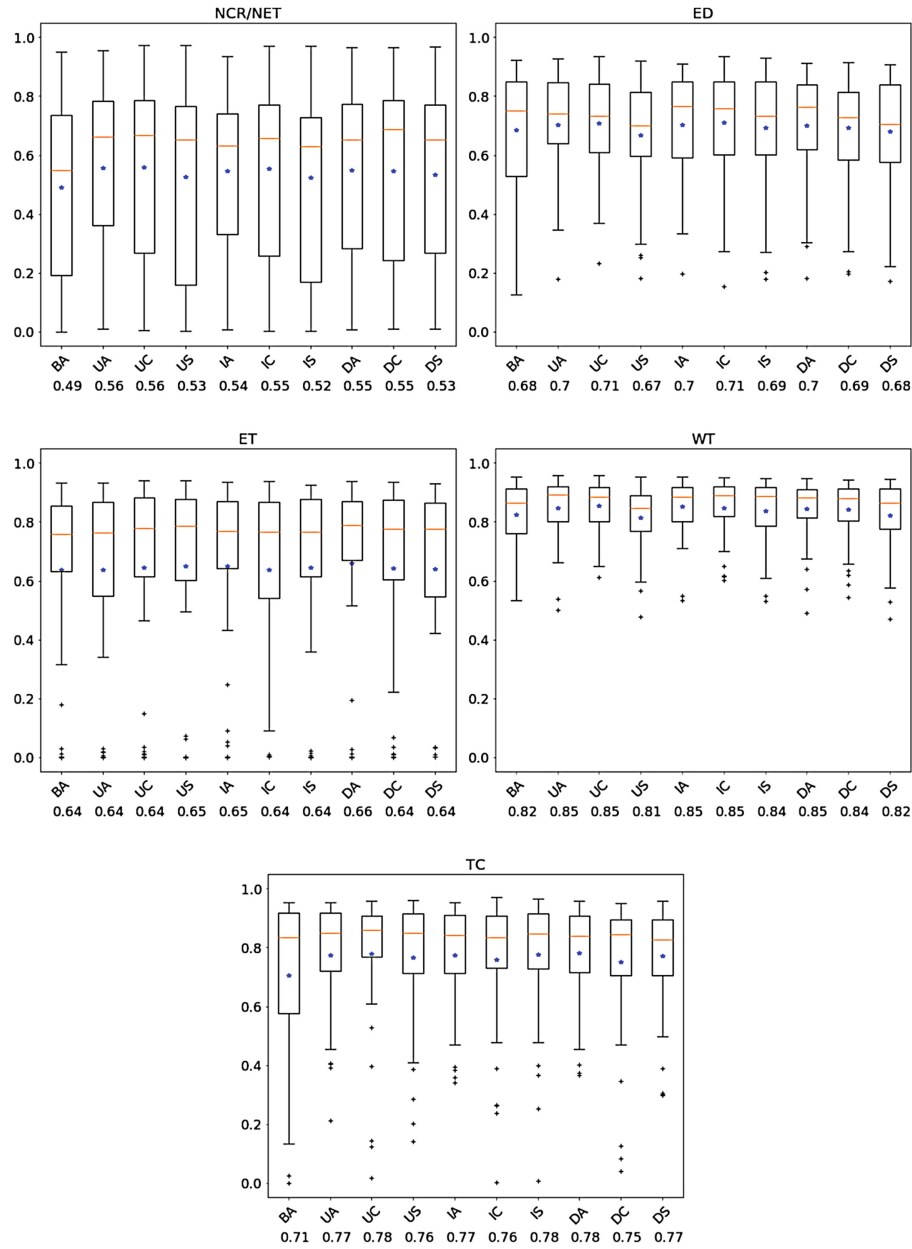


Fig. 6. Box plots of 3D Dice score for tumor substructures. Tested on the test-subset of the BRATS17 Training set. Top Left: Necrosis (NEC) and non-enhancing tumor (NET), Top Right: Edema (ED), Middle Left: Enhancing tumor (ET), Middle Right: Whole tumor (WT), Bottom: Tumor core (TC). The x-labels represent the models and training image orientation. For example, the first letters B, U, I, D mean baseline model, Unet with double convolutions, with Inception modules, with Dense modules. The second letters A, C, S are the orientations: axial, coronal and sagittal. Numbers below x-labels are the mean 3D Dice scores, also marked by blue star symbol on the plot. The y-axis is the 3d Dice score. The + symbols are the outliers. (Color figure online)

