Residual Encoder and Convolutional Decoder Neural Network for Glioma Segmentation

Kamlesh Pawar*[1,2,[0000−0001−6199−2312]] ✉, Zhaolin Chen[3,4], N. Jon Shah[1,4], and Gary Egan[1,2]

[1] Monash Biomedical Imaging, Monash University, Melbourne, Australia
kamlesh.pawar@monash.edu
[2] School of Psychological Sciences, Monash University, Melbourne, Australia
[3] Electrical and Computer System Engineering, Monash University, Melbourne, Australia
[4] Institute of Medicine, Research Centre Juelich, Germany

Abstract. A deep learning approach to glioma segmentation is presented. An encoder and decoder pair deep learning network is designed which takes T1, T2, T1-CE (contrast enhanced) and T2-Flair (fluid attenuation inversion recovery) images as input and outputs the segmented labels. The encoder is a 49 layer deep residual learning architecture that encodes the $240 \times 240 \times 4$ input images into $8 \times 8 \times 2048$ feature maps. The decoder network takes these feature maps and extract the segmented labels. The decoder network is fully convolutional network consisting of convolutional and upsampling layers. Additionally, the input images are downsampled using bilinear interpolation and are inserted into the decoder network through concatenation. This concatenation step provides spatial information of the tumor to the decoder, which was lost due to pooling/downlsampling during encoding. The network is trained on the BRATS-17 training dataset and validated on the validation dataset. The dice score, sensitivity and specificity of the segmented whole tumor, core tumor and enhancing tumor is computed on validation dataset. The mean dice score for whole tumor, core tumor and enhancing tumor for validation dataset were 0.824, 0.627 and 0.575 respectively.

Keywords: deep learning, image segmentation, computer vision, CNN

1 Introduction

Gliomas are the tumors of the central nervous system which arises from glial cells. The gliomas are classified into two types depending on the aggressiveness of the tumor: high grade (HGG) and low grade (LGG) gliomas, both types of tumors are malignant and need treatment [1]. The accurate segmentation of gliomas is important in grading, treating and monitoring tumor progression.

* Caffe model and code available at: https://github.com/kamleshpawar17/BratsNet-2017

Mutiple magnetic resonance (MR) image contrasts are used to evaluate the type and extent of tumors. The different contrasts T1, T2, T1-CE and T2-Flair are analysed by a radiologists and tumor regions are manually segmented. Segmenting brain tumor is a comprehensive task, and large intra-rater variability is often reported, e.g. 20% [2]. Thus it is imperative to have a reliable automatic segmentation algorithm that standardizes the process of segmentation, resulting in more precise planning, treatment and monitoring.

Manual segmentation by an expert is a time consuming and expensive process, thus computer assisted tumor segmentation is imperative to the problem of brain tumor segmentation. Computer assisted methods [3,4,5,6,7,8] can be broadly classified into two categories: semiautomatic brain tumor segmentation and automatic brain tumor segmentation. Semiautomatic approach involves seeding some initial information by an expert such as a location of the tumor and the fine delineation and computational task is offloaded to a computer. This approach minimises the time spent by human experts. However large number of MR images are generated routinely in clinics, demanding fully automatic segmentation methods. The automatic segmentation methods require no human intervention and can segment the brain tumor into different classes such as necrotic tumor, enhancing tumor, tumor core and edema.

In this work we present a deep learning based approach to brain tumor segmentation on the BRATS-17 dataset [9,10,11,12]. Deep learning methods [13,14] based on convolutional neural networks (CNN) [15,16] have demonstrated highly accurate results in image classification[17,18] and segmentation [19,20]. However, selection of the number of CNN layers is a complex task. On one hand, increasing the number of layers improves complexity of the network and leads to more accurate results. On the other hand, designing deeper CNN may result in performance degradation due to exploding/vanishing gradients. This problem is partially solved by the batch normalization layers [21] that minimize the chances of exploding/vanishing gradients. Another limitation of designing deep neural network is that the training process becomes difficult and, after a certain depth the network ceases to converge. However recently introduced residual networks [22], consisting of short cut connections, can be trained to the larger depths. In this paper we present an encoder-decoder based CNN architecture to solve the tumor segmentation problem.

