



# Multi-decoder Networks with Multi-denoising Inputs for Tumor Segmentation

Minh H. Vu<sup>1</sup>(✉), Tufve Nyholm<sup>1</sup>, and Tommy Löfstedt<sup>2</sup>

<sup>1</sup> Department of Radiation Sciences, Umeå University, Umeå, Sweden  
`minh.vu@umu.se`

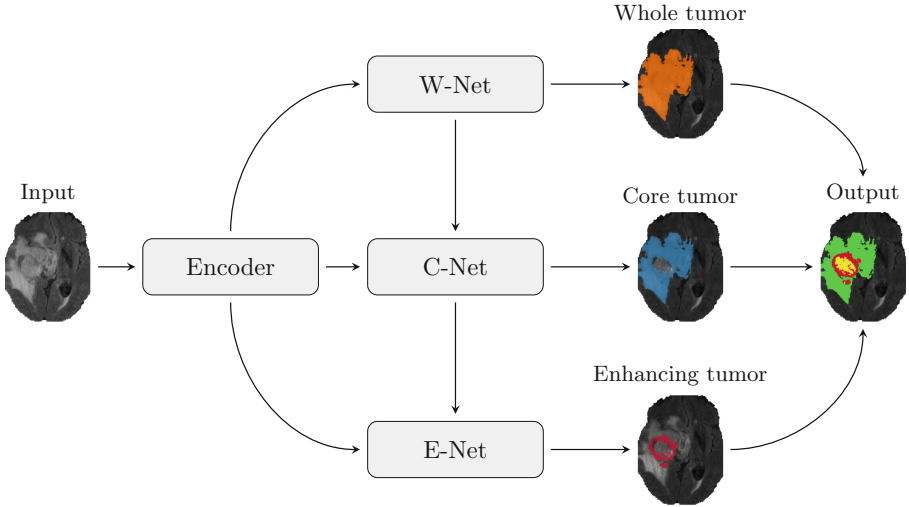
<sup>2</sup> Department of Computing Science, Umeå University, Umeå, Sweden  
`tommy@cs.umu.se`

**Abstract.** Automatic segmentation of brain glioma from multimodal MRI scans plays a key role in clinical trials and practice. Unfortunately, manual segmentation is very challenging, time-consuming, costly, and often inaccurate despite human expertise due to the high variance and high uncertainty in the human annotations. In the present work, we develop an end-to-end deep-learning-based segmentation method using a multi-decoder architecture by jointly learning three separate sub-problems using a partly shared encoder. We also propose to apply smoothing methods to the input images to generate denoised versions as additional inputs to the network. The validation performance indicates an improvement when using the proposed method. The proposed method was ranked 2nd in the task of Quantification of Uncertainty in Segmentation in the Brain Tumors in Multimodal Magnetic Resonance Imaging Challenge 2020.

**Keywords:** Brain tumor segmentation · Uncertainty estimation · Medical imaging · MRI · Ensemble · Deep learning

## 1 Introduction

Glioma is a particular kind of brain tumor that develops from glial cells. It is the most frequently occurring type of brain tumor and the one with the highest mortality rate. Glioma is categorized by the World Health Organization (WHO) into four grades: low-grade glioma (LGG) (class I and II), and high-grade glioma (HGG) (class III and IV), where HGG is being considered a dangerous and life-threatening tumor. Specifically, about 190,000 cases occur annually worldwide [6], and around 90 % [18] of patients die within 24 months of surgical resection. Segmentation of the tumor plays a role both for radiotherapy treatment planning and for diagnostic follow-up of the disease. Manual segmentation is time-consuming, subjective, and associated with uncertainties due to the variation of shape, location, and appearance of the tumors. Hence, decision



**Fig. 1.** Schematic visualization of the MDNet architecture.

support or automating the segmentation may improve the treatment quality as well as enhancing the efficiency when handling this patient group.

