An Hyperparameter Optimization Study of Brain Tumor Medical Image Segmentation Using U-net

Mahmoud FAHSI [a*], Cheikh MOUILAH[b], Nadir MAHAMMED[c]

[a] Evolutionary Engineering& Distributed Information Systems Laboratory, Univ. Sidi Bel-Abbes, P.B. 89, 22000, Algeria. *email: mahmoud.fahsi@univ-sba.dz*

[b]Biomathematics Laboratory, Univ. Sidi Bel-Abbes, P.B. 89, 22000, Algeria. *email: Cheikh.mouilah@univ-sba.dz*

[c] LabRI Laboratory, Higher School of Computer Science, SidiBel Abbes, Algeria, *email: n.mahammed@esi-sba.dz*

Abstract: TheClinicians take a significant amount of time in interpreting medical images in general and brain MRI in particular to minimize the error rate and identify the exact location of lesions and their types. The emergence of deep learning and the image segmentation performance have called us to apply it for brain tumors identification and thus assisting clinicians to solve this problem. Through the literature review, we can see that the U-Net method is one of the most promising methods already applied to biomedical images in general and on MRI segmentation in particular. But few works have studied the hyperparameters of this technique in order to make a comparison and choose the best one. The existing works focus on the application of the basic U-Net model only without trying to modify these parameters. In this paper, we have made a state of the art of existing U-Net approaches to study the different parametric configurations used and then we have proposed other configurations, based on other deep learning models. A final comparison between the initial approach and the proposed approach gave as good results.

Keywords: Deep Learning, Segmentation, U-Net Model, Magnetic Resonance Imaging MRI, Brain Tumors, Hyper Parameter Tuning.

1. Introduction

The exploration of the human body and its dysfunctions has been the subject of several research over the years, based on the progress of medicine and advances in physics, chemistry, applied mathematics and computer science, science has resulted today in a variety of imaging systems such as standard X-ray, ultrasound, MRI, CT scans, etc. With these, medical imaging makes it possible to observe an organ and see it working[1].

On the other hand, Artificial Intelligence (AI) and deep learning have already proven to be valuable tools to improve early detection, diagnosis and treatment of many serious diseases. If this intelligence were available to every physician today via medical imaging, not only could many lives be saved, but inappropriate treatments and unnecessary surgery could be avoided. Doctors would then have an unprecedented amount of information to automatically support them with the highest level of diagnostic and treatment expertise. In this way, even the smallest and rarest indicators, which a physician might never even encounter over decades of practice, could be identified.

Brain tissue Segmentation from medical data is an important issue for many applications, especially related to the diagnosis and monitoring of neurodegenerative diseases. In the context of brain imaging, Magnetic Resonance Imaging (MRI) has become an increasingly important tool in brain medicine or in cognitive neuroscience research, image segmentation is used in particular in pre-operative planning. To study the evolution of a tumor, it is necessary to know exactly the changes that have occurred on these images.

Among the most widely used Deep Learning approaches for image segmentation, we find the U-Net model [2] which allows both the segmentation and the reconstruction of the brain MRI image while identifying the three regions "whole tumor", "core tumor" and "enhanced tumor" with a very good accuracy. Several implementations of the U-NET approach exist, but most of them use the default U-Net architecture proposed by Ronneberger and al.[2]. Our work objective is to make a state of the art of the different U-Net approaches for biomedical image segmentation and then to test these architectures on brain images in particular. Then, we proposed changes in the settings of the hyperparameters of the best architecture to study their performances.

1.1. Image Segmentation :

Image segmentation is an image processing operation that aims to group pixels together according to predefined criteria [3]. The pixels are thus grouped into regions, which constitute paving or a partition of the image. It can be for example to separate objects from the background. Segmentation is an essential step in image processing. Nowadays, there are many segmentation methods, which can be grouped into four main classes:

- region-based segmentation.
- Edge-based segmentation.
- Thresholding based segmentation.
- Segmentation based on the combination of the first three segmentation methods.

In this context, we can consider U-Net architectures as an efficient method to solve region-based segmentation problems [4].