2 Deep residual learning networks

Deep learning CNN uses the layers of convolution as a feature extractor. The initial layers extract the basic features such as horizontal, vertical and slanted edges, the later layers extract more complicated features by combining the basic features. Thus increasing the depth results in more complicated features being extracted, hence better performance in classification/segmentation. However, it is found that increasing the network depth does not always increase the accuracy. The training error reduces with the depth of the network to a certain level but the error increasing as further layers are added to the network [23,24].

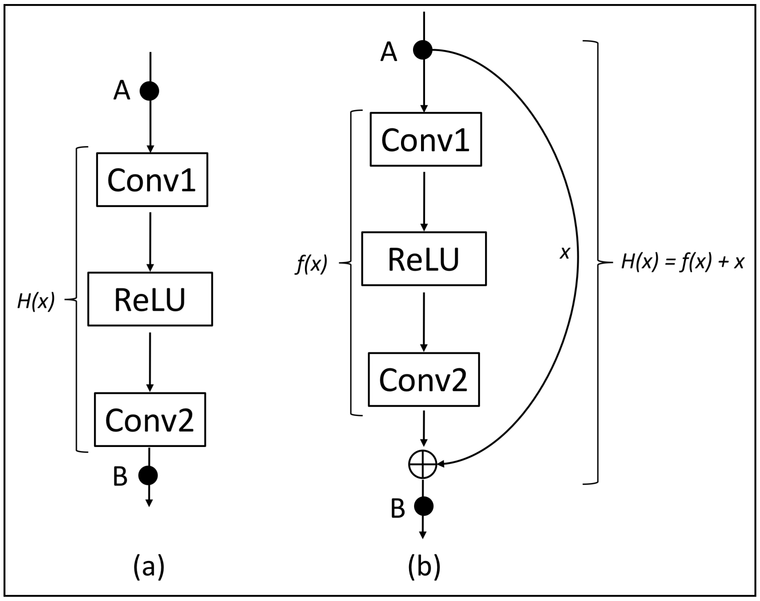

Fig. 1. If there exist a nonlinear function $H(x)$ between point A and B. **(a):** a network with no shortcut connection, here the layers Conv1 and Conv2 approximate the non linear function $H(x)$; **(b):** a network with shortcut connection, here the layers Conv1 and Conv2 approximate the nonlinear residual function $f(x) = H(x) - x$. Both (a) and (b) approximate the same non-linear function $H(x)$ between A and B, however individual layers learn different nonlinear functions.

This behaviour is counter intuitive, one may expect the error to decrease or stay constant after certain depth. The network may just learn identity mapping after certain depth and keep the error constant with even further increasing the depth of the network. Deep learning residual networks overcome this problem, by inserting the bottleneck units to increase the depth of the network. The bottleneck units learn the residual function that minimises the error.

Residual functions are realised with shortcut connection (Fig.1(b)), a shortcut connection is a direct connection between two layers in a network skipping one or few layers. Consider that there exists a nonlinear relationship between the two points A and B in the network given by $H(x)$ as shown in Fig.1. The conventional CNN without the shortcut connection (Fig.1(a)), would train the layers (Conv1 and Conv2) such that the combined effect of the learning represents $H(x)$. However in case of the network with shortcut connection (Fig.1(b)), the same layers (Conv1 and Conv2) would learn a residual function given by:

$$F(x) = H(x) - x \tag{1}$$

In both the cases the relationship between the point A and B remains the same, however the convolution layers have learned different functions. The advantage of using a residual network is that it may approximate an identity relationship between the point A and B, if the convolution layers Conv1 and Conv2 are deemed to be unnecessary. Thus the residual learning in effect avoids the performance degradation on going to arbitrarily large depths.