Inspired by a need of automatic segmentation of brain tumors in multimodal magnetic resonance imaging (MRI) scans, the Brain Tumors in Multimodal Magnetic Resonance Imaging Challenge 2020 (BraTS 2020) [2–5, 14] is a yearly challenge (associated with the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)) that aims to evaluate state-of-the-art methods for brain tumor segmentation. BraTS 2020 provides the participants with images from four structural MRI modalities: post-contrast T1-weighted (T1c), T2-weighted (T2w), T1-weighted (T1w), and T2 Fluid Attenuated Inversion Recovery (FLAIR) for brain tumor analysis and segmentation. Masks were annotated manually by one to four raters followed by improvements by expert raters. The segmentation performances of the participants were evaluated using the Sørensen-Dice coefficient (DSC), sensitivity, specificity, and the 95<sup>th</sup> percentile of the Hausdorff distance (HD95).

Since the introduction of the U-Net by Ronneberger *et al.* [16], Convolutional Neural Networks (CNNs) incorporating skip connections have become the baseline architecture for medical image segmentation. Various architectures, often building on or extending this baseline, have been proposed to address the brain tumor segmentation problem. In BraTS 2019, Jiang *et al.* [11], who was the first-place winner of the challenge, proposed an end-to-end two-stage cascaded U-Net to segment the substructures of brain tumors from coarse (in the first stage) to fine (in the second stage) prediction. In the same challenge, Zhao *et al.* [21], who won the second place, introduced numerous tricks for 3D MRI brain tumor segmentation including processing methods, model designing methods, and optimizing methods. McKinley *et al.* [13] proposed DeepSCAN, which is a modification

of their previous 3D-to-2D Fully Convolutional Network (FCN), by replacing batch normalization with instance normalization and adding a lightweight local attention mechanism to secure the third place in the BraTS 2019.

The architecture proposed in this work is an extension of the one in [20] from the BraTS 2019 where End-to-end Hierarchical Tumor Segmentation using Cascaded Networks (TuNet) was introduced. Despite achieving a decent performance, the main drawback of TuNet is that it comprises three cascaded networks that make it hard to fit a full volume, with shape  $240 \times 240 \times 155$ , into memory on any recent graphics processing units (GPUs). Because of this, the TuNet adapted a patch-based segmentation approach, leading to long training times. In addition to that, the TuNet might suffer from a lack of global information about the image.

Motivated by the successes of the cascaded networks, presented in *e.g.* [11, 20], the present work proposes a multi-decoder architecture, denoted End-to-end Multi-Decoder Cascaded Network for Tumor Segmentation (MDNet), to separate a complicated problem into simple sub-problems. We also propose to use multiple denoised versions of the original images as inputs to the network. The hypothesis was that this would counteract the salt and pepper noise often seen in MRI scans [1]. To the best of our knowledge, this is the first use of this technique.

The authors hypothesize that the MDNet will reduce overfitting problems by employing a shared encoder between three different decoders, while denoised MRI images will help the network to gain more insight into the multimodal input images with the presence of another two versions of the images: (i) a salt and pepper-free one from the use of a median filter, and (ii) one with reduced high-frequency components by employing a low-pass Gaussian filter.

## 2 Methods

Inspired by the drawbacks of the method proposed in [20], the authors here also propose an end-to-end framework that separates the complicated multi-class tumor segmentation problem into three simpler binary segmentation problems, but with a major change in the design. The MDNet consumes much less memory compared to the TuNet, which means that whole input volumes can be fit into the GPU memory. Hence, the proposed MDNet can take advantage of global details. In addition to that, the design of MDNet results in shorter training times since it uses whole volumes instead of patches, as was the case with the TuNet.

### 2.1 Encoder Network

The encoder network consists of conventional convolution blocks [16], where each block includes a convolution layer with batch normalization and a leaky rectified linear unit (LeakyReLU) activation function. Each convolutional block is then followed by a Squeeze-and-Excitation block (SEB) (see Sect. 2.3). Max-pooling

layers were used for downsampling. All convolutional filters had the size of  $3 \times 3 \times 3$ , and the initial numbers of filters were set to twelve, which in the proposed architecture is equivalent to three denoising methods applied to the four given modalities (see Sect. 2.4). The encoder output has shape  $96 \times 20 \times 24 \times 16$ . The complete architecture of the proposed encoder network is detailed in Table 1.

