Using separated inputs for multimodal brain tumor segmentation with 3D U-Net-like architectures

N. Boutry[1][0000−0001−6278−4638], J. Chazalon[1][0000−0002−3757−074X],
E. Puybareau[1][0000−0002−2748−6624], G. Tochon[1][0000−0003−4617−4922],
H. Talbot[2][0000−0002−2179−3498], and T. Géraud[1][0000−0002−0380−7948]

[1] EPITA Research and Development Laboratory (LRDE)
https://lrde.epita.fr
[2] CentraleSupelec - Université Paris-Saclay
https://hugues-talbot.github.io/

Abstract. The work presented in this paper addresses the MICCAI BraTS 2019 challenge devoted to brain tumor segmentation using magnetic resonance images. For each task of the challenge, we proposed and submitted for evaluation an original method. For the tumor segmentation task (Task 1), our convolutional neural network is based on a variant of the U-Net architecture of Ronneberger et al. with two modifications: first, we separate the four convolution parts to decorrelate the weights corresponding to each modality, and second, we provide volumes of size $240 * 240 * 3$ as inputs in these convolution parts. This way, we profit of the 3D aspect of the input signal, and we do not use the same weights for separate inputs. For the overall survival task (Task 2), we compute explainable features and use a kernel PCA embedding followed by a Random Forest classifier to build a predictor with very few training samples. For the uncertainty estimation task (Task 3), we introduce and compare lightweight methods based on simple principles which can be applied to any segmentation approach. The overall performance of each of our contribution is honorable given the low computational requirements they have both for training and testing.

Keywords: biomedical imaging · brain tumor segmentation · glioblastoma · CNN · U-Net

1 Introduction

The work presented in this paper was realized in the context of MICCAI BraTS 2019 Challenge [12, 3, 4, 2, 1], which aims at stimulating brain tumor detection, segmentation and analysis. This challenge is composed of 3 tasks, and we propose a contribution for each of them which will be described in separate sections.

Task 1 – Tumor segmentation Given a set of unknown brain scans with four modalities, segment tumoral regions. We propose a deep architecture which fully decorrelates each modality with partial 3D convolutions.

Task 2 – Survival prediction Predict the patient overall survival time. We propose a predictor based on kernel PCA, Random Forests and a custom brain atlas.

Task 3 – Quantification of uncertainty in segmentation Assess how reliable the results from Task 1 are. We propose a set of lightweight techniques based on intrinsic confusion and geometry properties of the segmentation.

2 Brain Tumor Segmentation — Task 1

Starting from a set of 335 brain images where tumors are segmented by neuroradiologists, the aim of Task 1 is to segment new brain images whose ground truth is not known. The provided modalities are magnetic resonance images (T1/T1CE/T2 and FLAIR). The resolution of the provided images is $240 * 240 * 155$ voxels of 1 mm^3. These images result from captures of different protocols, magnetic fields strengths and MRI scanners.

Previous work For BraTS 2018 challenge, the first place was won by Myronenko [13] who used a semantic segmentation network based on a encoder-decoder architecture. Due to limited training dataset size, he connected a variational auto-encoder (able to reconstruct the initial image) to this network during the training procedure. This way, some constraints are added on the layers of the shared encoder which is in some way "regularized" and also less sensible to the random initialization. A crop size of 160 x 192 x 128 has been used, which implied a batch size of 1 due to GPU memory limitations. Isensee *et al.* [7] won the second place and proved that a U-Net-like architecture with slight modifications (like using the LeakyReLU instead of the usual ReLU activation function and using instance normalization [18]) can be very efficient and hard to beat. They used a batch size of 2, a crop size of 128 x 128 x 128, and a soft Dice loss function [7]. They also used an additional training data from their own institution to optimize the enhancing tumor dice. McKinly *et al.* [11] shared the third place with Zhou *et al.* [20]. On one side, McKinly *et al.* [11] proposed an embedding of a DenseNet [6] structure using dilated convolutions into a U-Net [15] architecture, to obtain their segmentation CNN. On the other side, Zhou *et al.* [20] ensembled different networks in cascade.

For the BraTS 2017 challenge, the first place was won by Kamnitsas *et al.* [8] who ensembled several models (trained separately) for robust segmentation (EMMA): they combined DeepMedic [9], FCN [10], and U-Net [15] models. During the training procedure, they used a batch size of 8 and a crop size of 64 x 64 x 64 3D patch. Wang *et al.* [19] won the second place. They segmented tumor regions in cascade using anisotropic dilated convolutions with 3 networks for each tumor subregion.