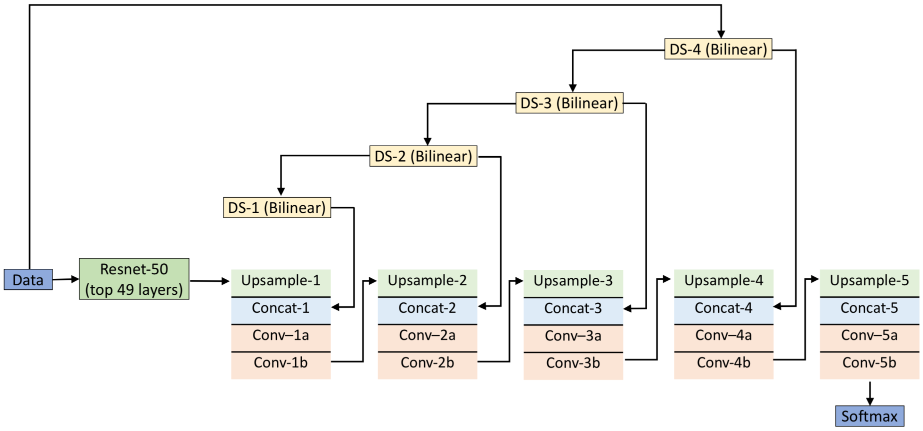

Fig. 2. Residual encoder and convolutional decoder network; the encoder is a 49 layer deep residual network and the decoder is a 10 layer deep fully convolutional network with bilinear upsampling layers. The input data is also downsampled using bilinear interpolation and is inserted back into the decoder through concatenation.

3 Proposed segmentation method

The method presented here is based on residual learning convolutional neural network [22] and the design is similar to the Unet encoder-decoder architecture [25]. The network is a 2D CNN, which performs segmentation on individual slices taken one by one from the full 3D dataset. The network is designed as an encoder-decoder pair, the four input images of size 240 × 240 are given as input to the encoder that encodes them into $8 \times 8 \times 2048$ data. This encoded data are provided to a fully convolutional decoder network that predict the labels for the glioma segmentation. The Resent-50 [22] which was the winner of ILSRVC 2015 image classification challenge is used as an encoder network followed by fully convolutional layers of decoder network.

3.1 Deep learning network architecture

Fig.2, shows the network architecture; it consists of the first 49 layers of Resnet-50 as the encoder. Each convolutional layer in the encoder is followed by a batch normalization and scaling layers [21], which avoids vanishing/exploding gradients. The first layer of the encoder consists of larger kernel size 7×7 with a stride of 2, and this reduces the input 240×240 image into 120×120, this is followed by a max-pooling layer that further reduces the size of the data to 60×60. The use of larger kernel increases the field of view, at the same time the stride of 2 and max pooling reduces the size to data, which results in reduction on memory requirement and reduces the training time. The layer parameters for encoder network, which is derived from first 49 layers of Resnet-50 are presented in [22].

The decoder network consists of upsampling layers that enlarges the dimension of image by a factor of 2. The weights of upsampling layer are fixed to bilinear upsampling and were not learned. During the encoding process the spatial information is lost due to pooling/downsampling, therefore the spatial information is reintroduced into the decoding network by concating the original images scaled by bilinear interpolation after each upsampling layer. After each convolutional layer, the batch normalization and scaling is performed followed by an ReLU non linear activation function. The last layer in the network is multinomial logistic layer that predicts the probability of a given pixel being either normal (label 0), necrotic/non-enhancing tumor (NCR/NET, label 1), edema (label 2) and enhancing tumor (label 4), . The decoder networks' input and output dimensions are shown in the Table.1. The input to the decoder network is 2048 features of size 8×8, which are passed successively through a series of convolution and upsampling layers, eventually generating a probability maps for the segmentation labels.