**Table 1.** The encoder architecture. “Conv3” denotes a  $3 \times 3 \times 3$  convolution, “BN” stands for batch normalization, “LeakyReLU” is the leaky rectified linear unit, and “SEB” denotes the Squeeze-and-Excitation block (see Sect. 2.3).

Name	Layers	Repeat	Output size
Input			$12 \times 160 \times 192 \times 128$
EncBlk-0	Conv3, BN, LeakyReLU, SEB	2	$12 \times 160 \times 192 \times 128$
EncDwn-1	MaxPooling	1	$12 \times 80 \times 96 \times 64$
EncBlk-1	Conv3, BN, LeakyReLU, SEB	2	$24 \times 80 \times 96 \times 64$
EncDwn-2	MaxPooling	1	$24 \times 40 \times 48 \times 32$
EncBlk-2	Conv3, BN, LeakyReLU, SEB	2	$48 \times 40 \times 48 \times 32$
EncDwn-3	MaxPooling	1	$48 \times 20 \times 24 \times 16$
EncBlk-3	Conv3, BN, LeakyReLU, SEB	2	$96 \times 20 \times 24 \times 16$

## 2.2 Multi-decoder Networks

Table 2 illustrates the proposed multi-decoder networks. The decoder networks include three separate paths, where each path is employed to cope with a specific aforementioned tumor region including whole, core, and enhancing, that are denoted by W-Net, C-Net, and E-Net, respectively. Each decoder path comprises skip connections as in U-Net. There was also a SEB after each convolution block and a concatenation operation of the output of the spatial upsampling layers with the feature maps from the encoder at the same level. To enrich the feature maps at the beginning of each level in the C-Net, the feature map at the end of the W-Net on the same level is used. A similar approach is employed in the decoder network of the E-Net and C-Net. By utilizing these, we hypothesize that the W-Net will constrain the C-Net, while the C-Net will constrain the E-Net. Figure 1 illustrates the proposed architecture.

## 2.3 Squeeze-and-Excitation Block

We added a channel-based SEB as proposed by Hu *et al.* [8] after each convolution block or concatenation operation. The idea of SEB is to adapt the weight of each channel in a feature map by adding a content-aware mechanism at almost no computational cost. In recent days, SEB has been widely employed to achieve a huge boost in performance. A conventional SEB includes the following layers in

**Table 2.** Decoder architectures. Here, “Conv3” means a  $3 \times 3 \times 3$  convolution, “Conv1” a  $1 \times 1 \times 1$  convolution, “BN” denotes for batch normalization, “LeakyReLU” means the leaky rectified linear unit, “SEB” denotes the Squeeze-and-Excitation block (see Sect. 2.3), “Up- $\{X\}$ ” represents the 3D linear spatial upsampling of block  $X$ , (+) denotes the concatenation operation. In the name column,  $W-$ ,  $C-$  and  $E-$  correspond to the whole, core, and enhancing tumor regions, respectively.