Proposed architecture Because the U-Net architecture [15] has demonstrated good performance in matter of biomedical image analysis, we propose here to

Table 1. Our U-Net-like multimodal 3D architecture, with 4 contractive branches.

Layer name	**Operation**	**Output shape**	**Input(s)**	
mod1_input	Input	240, 240, 3, 1		
mod1_conv1-1	Conv3D	240, 240, 3, 64	mod1_input	
mod1_conv1-2	Conv3D	240, 240, 3, 64	mod1_conv1-1	
mod1_conv2	BLOCK_A	120, 120, 3, 128	mod1_conv1-2	
mod1_conv3	BLOCK_A	60, 60, 3, 256	mod1_conv2	
mod1_conv4	BLOCK_A	30, 30, 3, 512	mod1_conv3	
mod1_conv5	BLOCK_A	15, 15, 3, 1024	mod1_conv4	
The branch for mod1 is repeated for each input (modality).				
concatenate_1	Concatenate	15, 15, 3, 4096	mod$i$_conv5	$\forall i \in [1, 4]$
up_samp3d	UpSampling3D	30, 30, 3, 4096	concatenate_1	
conv3d_1	Conv3D	30, 30, 3, 512	up_samp3d	
conv3d_2	BLOCK_B	60, 60, 3, 256	mod$i$_conv4 conv3d_1	$\forall i \in [1, 4]$
conv3d_3	BLOCK_B	120, 120, 3, 128	mod$i$_conv3 conv3d_2	$\forall i \in [1, 4]$
conv3d_4	BLOCK_B	240, 240, 3, 64	mod$i$_conv2 conv3d_3	$\forall i \in [1, 4]$
concatenate_2	Concatenate	240, 240, 3, 320	mod$i$_conv1-2 conv3d_4	$\forall i \in [1, 4]$
conv3d_5	Conv3D	240, 240, 3, 64	concatenate_2	
conv3d_6	Conv3D	240, 240, 3, 64	conv3d_5	
conv3d_7	Conv3D	240, 240, 3, 4	conv3d_6	
output	Conv3D	240, 240, 3, 4	conv3d_7	*k:* $1 \times 1 \times 1$

Table 2. Detail of the contractive block BLOCK_A.

Layer name	**Operation**	**Output shape**	**Input(s)**	
b1_input	Input	H, W, 3, C		
b1_mp	MaxPooling3D	H/2, W/2, 3, C	b1_input	*pool:* $2 \times 2 \times 1$
b1_conv	Conv3D	H/2, W/2, 3, 2*C	b1_mp	
b1_output	Conv3D	H/2, W/2, 3, 2*C	b1_conv	

Table 3. Detail of the expanding block BLOCK_B.

Layer name	**Operation**	**Output shape**	**Input(s)**	
b2_input_mod$i$	Input	H, W, 3, C		$\forall i \in [1, 4]$
b2_input_prev	Input	H, W, 3, C		
b2_concatenate	Concatenate	H, W, 3, 5*C	b2_input_mod$i$ b2_input_prev	$\forall i \in [1, 4]$
b2_conv3d_1	Conv3D	H, W, 3, C	b2_concatenate	
b2_conv3d_2	Conv3D	H, W, 3, C	b2_conv3d_1	
b2_up_samp3d	UpSampling3D	2*H, 2*W, 3, C	b2_conv3d_2	*pool:* $2 \times 2 \times 1$
b2_output	Conv3D	2*H, 2*W, 3, C/2	b2_up_samp3d	

re-adapt this architecture for multimodal biomedical image analysis. The complete architecture of our network is detailed in Tables 1, 2 and 3. We associate each modality to one input in our network. Then, each input is followed with a sequence of five layers made of two successive convolutional layers plus a max pooling and a dropout layer (contractive paths). Then, from the bottleneck, we apply five deconvolutional layers, each made of an upscaling layer followed with two convolutional layers (expanding path). Finally, skip connections are used to connect the contractive path to the expanding path at each scale. Note that the number of skip connections is multiplied by a factor of four due to the structure of our network. To ensure continuity in the segmentation results, we propose also to provide partial volumes as inputs in our network (we use the *Conv3D* layers of Keras on volumes of size $W * H * 3$ with kernels of shape $3 \times 3 \times 3$).