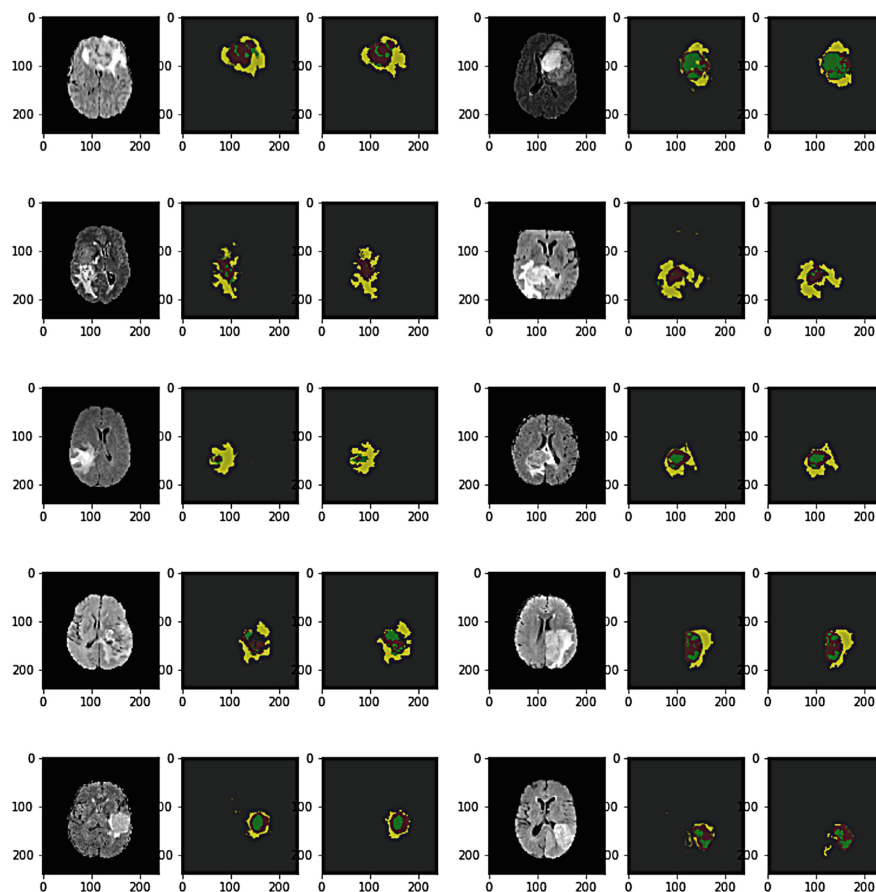


Fig. 7. Visualization of 10 random cases. For visualization purpose, ones with all three labels were selected. Each case is a set of three displays, which the left is the image (X) of the FLAIR modality, the middle is the prediction labels (Y_p), and the right is the manual segmentation labels (Y_t). For the label colors, Green: Necrosis+Non-enhancing Tumor, Yellow: Edema, Red: Enhancing Tumor. (Color figure online)

pushing the brightest 1% pixels to the same larger pixel values regardless of its original brightness, which makes more false positives for enhancing tumor. The second method (normalized by median) performs slightly better on the LGG data but shows little to no difference on the HGG+LGG data (Fig. 8).