2. State Of The Art:

There are several classical approaches for MRI image segmentation, each approach has advantages and disadvantages. There are no better approaches or methods for all image segmentation applications, but each approach strongly depends on the applications and images type.

Table 1 : State of the art of deep learning approaches for brain image segmentation

Author	Reprocessing	Method	Dataset
Periera and al.[5]	Normalization, bias field correction	CNNs	BRATS-2013
Havei and al. [6]	Bias correction	CNNs	BRATS-2013
Chang [7]	Normalization	CNNs	BRATS-2016
Kamnitsa and al [8]	Skull stripping, s normalization, registration	CNN+CRF	BRATS-2015
Zhao and al [9]	Bias correction, normalization	FCNN+CRF	BRATS-2013
Mlynarski and al [10]	Skull stripping	2D-3D CNN	BRATS-2017
t.lun and w.hsu[11]	Not mentioned	segNet FCN	BRATS-2015
Dora and al[12]	Normalization	NMF-LSM	BRATS-2015
Song and al.[13]	Normalization	Random Forest	BRATS-2015
Tuan and al. [14]	Brain Slice Category, normalization	U-Net	BRATS-2018

The previous table presents a summary of the most recent and modern deep learning approaches for brain image segmentation. These approaches are based on deep learning algorithms, in particular convolutional neural networks, U-Net and other variants.

3. The U-Net Architecture:

U-net was developed by Ronneberger and al.[2] for biomedical image segmentation. The U-Net model contains two paths. The encoder is used to capture the context in the image and it is formed by just a stack of convolutional layers and max pooling layers. The decoder is used to enable accurate localization using transposed convolutions. It is therefore an end-to-end fully convolutional network (FCN), i.e., it contains only convolutional layers and no dense layers, which allows it to accept images of any size.

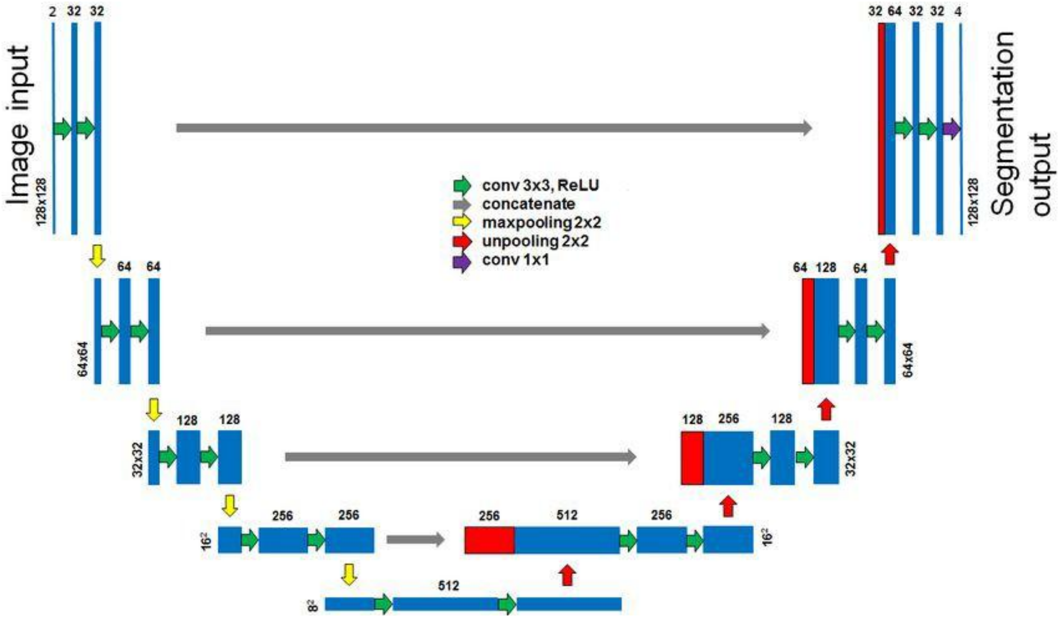

Figure 1: U-Net Original architecture developed by Olaf and al.[2]

We will use the state-of-the-art model hyperparameters to improve the performance of the basic U-Net model [2]and then compare the obtained results with the Tuan et al.[14]approach.

