

# Automatic Detection and Segmentation of Brain Tumor Using Random Forest Approach

Zoltán Kapás<sup>1</sup>, László Lefkovits<sup>1</sup>, and László Szilágyi<sup>1,2,3</sup>(✉)

<sup>1</sup> Sapientia University of Transylvania, Tîrgu-Mureş, Romania  
lalo@ms.sapientia.ro

<sup>2</sup> Department of Control Engineering and Information Technology,  
Budapest University of Technology and Economics, Budapest, Hungary

<sup>3</sup> University of Canterbury, Christchurch, New Zealand

**Abstract.** Early detection is the key of success in the treatment of tumors. Establishing methods that can identify the presence and position of tumors in their early stage is a current great challenge in medical imaging. This study proposes a machine learning solution based on binary decision trees and random forest technique, aiming at the detection and accurate segmentation of brain tumors from multispectral volumetric MRI records. The training and testing of the proposed method uses twelve selected volumes from the BRATS 2012/13 database. Image volumes were preprocessed to extend the feature set with local information of each voxel. Intending to enhance the segmentation accuracy, each detected tumor pixel is validated or discarded according to a criterion based on neighborhood information. A detailed preliminary investigation is carried out in order to identify and enhance the capabilities of random forests trained with information originating from single image records. The achieved accuracy is generally characterized by a Dice score up to 0.9. Recommendations are formulated for the future development of a complex, random forest based tumor detection and segmentation system.

**Keywords:** Decision tree · Random forest · Machine learning · Image segmentation

## 1 Introduction

The early detection of brain tumors is utmost important as it can save human lives. The accurate segmentation of brain tumors is also essential, as it can assist the medical staff in the planning of treatment and intervention. The manual segmentation of tumors requires plenty of time even for a well-trained expert. A fully automated segmentation and quantitative analysis of tumors is thus a highly beneficial service. However, it is also a very challenging one, because of the high variety of anatomical structures and low contrast of current imaging

---

Research supported in part by Marie Curie Actions, IRSES Project no. FP7-318943.

techniques which make the difference between normal regions and the tumor hardly recognizable for the human eye [1]. Recent solutions, usually assisted by the use of prior information, employ various image processing and pattern recognition methodologies like: combination of multi-atlas based segmentation with non-parametric intensity analysis [2], AdaBoost classifier [3], level sets [4], active contour model [5], graph cut distribution matching [6], diffusion and perfusion metrics [7], 3D blob detection [8], support vector machine [9], concatenated random forests [10,11], and fuzzy  $c$ -means clustering [12].

The main goal of our research work is to build a reliable procedure for brain tumor detection from multimodal MRI records, based on supervised machine learning techniques, using the MICCAI BRATS data set that contains several dozens of image volumes together with ground truth provided by human experts. In this paper we propose a solution based on binary decision trees and random forest, and present preliminary results together with a recommendations towards a complex and reliable brain tumor detection system.

## 2 Materials and Methods

The main goal of this study is to establish a machine learning solution to detect and localize tumors in MRI volumes. This paper presents preliminary results obtained using the random forest technique. The algorithm is trained to separate three tissue types, which are labeled as tumor, edema, and negative. The primary focus is on establishing the presence or absence of the tumor, while the accurate segmentation is a secondary goal.

### 2.1 BRATS Data Sets

Brain tumor image data used in this work were obtained from the MICCAI 2012 Challenge on Multimodal Brain Tumor Segmentation [13]. The challenge database contains fully anonymized images originating from the following institutions: ETH Zürich, University of Bern, University of Debrecen, and University of Utah. The image database consists of multi-contrast MR scans of 30 glioma patient, out of which 20 have been acquired from high-grade (anaplastic astrocytomas and glioblastoma multiform tumors) and 10 from low-grade (histological diagnosis: astrocytomas or oligoastrocytomas) glioma patients. For each patient, multimodal (T1, T2, FLAIR, and post-Gadolinium T1) MR images are available. All volumes were linearly co-registered to the T1 contrast image, skull stripped, and interpolated to 1 mm isotropic resolution. Each records contains approximately 1.5 millions of feature vectors. All images are stored as signed 16-bit integers, but only positives values are used. Each image set has a truth image which contains the expert annotations for “active tumor” and “edema”. Each voxel in a volume is represented by a four-dimensional feature vector:

$$\mathbf{x} = [x^{(T1)}, x^{(T2)}, x^{(T1C)}, x^{(FLAIR)}]^T. \quad (1)$$

These are the observed features of each pixel. Since these four values do not incorporate any information regarding the location or the neighborhood of the pixel, there is a strong need to extend the feature vector with further, computed features.