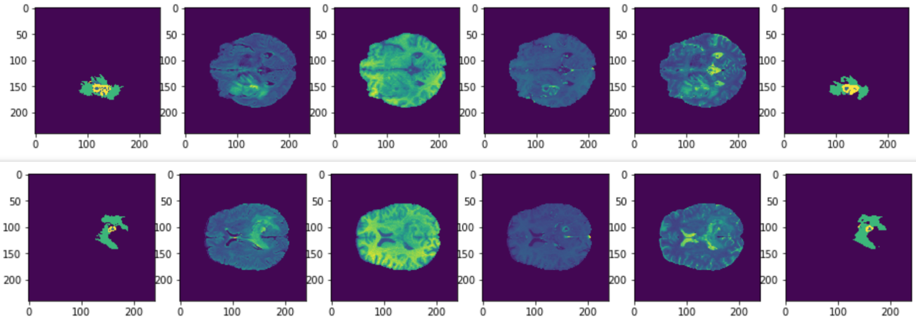


Fig. 1. Segmentation results with our architecture: on the left side, the ground truths, then the four modalities, and then our segmentation results.

Note that we know that the T1 modality and the T1CE one are strongly related, like the T2 one and the FLAIR one, but we are convinced that using separated weights for each inputs allows to improve segmentation results. This way we force the network to optimize different weights for each modality during the learning procedure.

Our motivation for our 3D approach (we provide partial volumes of size $3 * 240 * 240$) is twofold: first, the winners of the BraTS of 2018 used a full-3D approach [13], and second, we obtain smoother results thanks to the 3D convolutional layer (2D approaches generally lead to discontinuities along the z axis when slices are along x and y).

Note that we do not do any particular pre-processing, we just normalize each brain in the following manner like in [7]:

$$X_{norm} := \frac{X - \mu}{\sigma},$$

where μ and σ are respectively the statistical mean and standard deviation of the modality X corresponding to some patient. Also, we consider only the volumes (when we consitute the data set for the training procedure) where the number of voxels of the brain is greater than or equal to $(240 * 240 * 3)/6$. We do not use any post-processing.

Finally, we chose the standard parameters for our model: the number of filters are 64, 128, 256, 512, and 1024 for the 5 bi-convolutional layers, the number of filters are 512, 256, 128 and 64 for the bi-deconvolutional layers. Also, the learning rate is equal to 10^{-4}, we use categorical cross-entropy. We use the *selu* activation for all hidden layers, and use sigmoidal activation for the output layer.

Results Table 4 summarizes the results obtain by the proposed method on Task 1. At test time, segmentations are predicted on a single-pass without any augmentation. Furthermore, not post-processing was applied on the results we report. The proposed approach exhibits a reasonable performance regarding the computational constraints required for training: indeed, a single GPU card with 16 GB of memory was sufficient to conduct our experiments. The DICE measure suggests that for some volumes or for some specific areas, the method fails to detect the correct elements but succeeds most of the time. The Hausdorff measure suggests that the boundary of the detected regions are not very precise and that more regularization at inference time could improve the method.

Table 4. Mean values of the segmentation metrics for each region, for the validation set and the test set. ↑ (resp. ↓) indicates that a higher (resp. lower) value is better.

	DICE (%) ↑			Hausdorff95 (voxels) ↓		
Dataset	WT	TC	ET	WT	TC	ET
Validation	68.4	87.8	74.7	10.2	10.9	14.8
Test	73.7	86.2	75.1	5.6	10.7	15.4

3 Survival Prediction — Task 2

The second task of the MICCAI 2019 BraTS challenge is concerned with the prediction of patient overall survival from pre-operative scans (only for subjects with gross total resection (GTR) status). The classification procedure is conducted by labeling subjects into three classes: short-survivors (less than 10 months), mid-survivors (between 10 and 15 months) and long-survivors (greater than 15 months). For post-challenge analyses, prediction results are also compared in terms of mean and median square error of survival time predictions, expressed in days. For that reason, our proposed patient survival prediction algorithm is organized in two steps:

1 We first predict the overall survival class, *i.e.* short-, mid- or long-survival (hereafter denoted by class/label $1, 2$ and 3, respectively).

2 We then adjust our prediction within the predicted class by means of linear regression, in order to express the survival time in days.