3.2 Dataset

The training dataset [9,10,11,12] used to train the network was provided by organisers of BRATS-17 challenge. The data consisted of 3D brain images from 285 subjects with four different contrast: T1, T1-CE, T2 and T2-Flair. There were 210 HGG subjects and 75 LGG subjects. The manually segmented labels were also provided for each subject. We divided the dataset into two groups:

- Train dataset: 259 subjects consisting of 192 HGG and 67 HGG subjects, used to train the network
- Test dataset: 26 subjects consisting of 18 HGG and 8 LGG subjects, used to evaluate the generalisation of the network.

Another two datasets for which the ground truth was not known were provided: one for validation consisting of 46 subjects and other for testing consisting of 147 subjects.

3.3 Pre-processing

The Caffe [26] deep learning library was used to train the network. The images provided by the organisers were in the NIFTI format, which were first converted

Layer	Input dimension	Output dimension	Kernel size, Stride
Upsample-1	$8 \times 8 \times 2048$	$15 \times 15 \times 2048$	3x3, 2 (weights fixed to bilinear)
Concat-1	$15 \times 15 \times 2048$	$15 \times 15 \times 2052$	
Conv-1a	$15 \times 15 \times 2052$	$15 \times 15 \times 1024$	3x3, 1
Conv-1b	$15 \times 15 \times 1024$	$15 \times 15 \times 1024$	3x3, 1
Upsample-2	$15 \times 15 \times 1024$	$30 \times 30 \times 1024$	3x3, 2 (weights fixed to bilinear)
Concat-2	$30 \times 30 \times 1024$	$30 \times 30 \times 1028$	
Conv-2a	$30 \times 30 \times 1028$	$30 \times 30 \times 512$	3x3, 1
Conv-2b	$30 \times 30 \times 512$	$30 \times 30 \times 512$	3x3, 1
Upsample-3	$30 \times 30 \times 512$	$60 \times 60 \times 512$	3x3, 2 (weights fixed to bilinear)
Concat-3	$60 \times 60 \times 512$	$60 \times 60 \times 516$	
Conv-3a	$60 \times 60 \times 516$	$60 \times 60 \times 256$	3x3, 1
Conv-3b	$60 \times 60 \times 256$	$60 \times 60 \times 256$	3x3, 1
Upsample-4	$60 \times 60 \times 256$	$120 \times 120 \times 256$	3x3, 2 (weights fixed to bilinear)
Concat-4	$120 \times 120 \times 256$	$120 \times 120 \times 260$	
Conv-4a	$120 \times 120 \times 260$	$120 \times 120 \times 128$	3x3, 1
Conv-4b	$120 \times 120 \times 128$	$120 \times 120 \times 128$	3x3, 1
Upsample-5	$120 \times 120 \times 128$	$240 \times 240 \times 128$	3x3, 2 (weights fixed to bilinear)
Concat-5	$240 \times 240 \times 128$	$240 \times 240 \times 132$	
Conv-5a	$240 \times 240 \times 132$	$240 \times 240 \times 64$	3x3, 1
Conv-5b	$240 \times 240 \times 64$	$240 \times 240 \times 64$	3x3, 1
SotmaxwithLoss	$240 \times 240 \times 64$	$240 \times 240 \times 4$	

Table 1. Decoder network input and output dimensions for each layer, the blob dimensions are $width \times height \times features$

to HDF5 file format, so that it can be read by Caffe. All the 3D volumes were normalised using histogram matching [27], the reference histogram for matching was obtained by averaging histograms of all the training dataset. Only the voxels with signal were used for computing the reference histogram and match the histogram.