## 2.2 Data Preprocessing

Preprocessing steps in our application have three main goals.

1. Histogram normalization. Whether we like it or not, absolute intensity values in magnetic resonance imaging say nothing about the observed tissue. Intensities are relative and frequently contaminated with intensity inhomogeneity [14]. Treating the latter problem stays outside the scope of this study, as the MICCAI BRATS data set is free from intensity inhomogeneity. However, the histogram of each volume needs to be mapped on a uniform scale. In this order, all intensity values underwent a linear transformation  $x \rightarrow \alpha x + \beta$ , where parameters  $\alpha$  and  $\beta$  were established separately for each feature such a way that the middle fifty percent of the data fell between 600 and 800 after the transformation. Further on, we set up a minimum and maximum limit for intensity values at 200 and 1200, respectively. Intensities situated beyond the limit were replaced by the corresponding limit value.
2. Computed features. Since the observed data vectors bear no information on the position of the pixel, we included eight more features into the feature vector. For each of the four channels, two locally averaged intensities were computed within 10-element and 26-element neighborhood. The former contained eight direct neighbors within the slice and the two closest ones from neighbor slices. The latter contained all neighbors of the given pixel situated within a  $3 \times 3 \times 3$ -sized cube. Pixels having no valid neighbors in the specific neighborhood inherited the own intensity value of the pixel itself in the given channel.
3. Missing Data. Some pixels have zero valued observed features standing for a missing value. These pixels were not included in the main data processing. However, all existing features were used at the computation of averaged features, so pixels with missing values may have contributed to their neighbors, before being discarded.

## 2.3 Decision Tree

Binary decision trees (BDT) can describe any hierarchy of crisp (non-fuzzy) two-way decisions [15]. Given an input data set of vectors  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ , where  $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,m}]^T$ , a BDT can be employed to learn the classification that corresponds to any set of labels  $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ . The classification learned by the BDT can be perfect if  $\mathbf{x}_i = \mathbf{x}_j$  implies  $\lambda_i = \lambda_j$ ,  $\forall i, j \in \{1, 2, \dots, n\}$ . The BDT is built during the learning process. Initially the tree consists of a single node, the root, which has to make a decision regarding all  $n$  input vectors. If not all  $n$  vectors have the same label, which is likely to

be so, then the set of data is not homogeneous, there is a need for a separation. The decision will compare a single feature, the one with index  $k$  ( $1 \leq k \leq m$ ), of the input vectors with a certain threshold  $\alpha$ , and the comparison will separate the vectors into two subgroups: those with  $x_{i,k} < \alpha$  ( $i = 1 \dots n$ ), and those with  $x_{i,k} \geq \alpha$  ( $i = 1 \dots n$ ). The root will then have two child nodes, each corresponding to one of the possible outcomes of the above decision. The left child will further classify those  $n_1$  input vectors, which satisfied the former condition, while the right child those  $n_2$  ones that satisfied the latter condition. Obviously, we have  $n_1 + n_2 = n$ . For both child nodes, the procedure is the same as it was for the root. When at a certain point of the learning algorithm, all vectors being classified by a node have the same label  $\lambda_p$ , then the node is declared a leaf node, which is attributed to the class with index  $p$ . Another case when a node is declared leaf node is when all vectors to be separated by the node are identical, so there is no possible condition to separate the vectors. In this case, the label of the node is decided by the majority of labels, or if there is no majority, a label should be chosen from the present ones. In our application, this kind of rare cases use the priority list of labels defined as: (1) tumor, (2) edema, (3) negative.