3.1. Hyperparameters Setting:

By studying the approaches cited in Table 1, the best performing recommended hyperparameters were extracted. Our model was based on the following hyperparameters:

3.1.1.ReLU Activation Function

The Rectified Linear Unit (or ReLU) function allows us to perform a filter on our data. It allows positive values (x>0) to pass through the following layers of the neural network defined by:

$$f\ (x) = \ max(0, x) \qquad for\ all\ x\ real \qquad (1)$$

It is frequently used as an activation function in the hidden layers of the neural network for its simplicity of calculation, especially of its derivative[15].

3.1.2.Softmax Activation Function

It’s often used in the output layer of a neural network for multiclassification [16]. And it’s represented mathematically by the following formula:

$$softmax(x_j) = \frac{e(x_j)}{\sum_{k=1}^{n} e(x_k)} \qquad for\ j = 1, .., k \qquad (2)$$

That is, Softmax assigns decimal probabilities to each class in a multi-class problem. The sum of these decimal probabilities must be equal to 1. This additional constraint allows learning to converge faster than it would otherwise.

3.1.3.Loss Function:

the Loss function refers to the cost function that the model will use to minimize errors. The default loss function to use for multiclass classification problems is "categorical crossentropy", also called softmax loss. It is a Softmax activation plus a cross Entropy loss.

3.1.4. Optimization Algorithm (Optimizer)

The model is compiled with the efficient stochastic gradient descent algorithm "Adam"[17], as it is self-tuning and performs well in a wide range of problems. Adam algorithm uses bias correction initialization.

3.1.5. Dropout

Dropout is a method where randomly chosen neurons are skipped during training; they are "dropped".This implies that their contributions to neuron activation are temporarily suppressed during the forward pass and that weight updates are not being applied to the neuron during the reverse pass[18].

Dropout is most easily implemented by randomly selecting the nodes to be dropped with a given probability, e.g., in our case, 0.2 at every weight update round. This technique reduces the overfitting problem.We say that the predictive function generalizes poorly. And that the model suffers from Overfitting[18].

3.1.6. Learning Rate

The optimal learning rate was found to be 0.001 for 2D U-Net [19] (Learning rate used equal to 0.001)

A low learning rate slows down the learning process, but converges smoothly to the descent gradient which implies a minimum of information loss between our model prediction and the real one.

A higher learning rate speeds up the learning, but may not converge.

3.1.7. Batch Size

The batch size is the size of a batch of data, since a computer cannot load all the data in the dataset to use them, we usually divide the dataset in batches. Doing the training for all the batches once is an epoch. Batch size used 35.

3.1.8. Epochs

An epoch is when an entire dataset is passed back and forth through the neural network once[20]. The number of epochs used equal 35.

3.1.9. Callback

We used callbacks[21] which are functions provided by Keras to perform actions at different stages in the training (e.g., at the beginning or end of an epoch, before or after a single beat, etc.). Among the callback functions, we useThe CSVLogger which allows us to store the results obtained in a log file (facilitates the plotting of graphs).

3.1.10. ReduceLR:

Plateau to reduce the learning rate when a metric has stopped improving for 2 continuous epochs, Learning rate is varied between LR that we initialized already between 0.001 and 0.000001.

3.1.11. Other Hyperparameters:

- Patience = 2: is the number of epochs without improvement. The value 0 means that the training ends as soon as the performance measure worsens from one epoch to the next
- monitor = 'val_loss': use loss of validation as a performance measure to end the training.
- Stop the process early
- lower bound on learning rate
- factor: is the value through which the learning rate will be decreased. new_lr = lr-factor

3.2. Used Dataset:

To develop Deep Learning models for brain tumor segmentation, we need to use a dataset annotated with MRI images and validated by radiologists. Therefore, to validate these models, we need to evaluate our experiments on real patient data, in which we use a Brats18 and Brats20 Datasets. Each patient with four MRI sequences: Fluid-Attenuated Inversion Recovery (FLAIR), T1-weighted (T1), T1 with gadolinium-enhancing contrast (T1c) and T2-weighted (T2). In the validation file, there are no ground truths where manual annotations are not publicly available.