Name	Layers	Repeat	Output size
W-DecCat-2	Up-EncBlk-3 + EncBlk-2	1	$144 \times 40 \times 48 \times 32$
W-DecSae-2	SEB	1	$144 \times 40 \times 48 \times 32$
W-DecBlk-2	Conv3, BN, LeakyReLU, SEB	2	$48 \times 40 \times 48 \times 32$
W-DecCat-1	Up-DecBlk-2 + EncBlk-1	1	$72 \times 80 \times 96 \times 64$
W-DecSae-1	SEB	1	$72 \times 80 \times 96 \times 64$
W-DecBlk-1	Conv3, BN, LeakyReLU, SEB	2	$24 \times 80 \times 96 \times 64$
W-DecCat-0	Up-DecBlk-1 + EncBlk-0	1	$36 \times 160 \times 192 \times 128$
W-DecSae-0	SEB	1	$36 \times 160 \times 192 \times 128$
W-DecBlk-0	Conv3, BN, LeakyReLU, SEB	2	$12 \times 160 \times 192 \times 128$
W-Output	Conv1, Sigmoid	1	$1 \times 160 \times 192 \times 128$
C-DecCat-2	W-DecBlk-2 + W-DecCat-2	1	$192 \times 40 \times 48 \times 32$
C-DecSae-2	SEB	1	$192 \times 40 \times 48 \times 32$
C-DecBlk-2	Conv3, BN, LeakyReLU, SEB	2	$48 \times 40 \times 48 \times 32$
C-DecCat-1	W-DecBlk-1 + W-DecCat-1	1	$96 \times 80 \times 96 \times 64$
C-DecSae-1	SEB	1	$96 \times 80 \times 96 \times 64$
C-DecBlk-1	Conv3, BN, LeakyReLU, SEB	2	$24 \times 80 \times 96 \times 64$
C-DecCat-0	W-DecBlk-0 + W-DecCat-0	1	$48 \times 160 \times 192 \times 128$
C-DecSae-0	SEB	1	$48 \times 160 \times 192 \times 128$
C-DecBlk-0	Conv3, BN, LeakyReLU, SEB	2	$12 \times 160 \times 192 \times 128$
C-Output	Conv1, Sigmoid	1	$1 \times 160 \times 192 \times 128$
E-DecCat-2	C-DecBlk-2 + W-DecCat-2	1	$240 \times 40 \times 48 \times 32$
E-DecSae-2	SEB	1	$240 \times 40 \times 48 \times 32$
E-DecBlk-2	Conv3, BN, LeakyReLU, SEB	2	$48 \times 40 \times 48 \times 32$
E-DecCat-1	C-DecBlk-1 + W-DecCat-1	1	$96 \times 80 \times 96 \times 64$
E-DecSae-1	SEB	1	$96 \times 80 \times 96 \times 64$
E-DecBlk-1	Conv3, BN, LeakyReLU, SEB	2	$24 \times 80 \times 96 \times 64$
E-DecCat-0	C-DecBlk-0 + W-DecCat-0	1	$48 \times 160 \times 192 \times 128$
E-DecSae-0	SEB	1	$48 \times 160 \times 192 \times 128$
E-DecBlk-0	Conv3, BN, LeakyReLU, SEB	2	$12 \times 160 \times 192 \times 128$
E-Output	Conv1, Sigmoid	1	$1 \times 160 \times 192 \times 128$

sequence: global pooling, fully connected, rectified linear unit (ReLU) activation function, fully connected, and a sigmoid activation function [8].

## 2.4 Denoising the Inputs

The inputs to the network were the MRI modalities, and also each modality after denoising using two different methods: median denoising and Gaussian smoothing. The authors then concatenated the three versions of the images for each modality (the raw image, and the two denoised versions) to obtain a total of twelve images, that were input as different channels. For the median denoising, we used a  $3 \times 3 \times 3$  median filter; the Gaussian smoothing used a  $3 \times 3 \times 3$  Gaussian filter with a standard deviation of 0.5. In this sense, adding a Gaussian smoothed version of the input is similar to adding a down-scaled version of the input image as was proposed for the TuNet [20].

## 2.5 Preprocessing and Augmentation

All input images were normalized to have a mean zero and unit variance. In order to reduce overfitting and increase the diversity of data available for training models, we used on-the-fly data augmentation [9] comprising: (1) randomly rotating the images in the range  $[-1, 1]$  degrees on all three axes, (2) random mirror flipping with a probability of 0.5 on all three axes, (3) elastic transformation with a probability of 0.3, (4) random scaling in the range  $[0.9, 1.1]$  with a probability of 0.3, and (5) random cropping with subsequent resizing with a probability of 0.3.

As in [17], the elastic transformations used a random displacement field,  $\Delta$ , such that

$$R_w = R_o + \alpha\Delta, \quad (1)$$

where  $\alpha$  is the strength of the displacement, while  $R_w$  and  $R_o$  denote the location of a voxel in the warped and original image, respectively. For each axis, a random number was drawn uniformly in  $[-1, 1]$  such that  $\Delta_x \sim \mathcal{U}(-1, 1)$ ,  $\Delta_y \sim \mathcal{U}(-1, 1)$ , and  $\Delta_z \sim \mathcal{U}(-1, 1)$ . The displacement field was finally convolved with a Gaussian kernel having standard deviation  $\sigma$ . In the present case,  $\alpha = 1$  and  $\sigma = 0.25$ .