The separation of a finite set of data vectors always terminates in a finite number of steps. The maximum depth of the tree highly depends on the way of establishing the separation condition in each node. The most popular way, also employed in our application, uses entropy based criteria to choose the separation condition. Whenever a node has to establish its separation criterion for a subset of vectors  $\bar{\mathbf{X}} \subseteq \mathbf{X}$  containing  $\bar{n}$  items with  $1 < \bar{n} \leq n$ , the following algorithm is performed:

1. Find all those features which have at least 2 different values in  $\bar{\mathbf{X}}$ .
2. Find all different values for each feature and sort them in increasing order.
3. Set a threshold candidate at the middle of the distance between each consecutive pair of values for each feature.
4. Choose that feature and that threshold, for which the entropy-based criterion

$$E = \bar{n}_1 \log \frac{\bar{n}_1}{\bar{n}} + \bar{n}_2 \log \frac{\bar{n}_2}{\bar{n}} \quad (2)$$

gives the minimum value, where  $\bar{n}_1$  ( $\bar{n}_2$ ) will be the cardinality of the subset of vectors  $\bar{\mathbf{X}}_1$  ( $\bar{\mathbf{X}}_2$ ), for which the value of the tested feature is less than (greater or equal than) the tested threshold value.

After having the BDT trained, it can be applied for the classification of test data vectors. Any test vector is first fed to the root node, which according to the stored condition and the feature values of the vector, decides towards which child node to forward the vector. This strategy is followed then by the chosen child node, and the vector will be forwarded to a further child. The classification of a vector terminates at the moment when it is forwarded to a leaf node of the tree. The test vector will be attributed to the class indicated by the labeling of the reached leaf node.

## 2.4 Random Forest

A binary decision tree is an excellent tool, when the task is to accurately learn a certain complicated pattern. For example, it can reproduce every little detail of any MRI volume applied as training data, while keeping the maximum depth below one hundred. However, this marvellous property drags along a serious danger of overfitting. Learning all the small details of the train data builds a serious obstacle for the decision tree in making correct decisions concerning major properties of the test data. This is why, we followed the recipe of Breiman [16], and built forests of binary decision trees, using randomly chosen subsets of the learning data, and randomly chosen subset of features for each tree separately. Each tree in a random forest is a weak classifier. A large set of trees trained with randomly chosen data will make a single decision on a majority basis. In the current stage of this research, we tested how accurate decisions can be made by random forests trained by the data coming from a single MRI volume. There were several important questions to answer:

1. What is the right number of trees in a random forest? Too few trees are not likely to be accurate, while too many redundant ones will not be runtime efficient.
2. What is the right number of feature vectors to train each tree in the random forest? Again, too few vectors are not expected to lead to accurate decision, while too many vectors bring the risk to overfitting.
3. How to make a random forest accurate and effective, when being trained with data coming from several MRI volumes?

## 2.5 Post-Processing

Random forests are expected to identify the most part of vectors describing tumor pixels. Since negative pixels belong to a great variety of normal tissues (e.g. white matter, gray matter, cerebro-spinal fluid), some of them might be classified as tumor or edema. To be able to discard such cases, we proposed and implemented a posterior validation scheme for all pixels that are labeled as tumor or edema by the random forest. For each such pixel, we defined a 250-pixel neighborhood (all pixels situated at Euclidean distance below  $\sqrt{15}$  units, and counted how many of the neighbors are classified as tumor or edema. Those having a number of such neighbors below the predefined neighborhood threshold, are relabeled as negative pixels during post-processing. The appropriate value of the threshold is to be established as well.

## 2.6 Evaluation of Accuracy

The Jaccard index (JI) is a normalized score of accuracy, computed as

$$\text{JI} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \quad (3)$$

where TP stands for the number of true positives, FP for the false positives, and FN for false negatives. Further on, the Dice score (DS) can be computed as

$$DS = \frac{2TP}{2TP + FP + FN} = \frac{2JI}{1 + JI}. \quad (4)$$

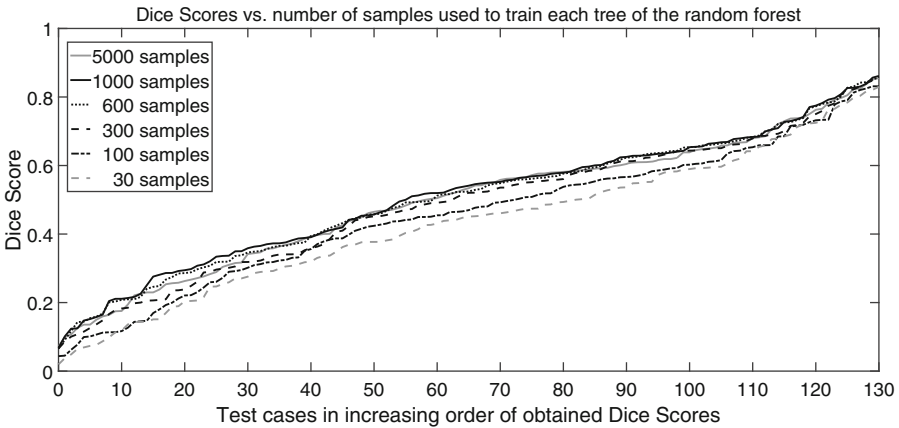
Both indices score 1 in case of an ideal clustering, while a fully random result is indicated by a score close to zero.