- **Brats18 :** the train folder, there are 285 data folders (also with ground truths), according to this article [22] we took this folder for all uses (train/test/validation)
- **Brats20 :** It includes 370 multimodal patient scans for training and 125 cases for validation as the dataset.

3.2.1. Data Pre-Processing

Reprocessing is a necessary step before any deep learning image segmentation. We perform three main segmentation operations:

- Normalization: each individual 3D image is scaled in the range [0-255].
- Brain Slice Category: as long as the data is in 3D and we are using U-Net 2D, we divide it into slices. The shape of the original volume is (240, 240, 155), so there are 155 axial slices. We group the slices that may contain tumors to get better accuracy. The implementation can be done automatically by a learning function or defined manually by omitting some first and last slices. In this paper, we detect the tumor from slices 50-145.
- Object regions: each 2D image can be cropped to effectively implement deep learning. Here, we crop the image size from (240, 240) to (128, 128).

3.2.2. The Dataset Classes

Each segment slice then contains only 4 values (0,1,2,4) where 0 means there is no tumor and the others are different tumor classes. So, we change the values from 4 to 3, therefore it is easier to use.

3.2.3. Data Distribution

- 65% train
- 20% validation
- 15% tests

3.3. Evaluation Metrics:

The segmentation obtained by our method is compared with manual segmentation performed by a medical radiology expert. To evaluate these results, we used the Dice Similarity Coefficient (DSC) which measures the overlap between the segmentation result regions of the proposed method and the ground truth of an image that is used as a reference.

$$DSC(P,T) = \frac{2x|P_1 \cap T_1|}{|P_1|+|T_1|} \qquad (3)$$

- P_1: where represents segmentation result (prediction)
- T_1 : the ground truth generated manually by the radiologists.

4. Experiments Setup

The main idea is to change the hyperparameters settings of the original U-Net approach. Then, train the model on the two datasets Brats18 and Brats20 and then compare the performance of the obtained model with the results of the original approach [2] and the approach of Tuan et al. [14]. We tested several settings and the most important ones are in the following table:

Table 2: Hyper parameters of the different U-Net approaches

Method	Original U-Net [2]	Model U-Net+5 epochs	Model U-Net+ Level Supp	Model U-Net+Layer Supp	Tuan et al.[14]
Network Size	23 layers conv 3x3 4 layers de pooling 2x2 (UpSampling) 4 layers de unpooling (DownSampling)	23 layers conv 3x3 4 layers de pooling 2x2 (UpSampling) 4 layers de unpooling (DownSampling)	**18 layers conv 3x3** **3 layers de pooling 2x2 (UpSampling)** **3 layers de unpooling (DownSampling)**	**11 layers conv 3x3** **3 layers de pooling 2x2 (UpSampling)** **3 layers de unpooling (DownSampling)**	**4 layers conv 3x3** **4 layers conv 5x5** **4 layers de pooling 2x2 (UpSampling)** **4 layers de unpooling (DownSampling)**
Dropout	0.2	0.2	0.2	0.2	0.2
Nb epochs	50	55	35	35	50
Batch size	32	32	32	32	32
Optimizer	Adam	Adam	Adam	Adam	Adam
LR	0,0001	0,0001	0,0001	0,0001	0,0001
Loss	Categorical crossentropy	Categorical crossentropy	Categorical crossentropy	Categorical crossentropy	Categorical crossentropy

We tested several settings for the Optimizer, but "Adam" gave the best result. Same for the Batch size, Learning Rate and Loss function.After testing the model, we obtained the results in the following table and graphs:

Table 3: Brats18 and Brats20 results

	Brats18			**Brats20**		
Method	**Whole tumor**	**coretumor**	**enhancing core**	**Whole tumor**	**coretumor**	**enhancing core**
Original U-Net [2]	0.3847	0.6095	0.4593	0.4854	0.7301	0.7417
Model U-Net+5 epochs	0.1134	0.1817	0.0608	0.5102	0.7492	0.6052
Model U-Net+ Level Supp	0.4206	**0.6695**	**0.5878**	0.5220	0.7411	0.7285
Model U-Net+Layer Supp	0.4287	0.6568	0.5579	0.5697	**0.7613**	**0.7894**
Tuan et al. [14]	**0.77338**	0.51291	0.47623		does not apply	

