An Ensemble of 2D Convolutional Neural Networks for Tumor Segmentation

Mark Lyksborg(✉), Oula Puonti, Mikael Agn, and Rasmus Larsen

Department of Applied Mathematics and Computer Science,
Technical University of Denmark, Kgs. Lyngby, Denmark
{mlyk,oupu,miag,rlar}@dtu.dk

Abstract. Accurate tumor segmentation plays an important role in radiosurgery planning and the assessment of radiotherapy treatment efficacy. In this paper we propose a method combining an ensemble of 2D convolutional neural networks for doing a volumetric segmentation of magnetic resonance images. The segmentation is done in three steps; first the full tumor region, is segmented from the background by a voxel-wise merging of the decisions of three networks learned from three orthogonal planes, next the segmentation is refined using a cellular automaton-based seed growing method known as growcut. Finally, within-tumor sub-regions are segmented using an additional ensemble of networks trained for the task. We demonstrate the method on the MICCAI Brain Tumor Segmentation Challenge dataset of 2014, and show improved segmentation accuracy compared to an axially trained 2D network and an ensemble segmentation without growcut. We further obtain competitive Dice scores compared with the most recent tumor segmentation challenge.

Keywords: Tumor segmentation · Convolutional neural network · Ensemble classification · Cellular automaton

1 Introduction

Segmentation of brain tumors plays a role in radiosurgery, radiotherapy planning, and for monitoring tumor growth. Segmentation is challenging since tumor location and appearance vary greatly between patients.

Many successful method for doing voxel-based segmentation are based on the random forest (RF) classification scheme which predicts segmentation labels from user engineered image features. Tustison et al. [15] proposed a two-stage RF approach, with features derived from a Gaussian mixture model followed by a Markov random field segmentation smoothing. The RF was also used by Reza et al. [12] who designed features using textons and multifractional Brownian motion. Menze et al. [10] proposed a generative probabilistic atlas-based model which adapts to the intensity distribution of different subjects and later combined it with the RF classifier [9]. An example of a successfull method that does not use a RF classifier is the patch-based approach [2]. Here voxels are

R.R. Paulsen and K.S. Pedersen (Eds.): SCIA 2015, LNCS 9127, pp. 201–211, 2015.
DOI: 10.1007/978-3-319-19665-7_17

segmented by comparing image patches to a dictionary consisting of training patches where the corresponding expert labels are used for segmentation.

In recent years and due to advancements in computational power, deep neural networks have been revived. In the most recent Brain Tumor Segmentation Challenge 2014 (BraTS2014), this was reflected by a number of contributions using deep neural networks. The work by Davy et al. [3] presented a 2D convolutional network trained from an axial perspective. Two others presented 3D networks [16], [18], and while their implementations differed, the results indicated a benefit of using 3D information. An important property of a network is that it learns image features relevant for the specific segmentation problem. This alleviate researchers from having to engineer such features.

We revisit the idea of Davy et al. [3] but instead of using one 2D network to do voxel-based segmentations, we learn an ensemble of networks, one for each of the axial, sagittal and coronal planes and fuse their segmentations into a more accurate 3D informed segmentation. Unlike previous works using convolutional networks we do not segment the tumor and its sub-regions using a single multi-label classifier. Instead, we split the problem into two sequential segmentation problems. The first segmentation separates tumor from healthy tissue and refine the segmentation using a growcut algorithm [17]. The second segmentation performs the within-tumor sub-region segmentation using the tumor mask of the first segmentation to select voxels of interest.

The method (Fig. 1) is demonstrated on the BraTS2014 dataset. We were able to achieve improved ground truth segmentation accuracy compared to a 2D axially trained network [3] and Dice scores [4] just below the top methods of the challenge leaderboard (https://www.virtualskeleton.ch/BRATS/Start2014).

2 Data

Two datasets were downloaded from the BraTS2014 website (November, 2014).