### 3 Results and Discussion

Twelve volumes from the BRATS 2012/13 data set were selected for the evaluation of the proposed methodology:

$$\mathcal{V} = \{HG01, HG02, \dots, HG07, HG09, HG11, HG13, HG14, HG15\}. \quad (5)$$

Let us denote by  $DS(i \rightarrow j)$  ( $i, j \in \mathcal{V}$ ) the Dice score given by the random forest trained with data chosen from volume  $i$  while tested on the whole volume  $j$ . Considering the size of the set  $\mathcal{V}$ , there are  $12 \times 11 = 132$  possible  $i \neq j$  scenarios, and 12 ones with  $i = j$ . We performed all possible such tests with various settings of main parameters like number of trees in the forest and number of samples used for the training of each tree. At the training of individual decision trees, an equal number of random samples were chosen from each of the three tissue types. For example, the so-called 100-sample training refers to the use of a total number of  $3 \times 100 = 300$  feature vectors.



**Fig. 1.** Classification accuracy benchmarks in case of random forests containing 100 trees, each tree trained with  $3 \times (30 \text{ to } 5000)$  feature vectors. Exceptionally, training with HG13 used 4000 samples instead of 5000, because it has less than 5000 tumor pixels.

Figure 1 exhibits the obtained Dice scores in case of trees trained by sample sizes varying from 30 to 5000 items. Each forest in this experiment consisted of 100 trees. For each type of trees, the obtained  $DS(i \rightarrow j)$  ( $i \neq j$ ) Dice scores were sorted in increasing order. The obtained curves indicate that the sample size can strongly influence the classification accuracy. Generally the larger the training sample, the more accurate the decisions, but at a certain level above 1000 samples per tissue type, traces of overfitting are observed. Table 1 shows numerical values of the obtained average Dice scores. The last column also reveals how accurate the classification can be when tested on the same volume that was used for training. Obviously, overfitting does not disturb classification accuracy on the train data set.

**Table 1.** Averaged accuracy benchmark scores obtained by forests of 100 trees each

Samples from each tissue type	Mean Dice score $DS(i \rightarrow i)$	Percentage of Dice score obtained for 1000 samples	Mean Dice score $DS(i \rightarrow i)$
5000 samples	0.4916	98.33 %	0.8124
1000 samples	0.5030	100.00 %	0.7810
600 samples	0.4976	98.93 %	0.7702
300 samples	0.4809	95.62 %	0.7626
100 samples	0.4510	89.66 %	0.7354
30 samples	0.4286	85.21 %	0.6912

**Table 2.** Dice scores obtained when training with one volume and testing on another, trees of 1000 samples per tissue type, and 100 trees in the forest

Volume $i$	Train data selected from volume $i$ Testing on each volume $j \neq i$				Testing on volume $j$ When trained on each volume $i \neq j$			
	Average	SD	Maximum	Minimum	Average	SD	Maximum	Minimum
HG01	0.5216	0.1722	0.7822	0.2133	0.5857	0.1627	0.8304	0.2508
HG02	0.6191	0.1280	0.8262	0.3500	0.4637	0.2117	0.7416	0.1013
HG03	0.5721	0.1767	0.8615	0.2945	0.4532	0.2309	0.8082	0.1579
HG04	0.5107	0.1983	0.8304	0.2043	0.3902	0.1502	0.6377	0.1474
HG05	0.3872	0.2188	0.6652	0.0552	0.3848	0.1235	0.5279	0.1519
HG06	0.5648	0.1537	0.8386	0.3113	0.5918	0.0955	0.7273	0.6980
HG07	0.5815	0.1537	0.8270	0.3368	0.4796	0.1666	0.7052	0.2110
HG09	0.2662	0.1626	0.5126	0.0684	0.4659	0.1230	0.5789	0.2043
HG11	0.4939	0.2064	0.8564	0.2111	0.5718	0.1320	0.7090	0.2863
HG13	0.5630	0.1992	0.8082	0.2758	0.3368	0.2351	0.6995	0.0552
HG14	0.4721	0.1804	0.7299	0.1248	0.5628	0.1621	0.8270	0.3340
HG15	0.4837	0.1359	0.6426	0.2255	0.7493	0.1144	0.8615	0.5000