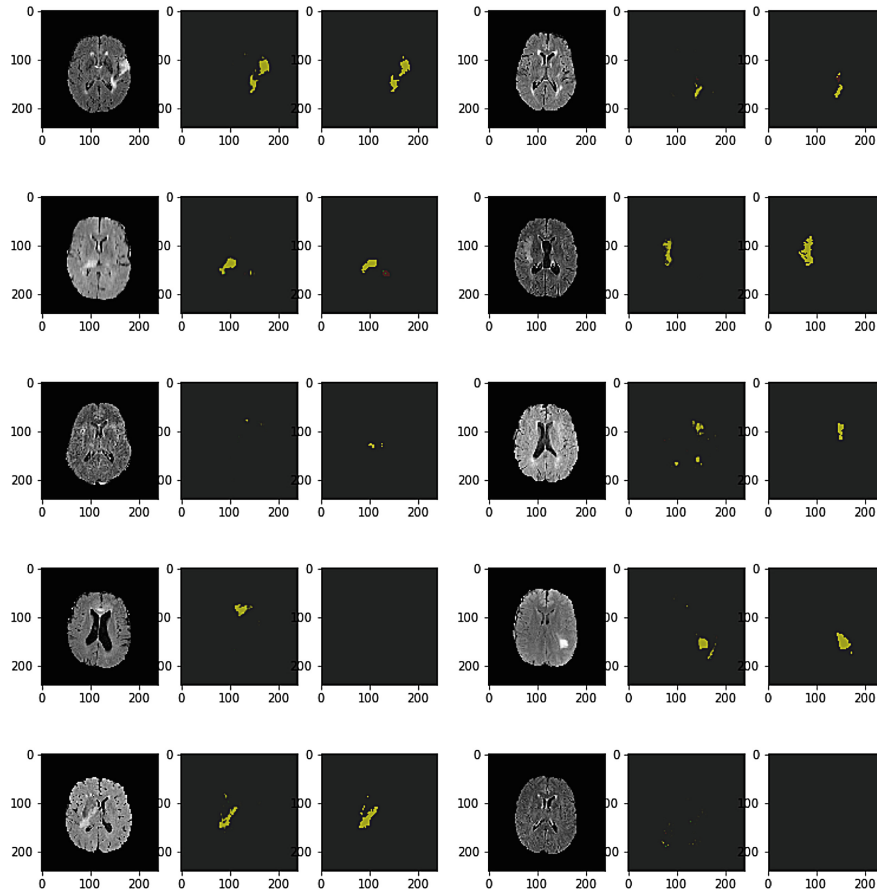


Fig. 8. Visualization of 10 random cases with small size tumors and difficult cases. Some cases with small tumors and slices without tumor may be prone to false positives. Each case consists of visualizations of the image X (FLAIR), prediction labels Y_p , target labels Y_t . Green: Necrosis+Non-enhancing Tumor, Yellow: Edema, Red: Enhancing Tumor. (Color figure online)

4 Conclusion

In this study, 2D fully convolutional neural networks were used to segment brain tumor substructures and address these questions: 1. Do models with deeper layers perform better? 2. How much a performance increases by ensembling models across views and across different architectures. In conclusion, a deeper layer architecture improves the performance. Ensembling can greatly enhance the performance especially when information from different views are combined. Perhaps is it due to that these models are based on 2D images and the information from other views help inferencing in 3D. An ensemble of the three models

(modified U-Nets with double convolution, Inception and Dense modules) trained on images in each of the three views (axial, coronal, and sagittal) gives 3D mean Dice scores 0.71/0.87/0.81 on the training set and generalizes well to similar values of 0.75/0.88/0.73 on the validation and 0.72/0.86/0.73 on the test set. Considering that the cases with the lowest scores are usually the rare cases (very small or unusual locations) or confusing cases (large whole tumor with little to no enhancing tumor), training with minority-oversampled and augmented data or curriculum learning may improve the result.

Acknowledgement. We thank NVIDIA for their kind donation of a TitanX GPU.