## 2.6 Post-processing

The most challenging task of BraTS 2020 specifically, and BraTS challenges in general, is to distinguish between LGG and HGG patients by labeling small vessels lying in the tumor core as edema or necrosis. In order to tackle this problem, we used the same strategy as proposed in our previous work [20]. In specific, we labeled all small enhancing tumor region with less than 500 connected voxels as necrosis. The proposed post-processing step aims to handle a few cases where the proposed networks fail to differentiate between the whole and core tumor regions.

## 2.7 Task 3: Quantification of Uncertainty in Segmentation

The organizers of the BraTS challenge introduced the task of ‘‘Quantification of Uncertainty in Segmentation’’ in BraTS 2019 and was held again in BraTS

2020. This task is aimed to measure the uncertainty in the context of glioma region segmentation by rewarding predictions that are (a) confident when correct and (b) uncertain when incorrect. Participants were expected to generate uncertainty maps in the range of  $[0, 100]$ , where 0 represents the most certain and 100 represents the most uncertain. The performance was evaluated based on three metrics: Dice Area Under Curve (DAUC), Ratio of Filtered True Positives (RFTPs), and Ratio of Filtered True Negatives (RFTNs).

Similar to [20], the proposed network, MDNet, predicts the probability of three tumor regions, it thus benefits from this task. Following [20], an uncertainty score,  $u_{i,j,k}^r$ , at voxel  $(i, j, k)$  is defined by

$$u_{i,j,k}^r = \begin{cases} 200(1 - p_{i,j,k}^r), & \text{if } p_{i,j,k}^r \geq 0.5, \\ 200p_{i,j,k}^r, & \text{if } p_{i,j,k}^r < 0.5, \end{cases} \quad (2)$$

where  $u_{i,j,k}^r \in [0, 100]^{|\mathcal{R}|}$  and  $p_{i,j,k}^r \in [0, 1]^{|\mathcal{R}|}$  are the uncertainty score map and probability map, respectively. Here,  $r \in \mathcal{R}$ , where  $\mathcal{R}$  is the set of tumor regions, *i.e.* whole, core, and enhancing region.

## 3 Experiments

### 3.1 Implementation Details and Training

The proposed method was implemented in Keras 2.2.4<sup>1</sup> with TensorFlow 1.12.0<sup>2</sup> as the backend. The experiments were trained on NVIDIA Tesla V100 GPUs from the High Performance Computer Center North (HPC2N) at Umeå University, Sweden. Seven models were trained from scratch for  $N_e = 200$  epochs, with a mini-batch size of one. The training time for a single model was about six days.

### 3.2 Loss

For evaluation of the segmentation performance, we used a combination of the DSC loss and categorical cross-entropy (CE) as the loss function. The DSC is defined as [19, 20]

$$D(u, v) = \frac{2 \cdot |u \cap v|}{|u| + |v|}, \quad (3)$$

where  $u$  and  $v$  are the output segmentation and its corresponding ground truth, respectively. To include the the DSC in the loss function, we employed the soft DSC loss, which is defined as [10, 19, 20]

$$\mathcal{L}_{DSC}(u, v) = \frac{-2 \sum_i u_i v_i}{\sum_i u_i + \sum_i v_i + \epsilon}, \quad (4)$$

<sup>1</sup> <https://keras.io>.

<sup>2</sup> <https://tensorflow.org>.

where for each label  $i$ , the  $u_i$  is the softmax output of the proposed network for label  $i$ ,  $v$  is a one-hot encoding of the ground truth labels (segmentation maps in this case), and  $\epsilon = 1 \cdot 10^{-5}$  is a small constant added to avoid division by zero.