Table 2 gives a detailed statistical report on obtained  $DS(i \rightarrow j)$  values, for each individual image volume. The left panel summarizes benchmark values for

cases when the given volume served as training data set, and the forest was tested on all other volumes. The right panel reports testing on the given volume, having forests trained with all other volumes separately. The two panels are far from being symmetric, as the best performing train data sets were HG02, HG07, and HG03, while highest accuracy benchmarks were obtained when testing on volumes HG15, HG06, and HG01. The minimum values indicate that data from a single volume cannot train a forest for high quality classification. On the other hand, the maximum values show that data from each volume can contribute to the classification accuracy, and for each test data set there exist possible training sets that yield acceptable classification.

Another aspect that deserves to be remarked and analysed is the poor symmetry of the obtained Dice scores. Having obtained a high value for a certain  $DS(i \rightarrow j)$  does not necessarily mean that  $DS(j \rightarrow i)$  will also have a high value. In order to numerically characterize the symmetry of the obtained results, we propose to compute the Averaged Symmetry Criterion (ASC), defined as:

$$ASC = \exp \left( \frac{1}{|\mathcal{V}|(|\mathcal{V}| - 1)} \sum_{i,j \in \mathcal{V}; i \neq j} \left| \log \frac{DS(i \rightarrow j)}{DS(j \rightarrow i)} \right| \right), \quad (6)$$

where  $|\mathcal{V}|$  stands for the cardinality of  $\mathcal{V}$ , namely 12 in our case. ASC values obtained for various train samples sizes are reported in Table 3. Dices scores seem to be closest, but still very far from symmetry at sample sizes that assure highest accuracy.

For certain couples of different volumes  $(i, j)$ , we performed 30 repeated training and testing processes. The goal was to monitor the variability of Dice scores  $DS(i \rightarrow j)$  obtained due to the random samples used for training. Figure 2 presents the outcome of repetitive evaluation. Seemingly using less samples means higher variance in benchmark results.

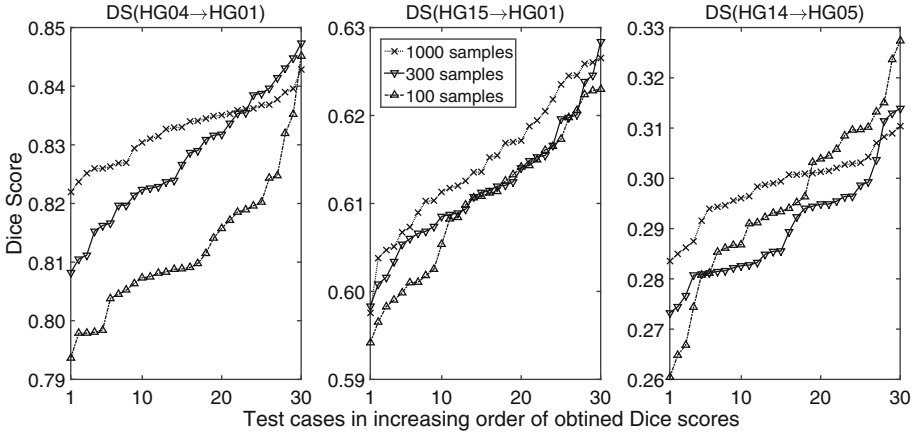
The applied post-processing scheme led to relevant improvement of classification accuracy. Figure 3 shows the histogram of all  $DS(i \rightarrow j)$  values before and after post-processing. Here the train data consisted of 600 randomly chosen samples per tissue type for each of the 100 trees in the forest. Dice scores after post-processing reported here are the maximum values obtained by choosing the optimal neighborhood threshold for each individual case. However, this cannot be done automatically. We need to establish either an acceptable constant value of the neighborhood threshold, or to define a strategy that sets the threshold while testing.