The first dataset (data1) consisted of 106 high grade glioma (HGG) and 25 low grade glioma (LGG) subjects (no longitudinal repetitions), all with ground truth segmentations of the tumors. It was randomly split into a training set of 76 HGG/15 LGG subjects, and the rest (30 HGG/10 LGG) were used as test data. For each subject, we used a set of multimodal magnetic resonance imaging (MRI) volumes, consisting of two T2-weighted images (Fluid-attenuated inversion recovery (FLAIR) and (T2)) and a T1-weighted image with gadolinium contrast (T1c). The MRIs were skull stripped, rigidly oriented according to MNI space and re-sliced to 1 mm^3 as described in [6]. The ground truth segmentation consisted of five labels (background=0, necrosis=1, edema=2, non-enhancing=3, enhancing=4).

The second dataset (data2) consisted of 187 multi-modal MRI volumes from 88 different subjects with 99 longitudinal repetitions. Since only the BraTS2014 challenge organizers know the ground truth segmentations, it allowed for a blinded segmentation evaluation via the challenge website.

3 Method

The proposed method, outlined in Fig. 1, consists of four steps. First, the MRI volumes are bias corrected for scanner field inhomogeneity and standardized to similar cross subject intensities. Second, an ensemble of convolutional networks segments the tumor from healthy tissue. The third step (growcut) post processes the segmentation to improve the segmentation. The fourth step does the within-tumor segmentation using an additional ensemble of networks. The four steps of the method are detailed successively in section 3.1-3.4.

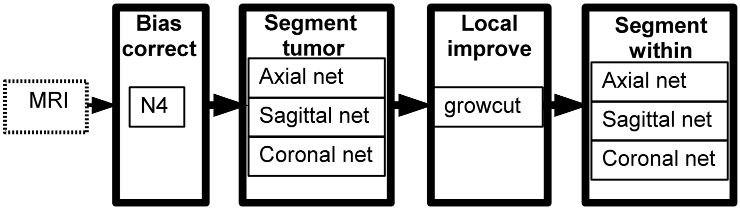

Fig. 1. Shows a schematic, outlining the pipeline of our method. The multi-modal MRI data is pushed through four successive stages of 1) bias correction, 2) whole tumor segmentation (tumor vs. none tumor), 3) localized post-processing of the segmentation and 4) a within-tumor segmentation stage.

3.1 Bias Correction and Standardization

MRI generally exhibits large intensity variations even within the same tissue type of a subject, largely due to field inhomogeneity of the scanner. To minimize this bias, the N4 method [14] was applied to each MRI.The N4 method works under the assumption that the bias field can be modeled by a smooth multiplicative model which is fitted iteratively to maximize the high frequency content of the MRI intensity distribution. To further standardize across different scanners, the maximum peak of each MRI intensity histogram was found, and the intensities scaled according to $I = I_c \cdot (I_b/I_p)$, where I_c is the N4 bias corrected image volume, I_p is the maximum peak intensity of I_c and I_b is a reference value which we fixed to $I_b = 200$. To achieve equal importance of the multi-modal MRI, their intensities were further standardised using a normal transformation applied to each of the different modalities.

3.2 Convolutional Network Ensemble: Whole Tumor

To segment tumor tissue, three convolutional neural networks were trained using a multi-modal image patch of dimension 46×46. Each 2D network learned to classify the same center voxel but viewed from an axial, sagittal and coronal perspective. Combining this ensemble of 2D networks enabled the segmentation method to become 3D aware.

The 2D networks are described by the architecture in Fig. 2. It shows a network consisting of 6 layers. Each perform an algebraic operation on the input data x and passes the result as input to the next layer. The process is repeated until reaching layer 6 which predicts the most probable classification label.