Definition and extraction of relevant features Extracting relevant features is critical for classification purposes. Here, we re-use the features implemented by our team in the framework of the patient survival prediction task of MICCAI 2018 BraTS challenge, which ranked tie second [14]. Those features were chosen after in-depth discussions with a practitioner and are the following:

feature 1: the patient age (expressed in years).
feature 2: the relative size of the necrosis (labeled 1 in the groundtruth) class with respect to the brain size.
feature 3: the relative size of the edema class (labeled 2 in the groundtruth) with respect to the brain size.
feature 4: the relative size of the active tumor class (labeled 4 in the groundtruth) with respect to the brain size.
feature 5: the normalized coordinates of the binarized enhanced tumor (thus only considering necrosis and active tumor classes).
feature 6: the normalized coordinates of the region that is the most affected by necrosis, in a home made brain atlas.

For the training stage, features $2, 3$ and 4 are computed thanks to the patient ground truth map for each patient. As this information is unknown during the test stage, the segmented volumes predicted by our Deep FCN architecture are used instead. In any case, these size features are expressed relatively to the total brain size (computed as the number of voxels in the T2 modality whose intensity is greater than 0).

In addition, we also re-use the home-made brain atlas that we also developed for the 2018 BraTS challenge. This atlas is divided into 10 crudely designed regions accounting for the frontal, parietal, temporal and occipital lobes and the cerebellum for each hemisphere (see [14] for more details regarding this atlas and how it is adjusted to each patient brain size). Feature 6 is defined as the coordinates of the centroid of the region within the atlas that is the most affected by the necrosis class (*i.e.*, the region that has the most voxels labeled as necrosis with respect to its own size). Note that this feature, as well as feature 5, is then normalized relatively to the brain bounding box. This leads to a feature vector with 10 components per patient (since both centroids coordinates are 3-dimensionals).

Training phase For the training phase, we modified our previous work [14] in the following way: while we maintained the final learning stage through random forest (RF) classifiers [17], we replaced the principal component analysis (PCA) transformation, acting as preprocessing step for the learning stage, by its kernel counterpart (kPCA) [16]. The rationale is that we hope to increase the RFs performances in terms of classification/prediction as the input features are highly non-linear in terms of survival labels.

More specifically, the training stage of our prediction algorithm is as follows:

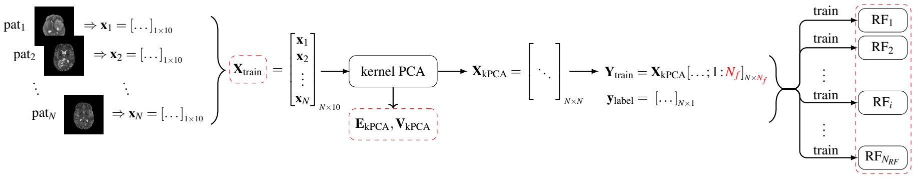

pat_1
$\Rightarrow \mathbf{x}_1 = [\ldots]_{1\times 10}$
pat_2
$\Rightarrow \mathbf{x}_2 = [\ldots]_{1\times 10}$
pat_N
$\Rightarrow \mathbf{x}_N = [\ldots]_{1\times 10}$
$\mathbf{X}_{\text{train}} = \begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\\ \vdots\\ \mathbf{x}_N\end{bmatrix}_{N\times 10}$
kernel PCA
$\mathbf{E}_{\text{kPCA}}, \mathbf{V}_{\text{kPCA}}$
$\mathbf{X}_{\text{kPCA}} = \begin{bmatrix} \ddots \end{bmatrix}_{N\times N}$
$\mathbf{Y}_{\text{train}} = \mathbf{X}_{\text{kPCA}}[\ldots; 1:N_f]_{N\times N_f}$
$\mathbf{y}_{\text{label}} = [\ldots]_{N\times 1}$
train
RF_1
train
RF_2
train
RF_i
train
$\text{RF}_{N_{RF}}$

Fig. 2. Workflow of the proposed class-based training procedure. The information stored after the training phase (necessary for the test phase) is written in red or encircled in dashed red.

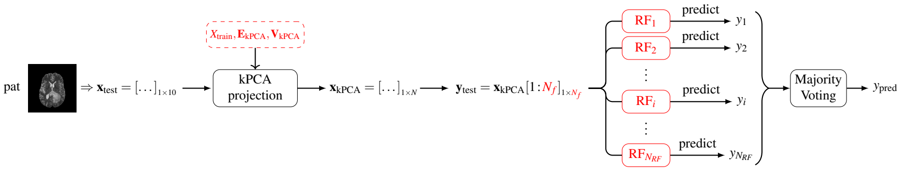

pat
$\Rightarrow \mathbf{x}_{\text{test}} = [\ldots]_{1\times 10}$
$X_{\text{train}}, \mathbf{E}_{\text{kPCA}}, \mathbf{V}_{\text{kPCA}}$
kPCA projection
$\mathbf{x}_{\text{kPCA}} = [\ldots]_{1\times N}$
$\mathbf{y}_{\text{test}} = \mathbf{x}_{\text{kPCA}}[1:N_f]_{1\times N_f}$
RF_1
predict
y_1
RF_2
predict
y_2
RF_i
predict
y_i
$\text{RF}_{N_{RF}}$
predict
$y_{N_{RF}}$
Majority Voting
y_{pred}

Fig. 3. Workflow of the proposed test procedure.