Figure 3 also reports the effect of the post-processing. On the left side each individual test is represented, showing the DS before and the maximum DS

**Table 3.** The relation between  $DS(i \rightarrow j)$  and  $DS(j \rightarrow i)$ , for  $i, j \in \mathcal{V}$  and  $i \neq j$

Train sample size per tissue type	30	100	300	600	1000	5000
Symmetry benchmark (ASC)	2.065	1.960	1.792	1.705	1.670	1.710





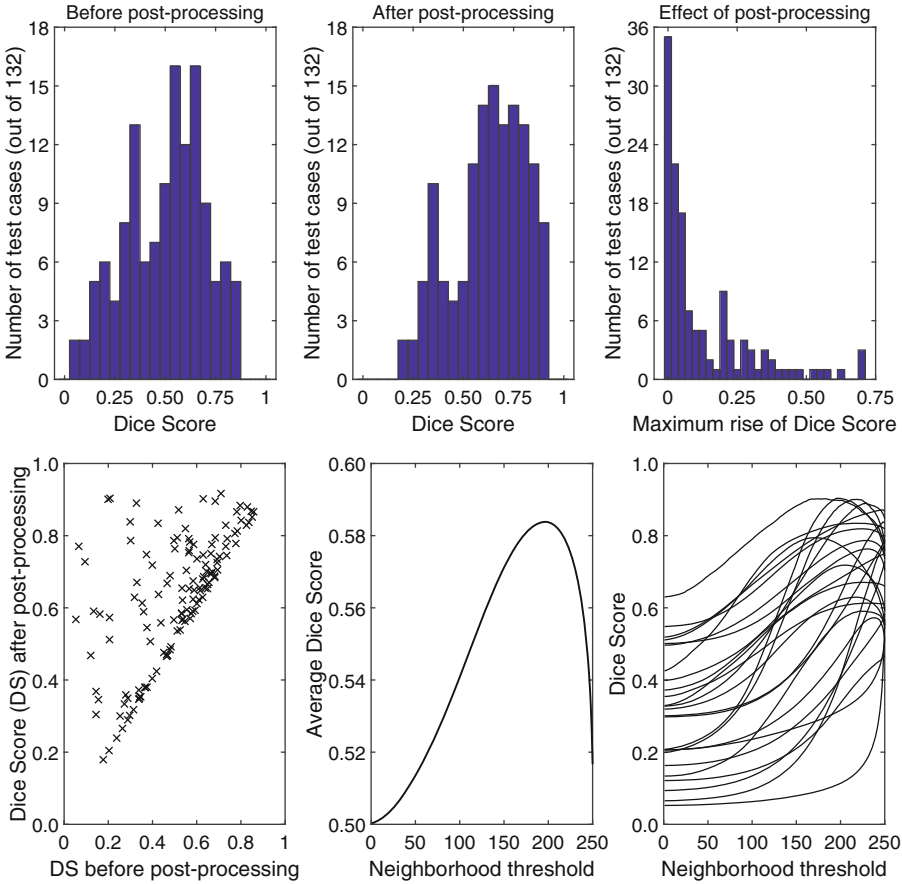
**Fig. 2.** Reproducibility benchmark: outcome of repeated training on random data samples from a given volume and testing on another volume.

after post-processing. The single curve in the middle plot presents the average effect of post-processing for each possible value of the neighborhood threshold, indicating that it is possible to choose such a threshold value between 190 and 200, for which the average DS rises from 0.502 to 0.583. The bottom right side of Fig. 3 shows those 25 test cases, which were most favorably affected by the post-processing.

Figure 4 shows the outcome of tumor segmentation without and with post-processing, by presenting detected and missed tumor pixels in several consecutive slices of volume HG11. The forest used here consisted of 100 trees, and each tree was trained using 600 samples of each tissue type, randomly selected from volume HG15. In this image, black pixels are the true positive ones, while gray shades represent false positives and false negatives. Post-processing in this certain case rose the Dice score from 0.5339 to 0.8036, which was achieved by discarding lots of false positives, mostly in slices where the real tumor was not present. Even this result could be further improved by implementing another post-processing step that would detect non-tumor (gray) pixels inside the tumor (among black pixels).