References

1. Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., Lanczi, L., Gerstner, E., Weber, M.A., Arbel, T., Avants, B.B., Ayache, N., Buendia, P., Collins, D.L., Cordier, N., Corso, J.J., Criminisi, A., Das, T., Delingette, H.: Demiralp, Ç., Durst, C.R., Dojat, M., Doyle, S., Festa, J., Forbes, F., Geremia, E., Glocker, B., Golland, P., Guo, X., Hamamci, A., Iftekharuddin, K.M., Jena, R., John, N.M., Konukoglu, E., Lashkari, D., Mariz, J.A., Meier, R., Pereira, S., Precup, D., Price, S.J., Raviv, T.R., Reza, S.M.S., Ryan, M., Sarikaya, D., Schwartz, L., Shin, H.C., Shotton, J., Silva, C.A., Sousa, N., Subbanna, N.K., Szekely, G., Taylor, T.J., Thomas, O.M., Tustison, N.J., Unal, G., Vasseur, F., Wintermark, M., Ye, D.H., Zhao, L., Zhao, B., Zikic, D., Prastawa, M., Reyes, M., Leemput, K.V.: The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Trans. Med. Imaging*. **34**, 1993–2024 (2015)
2. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J., Farahani, K., Davatzikos, C.: Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. *Nature Sci. Data* **4**, 170117 (2017)
3. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J., Farahani, K., Davatzikos, C.: Segmentation labels for the pre-operative scans of the TCGA-GBM collection. *Cancer Imaging Arch.* (2017)
4. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J., Farahani, K., Davatzikos, C.: Segmentation labels for the pre-operative scans of the TCGA-LGG collection. *Cancer Imaging Arch.* (2017)
5. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**, 211–252 (2015)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
8. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9 (2015)

9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
10. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)
11. Bakas, S., et al.: GLISTRboost: combining multimodal MRI segmentation, registration, and biophysical tumor growth modeling with gradient boosting machines for glioma segmentation. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Handels, H. (eds.) BrainLes 2015. LNCS, vol. 9556, pp. 144–155. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30858-6_13
12. Maier, O., Wilms, M., Handels, H.: Image features for brain lesion segmentation using random forests. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Handels, H. (eds.) BrainLes 2015. LNCS, vol. 9556, pp. 119–130. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30858-6_11
13. Meier, R., Karamitsou, V., Habegger, S., Wiest, R., Reyes, M.: Parameter learning for CRF-based tissue segmentation of brain tumors. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Handels, H. (eds.) BrainLes 2015. LNCS, vol. 9556, pp. 156–167. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30858-6_14
14. Song, B., Chou, C.-R., Chen, X., Huang, A., Liu, M.-C.: Anatomy-guided brain tumor segmentation and classification. In: Crimi, A., Menze, B., Reyes, M., Winzeck, S., Handels, H. (eds.) BrainLes 2016. LNCS, vol. 10154, pp. 162–170. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-55524-9_16
15. Zeng, K., et al.: Segmentation of gliomas in pre-operative and post-operative multimodal magnetic resonance imaging volumes based on a hybrid generative-discriminative framework. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Winzeck, S., Handels, H. (eds.) BrainLes 2016. LNCS, vol. 10154, pp. 184–194. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-55524-9_18
16. 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). https://doi.org/10.1007/978-3-319-30858-6_17
17. 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). https://doi.org/10.1007/978-3-319-30858-6_12
18. Chang, P.D.: Fully convolutional deep residual neural networks for brain tumor segmentation. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Winzeck, S., Handels, H. (eds.) BrainLes 2016. LNCS, vol. 10154, pp. 108–118. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-55524-9_11
19. Kamnitsas, K., et al.: DeepMedic for brain tumor segmentation. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Winzeck, S., Handels, H. (eds.) BrainLes 2016. LNCS, vol. 10154, pp. 138–149. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-55524-9_14
20. Zhao, X., Wu, Y., Song, G., Li, Z., Fan, Y., Zhang, Y.: Brain tumor segmentation using a fully convolutional neural network with conditional random fields. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Winzeck, S., Handels, H. (eds.) BrainLes 2016. LNCS, vol. 10154, pp. 75–87. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-55524-9_8

21. 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)
22. Kamnitsas, K., Ledig, C., Newcombe, V.F.J., Simpson, J.P., Kane, A.D., Menon, D.K., Rueckert, D., Glocker, B.: Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Med. Image Anal.* **36**, 61–78 (2017)
23. Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: *Advances in Neural Information Processing Systems*, pp. 2843–2851 (2012)
24. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440 (2015)
25. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) *MICCAI 2015*. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
26. Milletari, F., Navab, N., Ahmadi, S.-A.: V-Net: fully convolutional neural networks for volumetric medical image segmentation. [arXiv:1606.04797](https://arxiv.org/abs/1606.04797) [cs] (2016)
27. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely connected convolutional networks. [arXiv:1608.06993](https://arxiv.org/abs/1608.06993) [cs] (2016)