1. The feature vector $\mathbf{x}_i \in \mathbb{R}^{10}$ of each of the N patients in the training set is extracted as described in the previous section 3. All those feature vectors are then stacked in a $N \times 10$ feature matrix $\mathbf{X}_{\text{train}}$
2. A kPCA is performed on $\mathbf{X}_{\text{train}}$, yielding the $N \times N$ matrix $\mathbf{X}_{\text{kPCA}}$. This matrix is obtained through the computation, normalization and diagonalization of the so-called *kernel matrix* which represents the dot product between the N features vectors when mapped in the feature space through a kernel function (here defined as a polynomial kernel with degree $d = 3$).
4. The $N \times N_f$ matrix $\mathbf{Y}_{\text{train}}$ is defined from $\mathbf{X}_{\text{train}}$ by retaining the first N_f columns (corresponding to the leading N_f features in the feature space, here set to $N_f = 10$). N_{RF} RF classifiers [17] are finally trained on all rows of $\mathbf{Y}_{\text{train}}$ to learn to predict the survival class of each training patient using the true label vector $\mathbf{y}_{\text{label}}$ as target values. The used RF parameters (number of decision trees per RF, splitting criterion, total number of RFs N_{RF}) are defined as in [14].
5. Three linear regressors (one per survival class) are finally trained using the patient age and its whole tumor size (relatively to its brain size) as explanatory variables and its true survival time (expressed in days) as measured variable.

Steps 1. to 4. are depicted by the workflow in Fig.2. In addition to the three linear regressors, we also store (for the test phase) the training feature matrix $\mathbf{X}_{\text{train}}$, the eigenvector matrix V_{kPCA} and eigenvalues E_{kPCA} of the kernel matrix, and the number of retained features N_f after kPCA.

Table 5. Classification metrics of the proposed survival prediction method for the validation and test data sets, given the segmentation produced by our system for Task 1.

Data set	Accuracy	MSE	medianSE	stdSE	SpearmanR
Validation	0.414	158804	80437	194618	0.272
Test	0.505	464492	60237	1408716	0.363

Test phase The test phase is conducted in a similar fashion as the training phase. Given some input test patient, its overall survival class is first predicted, before being refined and expressed in terms of number of days. More specifically:

1. The features vector $\mathbf{x}_{\text{test}}$ of the test patient is retrieved as described previously.
2. This feature vector is then projected onto the principal axes learnt by the kPCA during the training phase. For that purpose, a new kernel matrix is computed and centered (hence the need for $\mathbf{X}_{\text{train}}$) before proper projection (through $\mathbf{V}_{\text{kPCA}}$) and scaling (with $\mathbf{E}_{\text{kPCA}}$).
3. This results in the projected vector $\mathbf{x}_{\text{kPCA}} \in \mathbb{R}^N$ from which the first N_f features are retained, yielding the test vector $\mathbf{y}_{\text{test}}$. This vector is then fed to the N_{RF} RF classifiers, leading to N_{RF} independent class label predictions. The final label prediction y_{pred} (1, 2 and 3 for short-, mid- and long-survivors, respectively) is eventually obtained by majority voting.
4. Once the survival class has been established, the final patient survival rate is predicted by means of the appropriate learnt linear regressor.

Steps 1. to 3. are illustrated by the workflow in Fig.3.

Results Table 5 presents the various classification performance metrics, namely the class-based accuracy, the mean, median and standard deviation square errors and Spearman R coefficient for survival predictions expressed in days, for the proposed prediction algorithm for the validation data set and the test data set. The validation and test data sets are comprised of $N = 27$ and $N = 107$ patients, respectively.
Results reported in Table 5 exhibit a slight improvement over the class-based classification accuracy between the validation set (0.414) and the test set (0.505).

4 Uncertainty Estimation in Segmentation — Task 3

The last task of the challenge is a new task which consists in estimating the uncertainty of the segmentation predictions produced in Task 1. The sub-regions considered for evaluation are: (i) the "enhancing tumor" (ET); (ii) the "tumor core" (TC); and (iii) the "whole tumor" (WT).