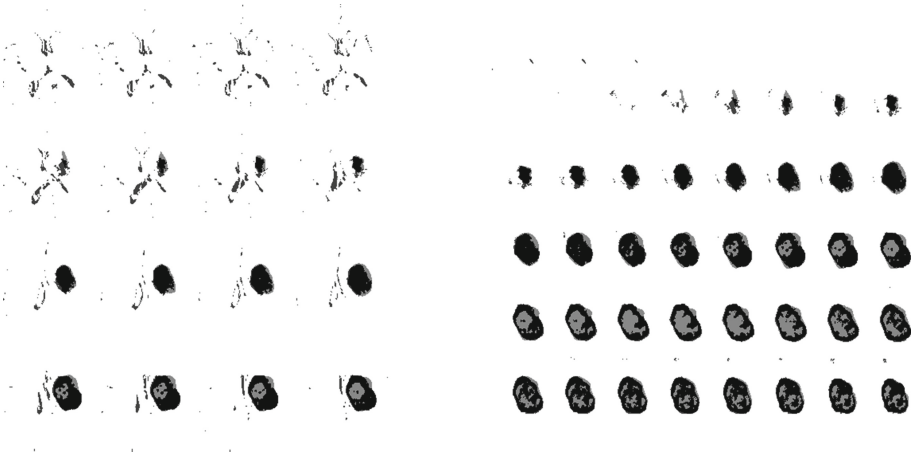
The size of tumors that are present in the volumes included in  $\mathcal{V}$  varied from  $4.5\text{ cm}^3$  in volume HG13 to  $110\text{ cm}^3$  in volume HG14. The segmentation accuracy of tumors depends on the size of the tumor, as indicated in Fig. 5. The post-processing seems to help more in case of small tumors, so it has a vital role in detecting early stage tumors.

The experiments carried out during this study showed us that a random forest trained with samples from a single volume cannot perform acceptably in all cases. On the other hand, each tested volume had one or more corresponding train volumes that assured fine detection and accurate segmentation of the tumor. The latter allows us to envision a complex random forest that will be suitable for a great majority of cases, which will be reliable enough for clinical deployment.

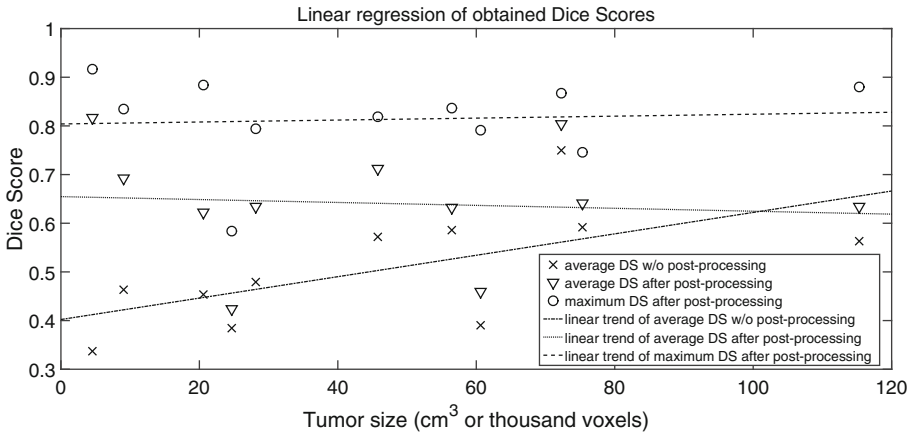


**Fig. 3.** Effects of the proposed post-processing. Histogram of Dice scores  $DS(i \rightarrow j)$ : (top left) before post-processing; (top middle) after post-processing; (top right) histogram of the differences caused by neighborhood-based post-processing; (bottom left) Individual  $DS(i \rightarrow j)$  values before and after post-processing; (bottom middle) The evolution of average Dice score plotted against the value of the neighborhood threshold; (bottom right) Variation of individual Dice scores  $DS(i \rightarrow j)$  plotted against the value of the neighborhood threshold, those 25 which are most affected by post-processing.

The future random forest solution will probably contain clusters of trees, where different clusters will be trained each using its dedicated reduced number of volumes. Clusters of trees will give their own opinion concerning test cases, and the forest will have the role to aggregate these individual opinions and produce the final positive or negative diagnosis. This preliminary study has shown that a random forest based learning algorithm, even if trained with a much more reduced number of features than other random forest based solutions (e.g. [10, 11]), can be suitable to detect the presence of the tumor.



**Fig. 4.** (left) Detected tumor without post-processing; (right) Detected tumor with neighborhood-based post-processing. Without validating each pixel classified as tumor, several scattered false positives are present in the volume.



**Fig. 5.** Obtained average and maximum Dice scores with or without post-processing, plotted against the size of the tumor in the test volume. Linear trends are also indicated.