3.4 Training

The training was performed on the train dataset using stochastic gradient descent with momentum in caffe. The training parameters were: base learning rate (lr_{base}) = 0.01, momentum = 0.9. The weights were decayed using the formula:

$$lr_{iter} = lr_{base} * (1 + \gamma * iter)^{(-power)} \tag{2}$$

where, lr_{iter} is learning rate during $iter^{th}$ iteration, $\gamma = 0.0001$, and $power = 0.75$. The network was trained for 45K iterations with the batch size of 8. The weights were regularized with l_2 regularization of 0.0005, during the training. Different base learning rate were tried and the one that provided minimum loss was used for the final training.

3.5 Evaluation method

The results of segmentation were evaluated using the dice score, sensitivity (true positive rate) and specificity (true negative rate). All the evaluation matrices were computed locally to test the local dataset of 26 subjects. The evaluation on the validation and test dataset was calculated on an online web-portal provided by BRATS-17 organisers.

4 Results

The trained network was tested on the validation data and the results of segmentation were uploaded on the computing portal provided by the organisers. The dice score, sensitivity, specificity and Hausdorff distance were computed on the segmented labels. The Table.2 and 3 show the result of segmentation of 46 subjects on validation dataset and 147 subjects on test dataset respectively. The results of the segmentation from 3 different subjects on the local testing dataset are shown in Fig.3. The whole tumor is defined as union of label 1, 2, and 4; the tumor core is defined as the union of label 1 and 4; and enhancing tumor is defined as label 4.

Table 2. Dice score, sensitivity, specificity and Hausdorfffor distance the segmentation on online validation dataset of 46 subjects

Metric	Whole Tumor	Core Tumor	Enhancing Tumor
Dice Score (mean)	0.824	0.627	0.575
Dice Score (median)	0.865	0.728	0.724
Sensitivity (mean)	0.831	0.669	0.595
Sensitivity (median)	0.885	0.746	0.690
Specificity (mean)	0.993	0.994	0.999
Specificity (median)	0.994	0.997	0.999
Hausdorff (median)	35.38	39.45	25.11

Table 3. Dice scores and Hausdorfffor distance for the segmentation on online test dataset of 147 subjects

	Whole Tumor	Core Tumor	Enhancing Tumor
Dice Score (mean)	0.784	0.577	0.502
Dice Score (Std. Dev)	0.152	0.280	0.315
Dice Score (median)	0.835	0.667	0.627
Dice Score (25 quantile)	0.731	0.359	0.200
Dice Score (75 quantile)	0.888	0.803	0.760
Hausdorff (median)	38.37	55.57	75.07