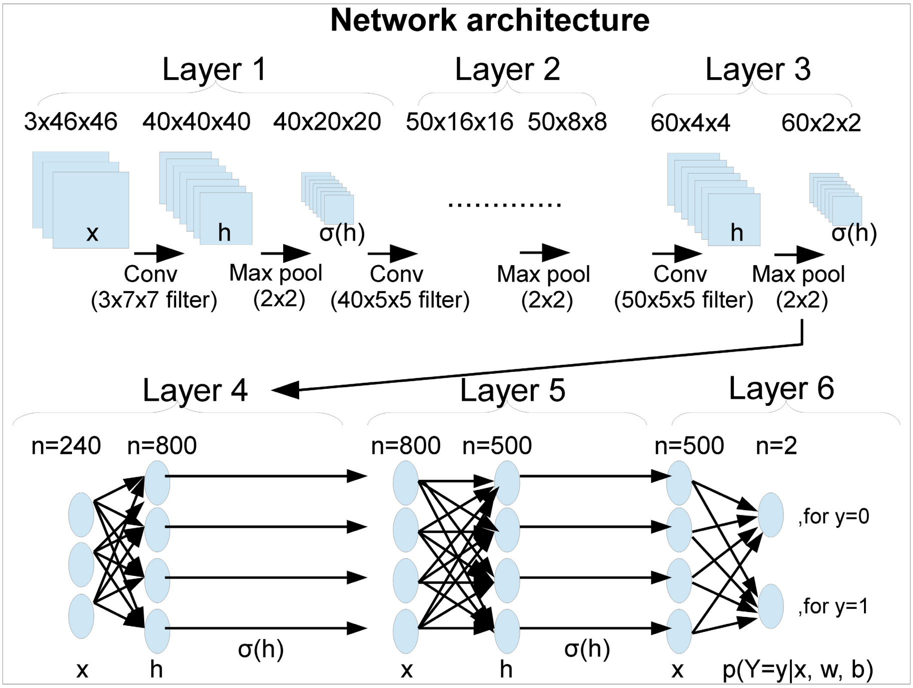

Fig. 2. Depicts a 2D deep neural network architecture consisting of six layers. The first three are convolutional layers, followed by two fully connected layers and a softmax layer where the arrows indicate the connections between layers. The squares illustrate the 2D nature of the input (x) and the intermediate representations (h) of the convolutional layers, where $x = [x_1 ... x_n]$ is a 3D matrix of n input patches and $h = [h_1 ... h_m]$, is the concatenation of m 2D filter response. The circles of the fully connected layers indicate its 1D nature with n being the number of neurons (=the circles), such that $x = [x_1 ... x_n]^T$ and $h = [h_1 ... h_n]^T$ are the 1D vector representations of the input and the neuronal activations.

Convolutional layers: The convolutional layers apply filtering and downsampling operations to image patches. The first layer uses a filter bank of size $40 \times 3 \times 7 \times 7$ which it applies to the $3 \times 46 \times 46$ image patch. This produces a feature map h of size $40 \times 40 \times 40$, where the first dimension indexes the feature maps, while the second and third dimensions indexes (row, column) coordinates. More specifically the j^{th} map is calculated by $h_j = b_j + \sum_{i=1}^{n}(w_{ij} * x_i)$, where i

indexes the input channel and a trainable filter w_{ij}, the $*$ operator denotes 2D convolution and $n = 3$ is the number of input channels. Subsequently a 2×2 max pooling strategy is used to downsample h to size $40 \times 20 \times 20$ and the rectified linear unit function, $\sigma(h) = max(0, h)$ is applied. The remaining convolutional layers (two and three) perform the same type of operations but using filter banks of size $50 \times 40 \times 5 \times 5$ and $60 \times 50 \times 5 \times 5$ for the respective layers. The application of these filters and downsampling steps result in a number of the intermediate feature maps with the dimensionalities listed in the top part of Fig. 2.

Fully connected layers: Layer 4, 5 and 6 are fully connected layers meaning each neuron is exposed to the full input x of the previoues layer. Each of the 800 neurons in layer 4, evaluates the product $h_j = w_j^T x + b_j$ and applies the non-linear activation function $\sigma(h_j)$. Thereby transforming the 240 dimensional vector x into an 800 dimensional vector $\sigma(h)$ which is passed to layer 5. Layer 5 works similar to layer 4, but now generating a 500 dimensional feature vector $\sigma(h)$ which is propagated to layer 6. Layer 6 evaluates the softmax function

$$p(Y = y|x, w, b) = \frac{e^{w_y x + b_y}}{\sum_j e^{w_j x + b_j}}, \tag{1}$$

generating posterior probabilities for a number of classification labels, $y = \{0, 1\}$. Here w_j refer to a vector of linear parameters for the j^{th} class, b_j is a bias weight and x is the 500 dimensional response vector from the previous layer.