## 4 Conclusion

In this paper we presented an automatic tumor detection and segmentation algorithm employing random forests and binary decision trees, in its preliminary stage of implementation. The proposed methodology already reliably detects tumors of 2 cm diameter. It is likely to obtain fine segmentation accuracy in the future using a complex random forest trained with data from dozens of volumes.

## References

1. Gordillo, N., Montseny, E., Sobrevilla, P.: State of the art survey on MRI brain tumor segmentation. *Magn. Reson. Imaging* **31**, 1426–1438 (2013)
2. Asman, A.J., Landman, B.A.: Out-of-atlas labeling: a multi-atlas approach to cancer segmentation. In: 9th IEEE International Symposium on Biomedical Imaging, pp. 1236–1239. IEEE Press, New York (2012)
3. Ghanavati, S., Li, J., Liu, T., Babyn, P.S., Doda, W., Lampropoulos, G.: Automatic brain tumor detection in magnetic resonance images. In: 9th IEEE International Symposium on Biomedical Imaging, pp. 574–577. IEEE Press, New York (2012)
4. Hamamci, A., Kucuk, N., Karamam, K., Engin, K., Unal, G.: Tumor-Cut: segmentation of brain tumors on contrast enhanced MR images for radiosurgery applications. *IEEE Trans. Med. Imaging* **31**, 790–804 (2012)
5. Sahdeva, J., Kumar, V., Gupta, I., Khandelwal, N., Ahuja, C.K.: A novel content-based active contour model for brain tumor segmentation. *Magn. Reson. Imaging* **30**, 694–715 (2012)
6. Njeh, I., Sallemi, L., Ben Ayed, I., Chtourou, K., Lehericy, S., Galanaud, D., Ben Hamida, A.: 3D multimodal MRI brain glioma tumor and edema segmentation: a graph cut distribution matching approach. *Comput. Med. Imaging Graph.* **40**, 108–119 (2015)
7. Svolos, P., Tsolaki, E., Kapsalaki, E., Theodorou, K., Fountas, K., Fezoulidis, I., Tsougos, I.: Investigating brain tumor differentiation with diffusion and perfusion metrics at 3T MRI using pattern recognition techniques. *Magn. Reson. Imaging* **31**, 1567–1577 (2013)
8. Yu, C.P., Ruppert, G., Collins, R., Nguyen, D., Falcao, A., Liu, Y.: 3D blob based brain tumor detection and segmentation in MR images. In: 11th IEEE International Symposium on Biomedical Imaging, pp. 1192–1197. IEEE Press, New York (2012)
9. Zhang, N., Ruan, S., Lebonvallet, S., Liao, Q., Zhou, Y.: Kernel feature selection to fuse multi-spectral MRI images for brain tumor segmentation. *Comput. Vis. Image Understand* **115**, 256–269 (2011)
10. Tustison, N.J., Shrinidhi, K.L., Wintermark, M., Durst, C.R., Kandel, B.M., Gee, J.C., Grossman, M.C., Avants, B.B.: Optimal symmetric multimodal templates and concatenated random forests for supervised brain tumor segmentation (simplified) with ANTsR. *Neuroinformatics* **13**, 209–225 (2015)
11. Pinto, A., Pereira, S., Dinis, H., Silva, C.A., Rasteiro, D.: Random decision forests for automatic brain tumor segmentation on multi-modal MRI images. In: 4th IEEE Portuguese BioEngineering Meeting, pp. 1–5. IEEE Press, New York (2015)
12. Szilágyi, L., Lefkovits, L., Iantovics, B., Iclanzan, D., Benyó, B.: Automatic brain tumor segmentation in multispectral MRI volumetric records. In: Arik, S., Huang, T., Lai, W.K., Liu, Q. (eds.) *ICONIP 2015*. LNCS, vol. 9492, pp. 174–181. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-26561-2\\_21](https://doi.org/10.1007/978-3-319-26561-2_21)
13. Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., et al.: The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Trans. Med. Imaging* **34**, 1993–2024 (2015)
14. Vovk, U., Pernuš, F., Likar, B.: A review of methods for correction of intensity inhomogeneity in MRI. *IEEE Trans. Med. Imaging* **26**, 405–421 (2007)
15. Akers, S.B.: Binary decision diagrams. *IEEE Trans. Comput.* **C-27**, 509–516 (1978)
16. Breiman, L.: Random forests. *Mach. Learn.* **45**, 5–32 (2001)