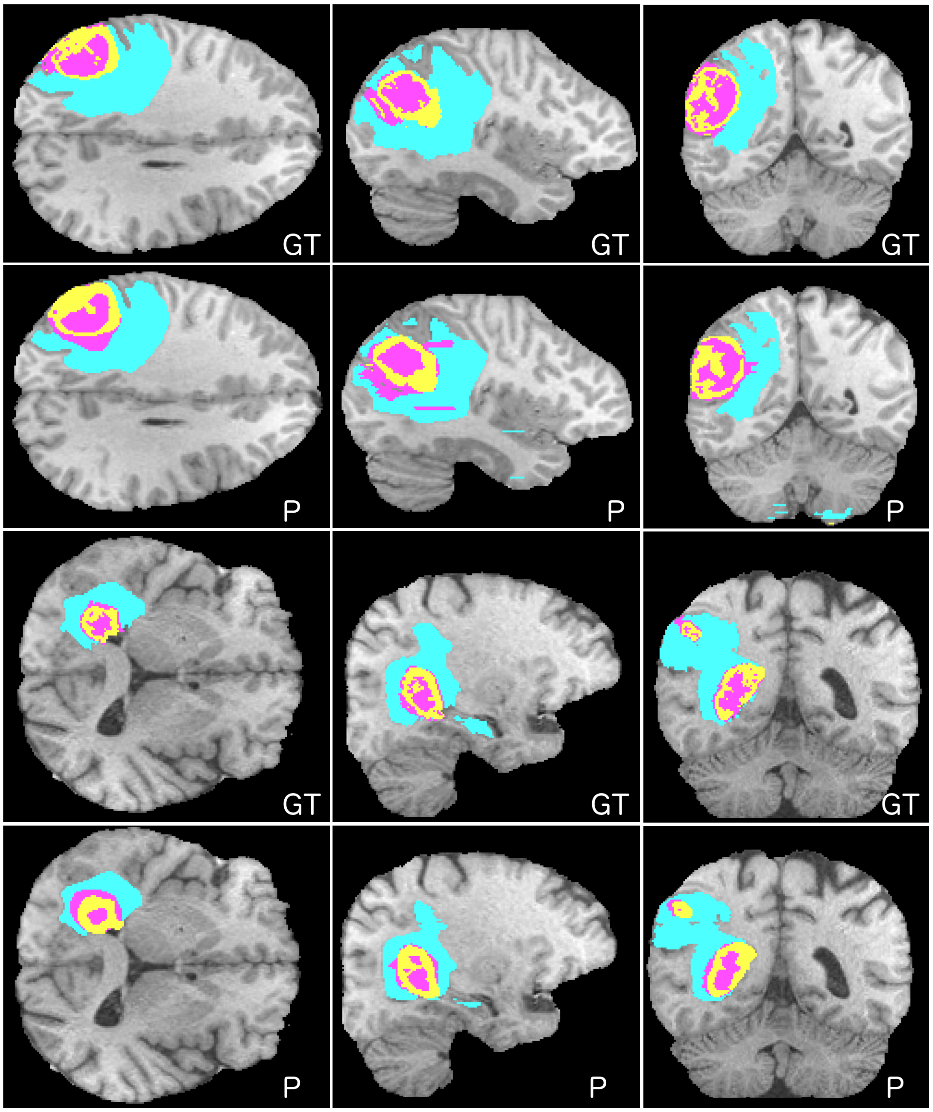

Fig. 3. Segmentation results on test dataset for 2 different subjects; **GT**: represents the ground truth segmentation; **P**: represents the segmentation predicted by the proposed residual encoder-decoder network. Colour coding scheme: magenta (label 0, NCR/NET), yellow (label 4, enhancing tumor) and cyan (label 2, edema)

5 Discussion

The median of dice score, sensitivity and specificity are all greater than the mean, which indicates that the proposed method performed well for most of the dataset but did not performed well for a few, that reduces the overall mean. The boundaries of the labels segmented by the proposed algorithm are smooth compared to the ground truth (Fig.3), this may be due to the fact that some of the ground truth were created using automated algorithms rather than human rater.

The network was trained only on the 2D axial slices of the 3D volume, thus it considered each slice as seperate instance and does not learn any correlation across the slices. This is evident from the predicted labels in the sagittal and coronal slices (Fig.3), where there are few isolated false positive prediction. These false positive prediction can be suppressed by incorporation, 3D information during training. One way to achieve this is to design a 3D residual network which takes a 3D volume and predicts a 3D label. This approach however would require large internal memory to store the network and 3D data/feature maps, which may not be feasible for a large deep networks. A moderate approach would be to use more than one slices at a time instead of full 3D volume for training. This approach would not increase the memory requirement and at the same time provides 3D information to the network. The number of slices to be used for this moderate approach would be determined by the available memory of the computing hardware. Ideally, all the slices should be used for the 3D architecture, but given the limitation of memory a trade of would be required between accuracy and memory.

6 Conclusion

In this paper, we developed a 59-layer deep residual encoder-decoder convolutional neural network that takes 2D slices of 3D MRI images as input and outputs the segmented labels. The bottleneck residual units make it feasible to train a 49 layer deep encoder. The bilinear scaling of input images serves as a guidance for the decoder network to spatially locate the tumor within a image.