Network Training Each of the 2D networks were trained by minimizing the following cost function

$$C(W, B) = \frac{1}{nd} \cdot \sum_{i=1}^{nd} - \ln(p(Y = y^i|x^i, W, B)) + \lambda \cdot \sum_{j=1}^{nw} W_j^2. \tag{2}$$

The first term of eq. (2) is the mean negative log-likelihood of the softmax probability and we have used capitalized (W, B) to indicate that it is a function of (w, b) parameters from different types of layers. Further, the training patches are denoted x^i, y^i, corresponding to the patch intensities and ground truth label of the i^{th} training example. The second term of eq. (2) is a regularization term that adds robustness to the optimization problem by limiting the solution space to models with smaller parameter weights. It does so by penalizing the 2-norm of the parameters and through experimentation we found $\lambda = 0.0001$ to be suitable.

The cost function was minimized using a stochastic gradient descent (SGD) which relied on the back propagation algorithm to estimate gradients. The SGD performed iterative updates based on gradients estimated from mini-batches with a batch size of 200 where an update occurred after each mini-batch. Each gradient update was further augmented by a moment based learning rule [13] which updated the parameters as a weighted combination of the current gradients and the gradients of previous iteration update. We used a momentum coefficient of 0.9. Layer 4 and 5 were trained using the dropout learning [5] (dropout rate=0.5) which activates half the neurons for each training example. As a consequences the activations of these layers($\sigma(h)$) were divided by 2 when a network was applied to an unseen test image patch.

A GPU implementation for training the three 2D networks was achived using Theano [1].

Network Ensemble Merging Having learned the parameters of the three networks, their complementary decision information were merged. This was done using the posterior probablities of the last layer (layer 6). If the networks agreed on the same label we were highly confident in this classification and assigned the label of voxel x with probability $p(Y|x) = 1$. Otherwise a majority vote decided the class label and the probability was set to reflect this uncertainty by averaging the class probabilities of the three networks, $p(Y|x) = (1/3)\sum_{i=1}^{3} p_i(Y|x, w, b)$. The resulting label segmentations and their probabilities were then used as input for the growcut algorithm.

3.3 Cellular Automaton: Growcut

The growcut algorithm was initially proposed as a continuous state cellular automata method for automated segmentation based on user labeled seed voxels [17]. From these labels and a local intensity transition rule the algorithm decides whether voxels should be re-labelled.

We used the algorithmic formulation of [17] which we extended to 3D. The algorithm models each voxel as a cell with a state set $S(\Theta, l, C)$ consisting of a strength value $\Theta \in [0, 1]$, a label l and an intensity feature vector C. It is an iterative algorithm and for each iteration the strength and labels of the previous iteration remain fixed. During an iteration each image cell r is attacked by its neighboring cells $s \in N(r)$ where $N(r)$ denote the $3 \times 3 \times 3$ neighborhood of a volume and only if $g(C_r, C_s) \cdot \Theta_s > \Theta_r$, will Θ_r, and l_r be updated before the next iteration. The local transition rule is given by

$$g(C_1, C_2) = 1 - \frac{||C_1 - C_2||_2}{k} \tag{3}$$

Where we have normalized the intensities of C to be in the range [0, 1] such that for $k = \sqrt{3}$, the value of $g(C_1, C_2) \in [0, 1]$. Since $g(C_1, C_2)$ can never exceed 1, any cells with strength $\Theta = 1$ will remain constant throughout the algorithm.

To use the growcut on the ensemble segmentations, the feature vector C was set to the multi-modal MRI intensities and the values of l, Θ were initialized with the labels and probability maps of the convolutional network ensemble. This initialization served as a strong prior for growcut segmentation, assuming that the segmentation was already near optimal.

Once growcut converged to a stable segmentation (100 iterations), a heuristic rule was used to identify the tumor. It was based on a connected components analysis to remove any spatially coherent clusters of voxels which were less than 80% of the biggest cluster.