Following [10, 19], for unbalanced data sets with small structures like in the BraTS 2020 data, we added the CE term to our loss function to make the loss surface smoother. The CE is defined as

$$\mathcal{L}_{CE}(u, v) = - \sum_i u_i \cdot \log(v_i). \quad (5)$$

The combination of the DSC loss and CE (denoted a *hybrid loss*) is simply defined as the sum of the two losses, as

$$\mathcal{L}_{\text{hybrid}}(u, v) = \mathcal{L}_{DSC}(u, v) + \mathcal{L}_{CE}(u, v). \quad (6)$$

The final loss function that was used for training contained one hybrid loss for each tumor region, and was thus

$$\mathcal{L}(u, v) = \sum_{r \in \mathcal{R}} \mathcal{L}_{\text{hybrid}}(u_r, v_r), \quad (7)$$

where  $\mathcal{R}$  again is the set of tumor regions (the whole, core, and enhancing regions) and  $\mathcal{L}_{\text{hybrid}}(u_r, v_r)$  is the hybrid loss for a particular tumor region.

The segmentation performance was also evaluated using the HD95, a common metric for evaluating segmentation performances. The Hausdorff distance (HD) is defined as [7]

$$H(u, v) = \max\{d(u, v), d(v, u)\}, \quad (8)$$

where

$$d(u, v) = \max_{u_i \in u} \min_{v_i \in v} \|u_i - v_i\|_2, \quad (9)$$

in which  $\|u_i - v_i\|_2$  is the spatial Euclidean distance between points  $u_i$  and  $v_i$  on the boundaries of output segmentation  $u$  and ground truth  $v$ .

### 3.3 Optimization

The authors used the Adam optimizer [12] with an initial learning rate of  $\alpha_0 = 1 \cdot 10^{-4}$  and momentum parameters of  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ . Following Myronenko *et al.* in [15], the learning rate was decayed as

$$\alpha_e = \alpha_0 \cdot \left(1 - \frac{e}{N_e}\right)^3, \quad (10)$$

where  $e$  and  $N_e = 200$  are epoch counter and total number of epochs, respectively.

The authors also used  $L_2$  regularization with a penalty parameter of  $1 \cdot 10^{-5}$ , which was applied to the kernel weight matrices, for all convolutional layers to counter overfitting. The activation function of the final layer was the logistic sigmoid function.



## 4 Results and Discussion

Table 3 shows the mean DSC and HD95 scores and standard deviations (SDs) computed from the five-folds of cross-validation on 369 cases of the training set. From Table 3 we see that: (i) the U-Net with denoised input improved the DSC and HD95 on all tumor regions, and (ii) the proposed model with denoising boosted the performance in both metrics (DSC and HD95) by a large margin.

**Table 3.** Mean DSC (higher is better) and HD95 (lower is better) and their SEs (in parentheses) computed from the five-folds of cross-validation on the training set (369 cases) for the different models.

Model	DSC			HD95		
	Whole	Core	Enh.	Whole	Core	Enh.
U-Net without denoising	90.66 (0.38)	86.93 (0.71)	76.16 (1.37)	4.91 (0.41)	4.78 (0.42)	3.46 (0.31)
U-Net with denoising	90.98 (0.31)	87.53 (0.68)	76.55 (1.36)	4.49 (0.26)	4.32 (0.29)	3.41 (0.29)
Proposed with denoising	92.75 (0.25)	88.34 (0.70)	78.13 (1.32)	4.32 (0.29)	4.30 (0.31)	3.29 (0.24)

Table 4 shows the mean DSC and HD95 scores on the validation set, computed on the predicted masks by the evaluation server<sup>3</sup> (team name *UmU*). The BraTS 2020 final validation dataset results were 90.55, 82.67 and 77.17 for the average DSC, and 4.99, 8.63 and 27.04 for the average HD95, for whole tumor, tumor core and enhanced tumor core, respectively. These results were slightly lower than the top-ranking teams.

Table 5 provides the mean DAUC, RFTP, and RFTN scores on the validation set obtained after uploading the predicted masks and corresponding uncertainty maps to the evaluation server<sup>4</sup>. As can be seen from Table 5, the RFTN scores were the best amongst the best-ranking participants.