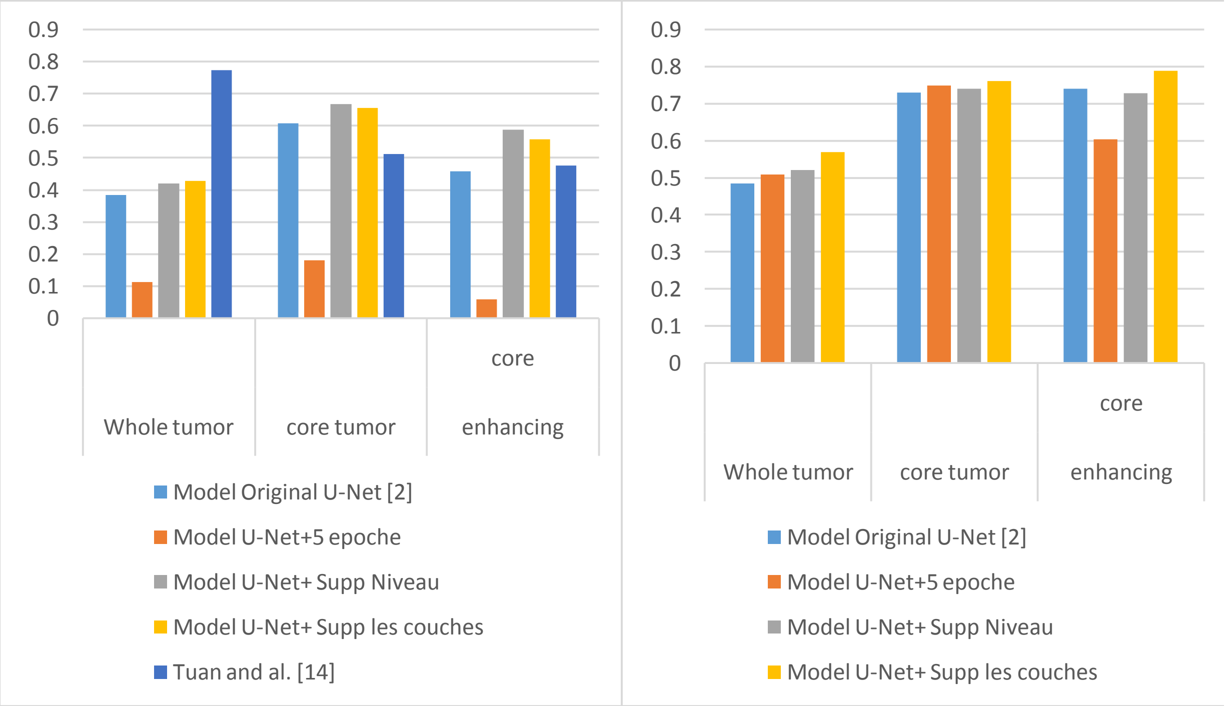

Figure 2:histogram of obtained results (left Brats18, right Brats20)

Through the obtained results on the two datasets (brats18 and brats20) and the different performed experiments, we observe that the "U-Net + layer deletion" model gives the best results with the brats20 dataset especially for the core tumor and enhancing tumor parts, while the "U-Net + level deletion" configuration gives better results in the core tumor and enhancing tumor segmentation. Knowing that the Tuan et al [14] approach is still better in Whole Tumor part segmentation.

5. Conclusion and Perspectives

In the presented work, we introduced a modified U-Net architecture. We have varied several preprocessing methods cited in the state of the art, based on hyperparameters, and our model has been trained on two different datasets. The results are evaluated with the Dice Coefficient measures which allows to compare them with the results quoted in the state of the art.