3.4 Convolutional Network Ensemble: Within-Tumor

This ensemble of convolutional networks was used to segment the within-tumor sub-regions. The architecture of each network is similar to the previously described, but considers a smaller image patch and has only two convolutional layers, two fully connected dropout layers and softmax probability layer. The input patch size is $3 \times 34 \times 34$ and the first convolutional layer uses a filter bank of size $50 \times 3 \times 7 \times 7$ while the second one uses a filter bank of size $60 \times 50 \times 5 \times 5$. The justification of choosing a smaller patch size is that the within-tumor segmentation uses information on a smaller scale compared to the whole tumor segmentation. As with the previously described networks, the fully connected layers use 800 and 500 neurons respectively while the softmax layer, predicts one of four possible classification labels. The SGD optimization was again used to train the networks but for these specific networks we used $\lambda = 0.00005$.

Network Ensemble Merging The voxel-based decisions of the ensemble of axial, sagittal and coronal networks were either set to the label they all agree on, or according to the most probable average probability of the softmax probability.

4 Results

4.1 Test and Phenotype Performance

Testing our method on the 40 left out subjects (data1), resulted in the segmentation performances of Table 1. This table shows ground truth scores for three methods; A 2D convolutional network applied to the axial plane similar to [3], a method using only the ensemble part of our method (ensem) and our full method which is ensem in combination with growcut (ensem+grow). The scores of the table are given for pathologically relevant tumor regions. These are the whole tumor (labels: necrosis, edema, non-enhancing, enhancing), the enhanced tumor region and the tumor core (labels: necrosis, non-enhancing, enhancing). We see that using an ensemble improved the segmentation relative to a 2D network and achieved further improvement by including growcut post-processing. As a visual comparison example, two tumor segmentations based on our method and their

Table 1. Average segmentation performance scores of three convolutional neural network methods evaluated on 40 subjects of data1. The scores (Dice, positive predictive and sensitivity) were calculated for the different tumor regions.

Method	Dice scores			Positive predictive			Sensitivity		
	Whole	Core	Enh.	Whole	Core	Enh.	Whole	Core	Enh.
axial	0.744	0.642	0.629	0.732	0.624	0.642	0.811	0.746	0.707
ensem	0.786	0.686	0.676	0.786	0.707	0.693	0.825	0.743	0.717
ensem+grow	0.810	0.697	0.681	0.833	0.718	0.701	0.825	0.750	0.720