Participants had to produce uncertainty maps for each glioma sub-region. Each map contains integer values ranging from 0 (certain) to 100 (uncertain), and indicates the confidence of a decision to classify a particular voxel as belonging or not belonging to the a particular sub-region.

Results are reported using two metrics. (i) The area under the curve formed by the the DICE scores computed for each uncertainty threshold (DICE score computed only on voxels for which the uncertainty is strictly inferior to the current threshold). This metric is the principal metric used for ranking. (ii) The area under the curve formed by the ratio of filtered true positive for each uncertainty threshold (wrongly discarded as being uncertain).

Uncertainty Estimation Methods We focused on the study of lightweight uncertainty estimation techniques relying on two aspects of the predictions made by our segmentation system: (i) the consistency between independent predictions made for each classes; and (ii) the instability at the spatial boundary between two regions predicted as belonging to different classes. We believe that such approaches can be complementary to approaches based on the stability of the prediction under perturbations like Monte Carlo Dropout [5] which tend to be computationally demanding.

To take into account the consistency between independent predictions made for each classes, we propose a simple indicator called "weighted score difference" (abbreviated "WSDIFF") which estimates the uncertainty by computing the difference of activation between the most likely (maximally activated) class and the others, weighted by the absolute value of the greatest activation (in order to penalize cases where there is no clear activation of any class). This requires that the segmentation network outputs predictions for each class in an independent way (therefore it *cannot use a softmax* which would constrain predictions to be mutually exclusive).
Let c_i be the activation maps for each class i belonging to the sub-region R to consider, then the WSDIFF indicator for this sub-region R is computed as:

$$\mathrm{WSDIFF}_R = (1 - max(s_R, s_{\overline{R}})\,|s_R - s_{\overline{R}}|) * 100,$$

where:

$$s_R = max_{\forall i \in R}(c_i) \qquad \text{and} \qquad s_{\overline{R}} = max_{\forall i \notin R}(c_i).$$

SDIFF ("score difference") is the variant of this indicator without weighting:

$$\mathrm{SDIFF}_R = (1 - |s_R - s_{\overline{R}}|) * 100.$$

As shown later in the results, the weighting factor increased the performance of this indicator in our tests. Other attempts using the sum of the activation maps for each set of classes gave poor results and were harder to normalize.

Regarding the instability of the spatial boundary between two regions predicted as belonging to different classes, we designed an indicator (abbreviated "BORDER") which assigns a maximal uncertainty (100) at the boundary between two regions, and linearly decreases this uncertainty to the minimal value (0) at a given distance to the boundary. This distance defines the (half) width

of an “uncertainty border” between two regions.
It is calibrated independently for each class and was estimated with respect to the 95$^{\mathrm{th}}$ percentile of the Hausdorff distance metric reported for our segmentation method for this particular class. In practice, we used the following parameters: for the whole tumor (WT) we used a half-width of 9 voxels, for the tumor core (TC), 12 voxels, and for the enhancing tumor (ET), 7 voxels.
To compute this indicator, we first compute the Boundary Distance Transform $\mathrm{BDT} = max(\mathrm{DT}(R), \mathrm{DT}(\overline{R}))$ using the Distance Transform DT to the given sub-region R and its complement $\overline{R}$. Then, we invert, shift and clip the BDT such that the map is maximal on the boundary and have 0 values at a distance greater or equal to the half-width of the border. We finally scale the resulting map so its values are comprised between 0 (far from the boundary) and 100 (on the boundary). The resulting uncertainty map for a given class exhibits a triangular activation shape on the direction perpendicular to the boundary of the objects detected by the segmentation stage.