Supporting information

The trained caffe model and all the scrips to train/finetune the network presented in this work are available at: `https://github.com/kamleshpawar17/BratsNet-2017`

References

1. D. N. Louis, H. Ohgaki, O. D. Wiestler, W. K. Cavenee, P. C. Burger, A. Jouvet, B. W. Scheithauer, and P. Kleihues, “The 2007 who classification of tumours of the central nervous system,” *Acta neuropathologica*, vol. 114, no. 2, pp. 97–109, 2007.
2. G. P. Mazzara, R. P. Velthuizen, J. L. Pearlman, H. M. Greenberg, and H. Wagner, “Brain tumor target volume determination for radiation treatment planning through automated mri segmentation,” *International Journal of Radiation Oncology* Biology* Physics*, vol. 59, no. 1, pp. 300–312, 2004.

3. J. Liu, M. Li, J. Wang, F. Wu, T. Liu, and Y. Pan, “A survey of mri-based brain tumor segmentation methods,” *Tsinghua Science and Technology*, vol. 19, no. 6, pp. 578–595, 2014.
4. E. D. Angelini, O. Clatz, E. Mandonnet, E. Konukoglu, L. Capelle, and H. Duffau, “Glioma dynamics and computational models: a review of segmentation, registration, and in silico growth algorithms and their clinical applications,” *Current Medical Imaging Reviews*, vol. 3, no. 4, pp. 262–276, 2007.
5. M. P. Gupta, M. M. Shringirishi *et al.*, “Implementation of brain tumor segmentation in brain mr images using k-means clustering and fuzzy c-means algorithm,” *International Journal of Computers & Technology*, vol. 5, no. 1, pp. 54–59, 2013.
6. J. J. Corso, E. Sharon, S. Dube, S. El-Saden, U. Sinha, and A. Yuille, “Efficient multilevel brain tumor segmentation with integrated bayesian model classification,” *IEEE transactions on medical imaging*, vol. 27, no. 5, pp. 629–640, 2008.
7. N. Sharma and L. M. Aggarwal, “Automated medical image segmentation techniques,” *Journal of medical physics/Association of Medical Physicists of India*, vol. 35, no. 1, p. 3, 2010.
8. D. L. Pham, C. Xu, and J. L. Prince, “Current methods in medical image segmentation,” *Annual review of biomedical engineering*, vol. 2, no. 1, pp. 315–337, 2000.
9. B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M.-A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, C. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H.-C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. V. Leemput, “The multimodal brain tumor image segmentation benchmark (brats),” *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2015.
10. S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos, “Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features,” *Nature Scientific Data*, 2017.
11. S. Bakas, H. Sotiras, M. Bilello, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos, “Segmentation labels and radiomic features for the pre-operative scans of the tcga-gbm collection,” *The Cancer Imaging Archive*, 2017, DOI: 10.7937/K9/TCIA.2017.KLXWJJ1Q.
12. S. Bakas, H. Sotiras, M. Bilello, J. Kirby, J. Freymann., K. Farahani, and C. Davatzikos, “Segmentation labels and radiomic features for the pre-operative scans of the tcga-lgg collection,” *The Cancer Imaging Archive*, 2017, DOI: 10.7937/K9/TCIA.2017.GJQ7R0EF.
13. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
14. J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural networks*, vol. 61, pp. 85–117, 2015.
15. Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, “Handwritten digit recognition with a back-propagation network,” in *Advances in neural information processing systems*, 1990, pp. 396–404.

16. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
17. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
18. D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.* IEEE, 2012, pp. 3642–3649.
19. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.
20. N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.
21. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.
22. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
23. K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5353–5360.
24. R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in neural information processing systems*, 2015, pp. 2377–2385.
25. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.
26. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.
27. D. Shapira, S. Avidan, and Y. Hel-Or, "Multiple histogram matching," in *Image Processing (ICIP), 2013 20th IEEE International Conference on.* IEEE, 2013, pp. 2269–2273.