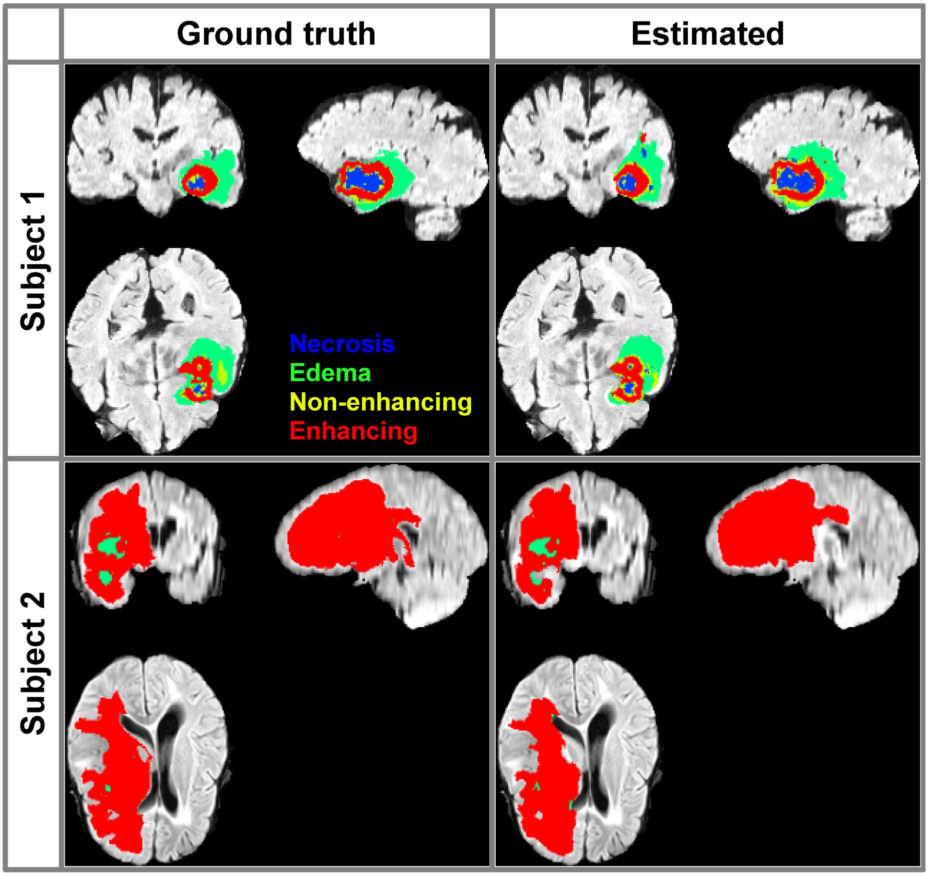

Fig. 3. This visual comparison shows both the proposed segmentation method and corresponding ground truth for two subjects. The Dice scores of subject 1 were 0.825 (whole), 0.795 (core) and 0.842 (enhanced) and for subject 2 they were, 0.892 (whole), 0.840 (core) and 0.854 (enhanced).

ground truth, are shown in Fig. 3. By dividing the test subjects based on tumor types (HGG/LGG), we evaluated their impact on method performance. This comparison (Fig. 4), reveals higher Dice scores with less variance for the HGGs, indicating a methodological bias towards the tumor type.

4.2 Blinded Challenge Performance

Testing our method on the blinded challenge dataset previously denoted data2 and performing an on-line evaluation of the segmentations, resulted in the average performance scores of Table 2. It lists the scores for the first time point of the 99 subjects (cross sectional) and the full challenge data (full data) where similar performances are achieved. It also includes the top 3 scores of the BraTS2014 challenge where our method is ranked amongst.

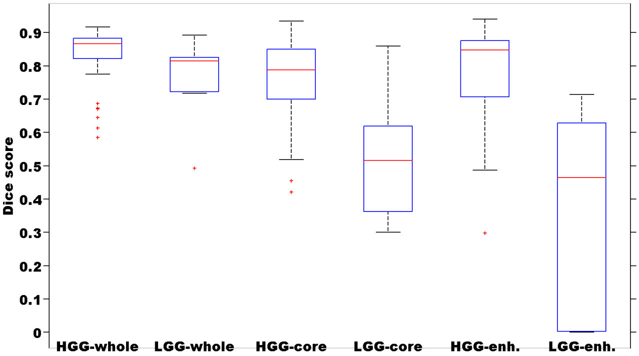

Fig. 4. Ground truth Dice scores performance for two different types of tumors (HGG and LGG). Red line indicate mean Dice score, blue boxes show the 25 and 75 percentiles of the scores while extreme observations are show with red dots.

Table 2. Shows the average segmentation performance scores of our method in grey (cross sectional and full data), for the BraTS2014 challenge data (data2). Also listed are the top three of the challenge (15/12-2014), ranked according to their whole tumor Dice scores. These are Urbag [16], Kleej [7], Dvorp [8].

Method	Dice scores			Positive predictive			Sensitivity		
	Whole	Core	Enh.	Whole	Core	Enh.	Whole	Core	Enh.
Cross sectional	0.801	0.637	0.586	0.803	0.682	0.554	0.857	0.715	0.745
Full data	0.799	0.631	0.625	0.783	0.629	0.580	0.861	0.736	0.776
Urbag	0.87	0.76	0.72	0.91	0.80	0.69	0.85	0.76	0.81
Kleej	0.87	0.76	0.73	0.90	0.73	0.66	0.85	0.83	0.87
Dvorp	0.60	0.30	0.29	0.86	0.58	0.56	0.53	0.27	0.28