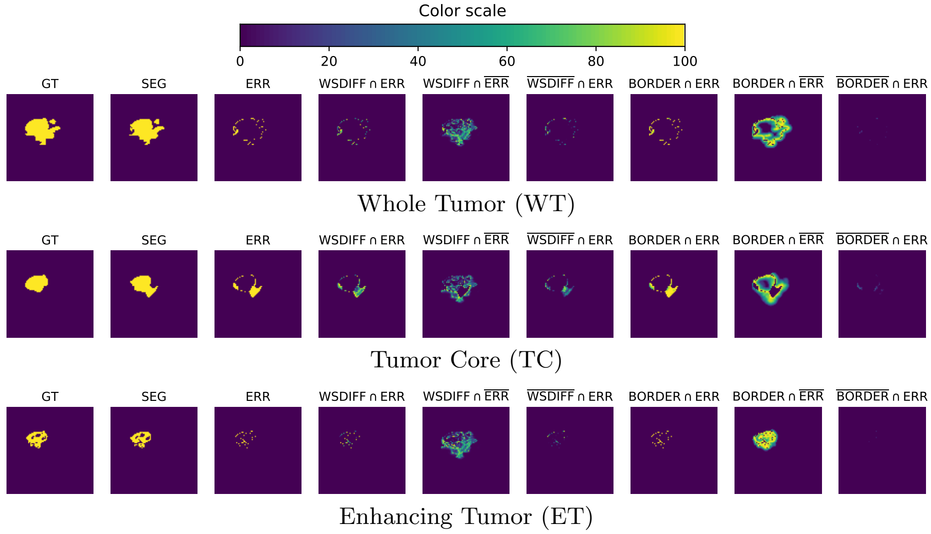
Results and Discussion Experimental results regarding the different uncertainty estimation methods are reported in Table 6. They indicate the results obtained for the validation set computed by the official competition platform.

Table 6. Mean values of the metrics for each region, computed by the official competition platform on the validation set for Task 3 (uncertainty estimation), for each of our uncertainty estimation approaches. ↑ (resp. ↓) indicates that a higher (resp. lower) value is better. Best values are in **bold** face.

	DICE_AUC (%) ↑			FTP_RATIO_AUC (%) ↓		
Metric	WT	TC	ET	WT	TC	ET
(original DICE score)	*85.2*	*62.0*	*59.9*	-	-	-
SDIFF	85.9	76.2	72.0	**20.7**	**17.8**	**16.8**
WSDIFF	85.9	77.5	**74.1**	30.7	24.0	21.4
BORDER	**87.8**	**80.6**	68.3	58.1	68.7	70.9
MEAN BORDER WSDIFF	86.7	79.5	73.3	44.1	45.7	45.9

Regarding the *DICE AUC* metric, the BORDER approach exhibits better results for glioma sub-regions WT and TW, while the WSDIFF approach performs better for the ET sub-region. The integration of a weighting of the uncertainty according to the activation of a given class provided some improvement to the WSDIFF method over the SDIFF one.
Regarding the *FTP Ratio AUC* metric, the BORDER method filters true positives quite aggressively and gives very high measures. The SDIFF method, on the other side of the spectrum, filters much less true positives. The WSDIFF method presents an interesting compromise in terms of true positive filtering. We can also notice that mean of the BORDER and WSDIFF indicators yields

some form of compromise (sub-optimal results, but less aggressive filtering). The best balance seems to use the BORDER indicator for WT and TC regions, and the WSDIFF indicator for ET regions: *this the strategy we used.*
Figure 4 illustrates the responses of the uncertainty estimation methods on a case for which the segmentation step performed reasonably well. We can see that while the BORDER method generates a lot of false positives, it successfully captures erroneous regions with a high uncertainty score. A better calibration of this method may improve its performance. For ET regions, the WSDIFF method is more selective and yields a lower amount of false positives.


Color scale
0
20
40
60
80
100
GT
SEG
ERR
WSDIFF ∩ ERR
WSDIFF ∩ $\overline{\text{ERR}}$
$\overline{\text{WSDIFF}}$ ∩ ERR
BORDER ∩ ERR
BORDER ∩ $\overline{\text{ERR}}$
$\overline{\text{BORDER}}$ ∩ ERR

Whole Tumor (WT)

Tumor Core (TC)

Enhancing Tumor (ET)

Fig. 4. Comparison of the WSDIFF and BORDER indicators on a reasonably well segmented case. Each row illustrates the response of uncertainty estimation methods for a different glioma region. The GT column is the ground truth, SEG the predicted segmentation, ERR the prediction error, and for each uncertainty estimation $\mathrm{METHOD} \in \{\mathrm{WSDIFF}, \mathrm{BORDER}\}$: $\mathrm{METHOD} \cap \mathrm{ERR}$ shows the uncertainty values for erroneous areas (true positives – higher is better), $\mathrm{METHOD} \cap \overline{\mathrm{ERR}}$ shows the uncertainty values for well-classified areas (false positives – lower is better), and $\overline{\mathrm{METHOD}} \cap \mathrm{ERR}$ shows the inverted $(100 - x)$ uncertainty values for erroneous areas (false negative – lower is better).