Table 6 and Table 7 show the mean DSC and HD95, and the mean DAUC, RFTP, and RFTN scores on the test set, respectively. In the task of Quantification of Uncertainty in Segmentation, our proposed method was ranked 2nd.

<sup>3</sup> <https://www.cbica.upenn.edu/BraTS20/lboardValidation.html>.

<sup>4</sup> <https://www.cbica.upenn.edu/BraTS20/lboardValidationUncertainty.html>.

**Table 4.** Results of Segmentation Task on BraTS 2020 validation data (125 cases). The results were obtained by computing the mean of predictions of seven models trained from the scratch. “UmU” denotes the name of our team. The metrics were computed by the online evaluation platform. All the predictions were post-processed before submitting to the server. The top rows correspond to the top-ranking teams from the online system retrieved at 11:38:02 EDT on August 3, 2020.

Team	DSC			HD95		
	Whole	Core	Enh.	Whole	Core	Enh.
deepX	91.02	85.00	78.53	4.44	5.90	24.06
Radicals	90.82	84.96	78.69	4.71	8.56	35.01
WassersteinDice	90.58	83.79	78.01	4.74	8.96	27.02
CKM	90.83	83.82	78.59	4.87	5.97	26.57
UmU	90.55	82.67	77.17	4.99	8.63	27.04

**Table 5.** Results of Quantification of Uncertainty Task on BraTS 2020 validation data (125 cases) including mean DAUC (higher is better), RFTPs (lower is better) and RFTNs (lower is better). The results were obtained by computing the mean of predictions of seven models trained from scratch. “UmU” denotes the name of our team and the ensemble of seven models that were trained from the scratch. The metrics were computed by the online evaluation platform. The top rows correspond to the top-ranking teams from the online system retrieved at 11:38:02 EDT on August 3, 2020.

Team	DAUC			RFTPs			RFTNs		
	Whole	Core	Enh.	Whole	Core	Enh.	Whole	Core	Enh.
med_vision	95.24	92.23	83.24	0.28	0.62	0.93	87.74	98.74	98.74
nsu_btr	93.58	90.04	85.14	35.72	48.18	9.59	98.44	98.60	98.64
SCAN	93.46	82.98	80.64	12.40	19.95	21.53	0.87	0.42	0.24
UmU	92.59	83.61	78.83	4.48	10.13	7.95	0.27	0.17	0.08

**Table 6.** Results of Segmentation Task on BraTS 2020 test data (166 cases). The results were obtained by computing the mean of predictions of seven models trained from the scratch. The metrics were computed by the online evaluation platform. All the predictions were post-processed before submitting to the server.

Team	DSC			HD95		
	Whole	Core	Enh.	Whole	Core	Enh.
UmU	88.26	82.49	80.84	6.30	22.27	20.06

**Table 7.** Results of Quantification of Uncertainty Task on BraTS 2020 test data (166 cases) including mean DAUC (higher is better), RFTP<sub>s</sub> (lower is better) and RFTN<sub>s</sub> (lower is better). The results were obtained by computing the mean of predictions of seven models trained from scratch. The metrics were computed by the online evaluation platform.

Team	DAUC			RFTP <sub>s</sub>			RFTN <sub>s</sub>		
	Whole	Core	Enh.	Whole	Core	Enh.	Whole	Core	Enh.
UmU	90.61	85.83	83.03	4.18	5.49	4.45	0.31	1.68	0.07

## 5 Conclusion

In this work, we proposed a multi-decoder network for segmenting tumor sub-structures from multimodal brain MRI images by separating a complex problem into simpler sub-tasks. The proposed network adopted a U-Net-like structure with Squeeze-and-Excitation blocks after each convolution and concatenation operation. We also proposed to stack original images with their denoised versions to enrich the input and demonstrated that the performance was boosted in both DSC and HD95 metrics by a large margin. The results on the test set indicated that: (i) the proposed method performed competitively in the task of Segmentation, with DSC scores of 88.26/82.49/80.84 and HD95 scores of 6.30/22.27/20.06 for the whole tumor, tumor core, and enhancing tumor core, respectively, (ii) the proposed method was top 2 performing ones in the task of Quantification of Uncertainty in Segmentation.