Work accomplished; we were able to achieve our objectives that we set at the beginning. We have introduced a rather efficient method to segment from MRI the brain tumors that affect our daily life and this with 2 different datasets Brats18 and Brats20. The experiments show the power of neural networks.

We believe that automatic segmentation of brain tumors, with little or no human intervention, can achieve very high accuracy in a very short time compared to manual segmentation.

As perspectives, we think to do more experiments by changing more parameters: number of neurons of the conventional layer, the drop-out rate, etc. maybe also modify the architecture by adding or removing layer(s) and implement U-Net with 3D conventional layers and try other preprocessing methods.

6. Bibliography

[1] M. Viallon *et al.*, "State-of-the-art MRI techniques in neuroradiology: principles, pitfalls, and clinical applications," *Neuroradiology,* vol. 57, no. 5, pp. 441-67, May 2015, doi: 10.1007/s00234-015-1500-1.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015: Springer, pp. 234-241.

[3] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence,* 2021.

[4] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," *Neurocomputing,* 2022.

[5] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Deep Convolutional Neural Networks for the Segmentation of Gliomas in Multi-sequence MRI," Cham, 2016: Springer International Publishing, in Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries, pp. 131-143.

[6] M. Havaei *et al.*, "Brain tumor segmentation with deep neural networks," *Medical image analysis,* vol. 35, pp. 18-31, 2017.

[7] P. D. Chang, "Fully convolutional neural networks with hyperlocal features for brain tumor segmentation," in *Proceedings MICCAI-BRATS Workshop*, 2016, pp. 4-9.

[8] K. Kamnitsas *et al.*, "DeepMedic for brain tumor segmentation," in *International workshop on Brainlesion: Glioma, multiple sclerosis, stroke and traumatic brain injuries*, 2016: Springer, pp. 138-149.

[9] X. Zhao, Y. Wu, G. Song, Z. Li, Y. Zhang, and Y. Fan, "A deep learning model integrating FCNNs and CRFs for brain tumor segmentation," *Medical image analysis,* vol. 43, pp. 98-111, 2018.

[10] P. Mlynarski, H. Delingette, A. Criminisi, and N. Ayache, "3D convolutional neural networks for tumor segmentation using long-range 2D context," *Computerized Medical Imaging and Graphics,* vol. 73, pp. 60-72, 2019.

[11] T. Lun and W. Hsu, "Brain tumor segmentation using deep convolutional neural network," *Proceedings of BRATS-MICCAI,* 2016.

[12] D. Dera, N. Bouaynaya, and H. Fathallah-Shaykh, "Assessing the non-negative matrix factorization level set segmentation on the BraTS benchmark," in *Proc. MICCAI-BraTS Workshop*, 2016, pp. 10-13.

[13] B. Song, C.-R. Chou, X. Chen, A. Huang, and M.-C. Liu, "Anatomy-guided brain tumor segmentation and classification," in *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, 2016: Springer, pp. 162-170.

[14] T. A. Tuan, "Brain tumor segmentation using bit-plane and unet," in *International MICCAI Brainlesion Workshop*, 2018: Springer, pp. 466-475.

[15] A. K. Dubey and V. Jain, "Comparative study of convolution neural network's relu and leaky-relu activation functions," in *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*: Springer, 2019, pp. 873-880.

[16] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] J. Brownlee, *Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras*. Machine Learning Mastery, 2016.

[19] T. Nemoto *et al.*, "Efficacy evaluation of 2D, 3D U-Net semantic segmentation and atlas-based segmentation of normal lungs excluding the trachea and main bronchi," *Journal of radiation research*, vol. 61, no. 2, pp. 257-264, 2020.

[20] J. H. Cho and H. Lee, "Optimization of Machine Learning in Various Situations Using ICT-Based TVOC Sensors," *Micromachines*, vol. 11, no. 12, p. 1092, 2020.

[21] https://keras.io/api/callbacks/ visited on december 2022

[22] C. A. Silva, A. Pinto, S. Pereira, and A. Lopes, "Multi-stage deep layer aggregation for brain tumor segmentation," in *International MICCAI Brainlesion Workshop*, 2020: Springer, pp. 179-188.