5 Discussion

We have presented a method, combining an ensemble of 2D convolutional networks with the growcut method for making a 3D informed segmentation. It showed improved accuracy compared to a 2D network and an ensemble segmentation without growcut thereby validating the usefulness of the proposed method. The investigation of tumor type showed better performance for HGG, likely due to the imbalanced training data distribution (76 HGG/15 LGG). It could also indicate the presence of a measurable pathologic difference. If so, the training of a segmentation method for each type could lead to improved segmentations for both types. This would require knowing the tumor type in advance, information that was not readily available for the blinded challenge data. Our challenge results showed a nice performance although sub-par to the top two methods of the challenge but was superior to the remaining 11. It is noted that our methods performance is in the Dice score range that manual annotators can achieve according the results of [11]. They reported the Dice accuracy of annotators to be in the range of (0.74-0.85). This is comparable to the proposed method. A simple strategy for improving our work would be to extend the ensemble to use 3D network (computationally costly) or to investigate the inclusion of networks trained from more than orthogonal planes. In addition, the usage of using longitudinal information could also play a role towards improving segmentations.

References

1. Bergstra, J., et al.: Theano: a CPU and GPU math expression compiler. In: Python for Scientific Computing Conference (SciPy) (2010)
2. Cordier, N., Menze, B., Delingette, H., Ayache, N.: Patch-based segmentation of brain tissues. In: MICCAI-BraTS (Challenge on Multimodal Brain Tumor Segmentation), pp. 6–17 (2013)
3. Davy, A., Havaei, M., Warde-Farley, D., Biard, A., Tran, L., Jodoin, P.M., Courville, A., Larochelle, H., Pal, C., Bengio, Y.: Brain tumor segmentation with deep neural networks. In: MICCAI-BraTS, pp. 1–5 (2014)
4. Dice, L.R.: Measures of the amount of ecologic association between species. Ecology (1945)
5. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors. CoRR (2012)
6. Jakab, A.: Segmenting brain tumors with the slicer 3d software. Tech. rep., University of Debrecen / ETH Zürich (2012)
7. Kleesiek, J., Biller, A., Urban, G., Kothe, U., Bendszus, M., Hamprecht, F.: Ilastik for multi-modal brain tumor segmentation. In: MICCAI-BraTS, pp. 12–17 (2014)
8. Kwon, D., Akbari, H., Da, X., Gaonkar, B., Davatzikos, C.: Multimodal brain tumor image segmentation using glistr. In: MICCAI-BraTS, pp. 18–19 (2014)
9. Menze, B., Geremia, E., Ayache, N., Szekely, G.: Segmenting glioma in multimodal images using a generative-discriminative model for brain lesion segmentation. In: MICCAI-BraTS, pp. 56–63 (2012)
10. Menze, B., Leemput, K.V., Lashkar, D., Weber, M., Ayache, N., Golland, P.: Segmenting glioma in multi-modal images using a generative model for brain lesion segmentation, pp. 49–55 (2012)
11. Menze, B.H., et al.: The multimodal brain tumor image segmentation benchmark (brats). IEEE Transactions on Medical Imaging (2014)
12. Reza, S., Iftekharuddin, K.: Improved brain tumor tissue segmentation using texture features. In: MICCAI-BraTS, pp. 27–30 (2014)
13. Sutskever, I., Martens, J., Dahl, G.E., Hinton, G.E.: On the importance of initialization and momentum in deep learning. In: 30th International Conference on Machine Learning (ICML 2013), vol. 28, pp. 1139–1147, May 2013
14. Tustison, N.J., Avants, B.B., Cook, P.A., Zheng, Y., Egan, A., Yushkevich, P.A., Gee, J.C.: N4ITK: Improved N3 Bias Correction. IEEE Trans. Med. Imaging **29**(6), 1310–1320 (2010)
15. Tustison, N., Wintermark, M., Durst, C., Avants, B.: Ants and arboles. In: MICCAI-BraTS, pp. 47–50 (2013)

16. Urban, G., Bendszus, M., Hamprecht, F.A., Kleesiek, J.: Multi-modal brain tumor segmentation using deep convolutional neural networks. In: MICCAI-BraTSs, pp. 31–35 (2014)
17. Vezhnevets, V., Konouchine, V.: GrowCut - interactive multi-label n-d image segmentation by cellular automata. In: Proceedings of Graphicon (2005)
18. Zikic, D., Ioannou, Y., Brown, M., Criminisi, A.: Segmentation of brain tumor tissues with convolutional neural networks. In: MICCAI-BraTS, pp. 36–39 (2014)