**Acknowledgement.** The computations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at the HPC2N in Umeå, Sweden. We are grateful for the financial support obtained from the Cancer Research Fund in Northern Sweden, Karin and Krister Olsson, Umeå University, The Västerbotten regional county, and Vinnova, the Swedish innovation agency.

## References

1. Ali, H.M.: A new method to remove salt pepper noise in magnetic resonance images. In: 2016 11th International Conference on Computer Engineering Systems (ICCES), pp. 155–160 (2016)
2. Bakas, S., et al.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-GBM collection. The cancer imaging archive (2017) (2017)
3. Bakas, S., et al.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-LGG collection. Cancer Imaging Archive **286** (2017)
4. Bakas, S., et al.: Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Sci. Data **4**, 170117 (2017)
5. Bakas, S., 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](https://arxiv.org/abs/1811.02629) (2018)

6. Castells, X., et al.: Automated brain tumor biopsy prediction using single-labeling CDNA microarrays-based gene expression profiling. *Diagn. Mol. Pathol.* **18**, 206–218 (2009)
7. Hausdorff, F.: Erweiterung einer stetigen Abbildung, pp. 555–568. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-76807-4\\_16](https://doi.org/10.1007/978-3-540-76807-4_16)
8. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
9. Isensee, F., et al.: batchgenerators—a python framework for data augmentation, January 2020
10. Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., Maier-Hein, K.H.: No new-net. In: Crimi, A., Bakas, S., Kuijf, H., Keyvan, F., Reyes, M., van Walsum, T. (eds.) *BrainLes 2018. LNCS*, vol. 11384, pp. 234–244. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-11726-9\\_21](https://doi.org/10.1007/978-3-030-11726-9_21)
11. Jiang, Z., Ding, C., Liu, M., Tao, D.: Two-stage cascaded U-Net: 1st place solution to BraTS challenge 2019 segmentation task. In: Crimi, A., Bakas, S. (eds.) *BrainLes 2019. LNCS*, vol. 11992, pp. 231–241. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-46640-4\\_22](https://doi.org/10.1007/978-3-030-46640-4_22)
12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
13. McKinley, R., Rebsamen, M., Meier, R., Wiest, R.: Triplanar ensemble of 3D-to-2D CNNs with label-uncertainty for brain tumor segmentation. In: Crimi, A., Bakas, S. (eds.) *BrainLes 2019. LNCS*, vol. 11992, pp. 379–387. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-46640-4\\_36](https://doi.org/10.1007/978-3-030-46640-4_36)
14. Menze, B.H., et al.: The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE TRans. Med. Imaging* **34**(10), 1993–2024 (2014)
15. Myronenko, A.: 3D MRI brain tumor segmentation using autoencoder regularization (2018). <http://arxiv.org/abs/1810.11654>
16. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) *MICCAI 2015. LNCS*, vol. 9351, pp. 234–241. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
17. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings, pp. 958–963 (2003)
18. Thurnher, M.: The 2007 WHO classification of tumors of the central nervous system—what has changed? *Am. J. Neuroradiol.* (2012)
19. Vu, M.H., Grimbergen, G., Nyholm, T., Löfstedt, T.: Evaluation of multi-slice inputs to convolutional neural networks for medical image segmentation. arXiv preprint [arXiv:1912.09287](https://arxiv.org/abs/1912.09287) (2019)
20. Vu, M.H., Nyholm, T., Löfstedt, T.: TuNet: end-to-end hierarchical brain tumor segmentation using cascaded networks. In: Crimi, A., Bakas, S. (eds.) *BrainLes 2019. LNCS*, vol. 11992, pp. 174–186. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-46640-4\\_17](https://doi.org/10.1007/978-3-030-46640-4_17)
21. Zhao, Y.-X., Zhang, Y.-M., Liu, C.-L.: Bag of tricks for 3D MRI brain tumor segmentation. In: Crimi, A., Bakas, S. (eds.) *BrainLes 2019. LNCS*, vol. 11992, pp. 210–220. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-46640-4\\_20](https://doi.org/10.1007/978-3-030-46640-4_20)