When comparing our results with other approaches from the public leader board for Task 3 (for the validation set), it should be noted that direct comparison is hard because the performance at Task 3 is directly linked to the performance at Task 1, hence a measure of a relative gain or loss might provide some hint, but ultimately each uncertainty estimator should be tested on every segmentation method. Nevertheless, we identified three interesting trends among those results. (1) Methods with high performance in both Task 1 and Task 3:

the uncertainty estimation may be great but the score at Task 3 is indubitably boosted by the one at Task 1. (2) Methods with average scores at Task 1 but with a noticeable improvement with respect to the DICE AUC score at Task 3: those methods seem to have an efficient uncertainty estimation strategy. Such methods may have: (2.1) a good score for the FTP Ratio AUC metric of Task 3, indicating an efficient approach; (2.2) an average score for this metric: we believe our approach belongs to this category.
Those results let us believe that our uncertainty estimation methods are better suited for cases were the underlying segmentation method already performs quite well. Because of their simplicity and fast computation, they may be a natural baseline for more complex methods to be compared against.

5 Conclusion

We proposed contributions for each task of the MICCAI BraTS 2019 challenge. For the tumor segmentation task (Task 1), our deep architecture based on a decorrelation of inputs and partial 3D convolutions exhibits an honorable performance given the fact the training can be performed on a single GPU with 16 GB of RAM. For the overall survival prediction task (Task 2), our approach based on a kernel PCA before using a random forest classifier provides an encouraging performance (given the few training examples available) while being based on explainable features. Finally, for the uncertainty estimation task (Task 3), we introduced and compared several lightweight methods which can be combined and could be better tuned to produce a less aggressive filtering.

Acknowledgments

We would like thank NVidia Corporation for their *Quadro P6000* GPU donation.

References

1. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J., Farahani, K., Davatzikos, C.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-GBM collection. The Cancer Imaging Archive **286** (2017)
2. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J., Farahani, K., Davatzikos, C.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-LGG collection. The Cancer Imaging Archive **286** (2017)
3. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.S., Freymann, J.B., Farahani, K., Davatzikos, C.: Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. Scientific data **4**, 170117 (2017)

4. Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R.T., Berger, C., Ha, S.M., Rozycki, M., et al.: Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. arXiv preprint arXiv:1811.02629 (2018)
5. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In: Proceedings of the 33rd International Conference on Machine Learning (ICML-16). pp. 1050–1059 (Jun 2015)
6. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
7. Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., Maier-Hein, K.H.: No newnet. In: International MICCAI Brainlesion Workshop. pp. 234–244. Springer (2018)
8. Kamnitsas, K., Bai, W., Ferrante, E., McDonagh, S., Sinclair, M., Pawlowski, N., Rajchl, M., Lee, M., Kainz, B., Rueckert, D., et al.: Ensembles of multiple models and architectures for robust brain tumour segmentation. In: International MICCAI Brainlesion Workshop. pp. 450–462. Springer (2017)
9. Kamnitsas, K., Ledig, C., Newcombe, V.F., 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. Medical image analysis **36**, 61–78 (2017)
10. 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)
11. McKinley, R., Meier, R., Wiest, R.: Ensembles of densely-connected cnns with label-uncertainty for brain tumor segmentation. In: International MICCAI Brainlesion Workshop. pp. 456–465. Springer (2018)
12. Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al.: The multimodal brain tumor image segmentation benchmark (brats). IEEE transactions on medical imaging **34**(10), 1993–2024 (2014)
13. Myronenko, A.: 3d mri brain tumor segmentation using autoencoder regularization. In: International MICCAI Brainlesion Workshop. pp. 311–320. Springer (2018)
14. Puybareau, E., Tochon, G., Chazalon, J., Fabrizio, J.: Segmentation of gliomas and prediction of patient overall survival: A simple and fast procedure. In: International MICCAI Brainlesion Workshop. pp. 199–209. Springer (2018)
15. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
16. Schölkopf, B., Smola, A., Müller, K.R.: Kernel principal component analysis. In: International conference on artificial neural networks. pp. 583–588. Springer (1997)
17. Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P., Feuston, B.P.: Random forest: a classification and regression tool for compound classification and QSAR modeling. Journal of chemical information and computer sciences **43**(6), 1947–1958 (2003)
18. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
19. Wang, G., Li, W., Ourselin, S., Vercauteren, T.: Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In: International MICCAI Brainlesion Workshop. pp. 178–190. Springer (2017)
20. Zhou, C., Chen, S., Ding, C., Tao, D.: Learning contextual and attentive information for brain tumor segmentation. In: International MICCAI Brainlesion Workshop. pp. 497–507. Springer (2018)