Multi Featured Survival Model

Onur Fatih Demir
Hacettepe University
21727102
b21727102@cs.hacettepe.edu.tr

Gökberk Şahin
Hacettepe University
21627615
b21627615@cs.hacettepe.edu.tr

Abstract

Identifying brain tumor tissues in magnetic resonance images (MRIs) is crucial part of the diagnosis, treatment and monitoring of the disease. Manually identifying brain tumors require years of medical training and prone to human error which can cost lives. Automating this process will save us from the time cost of training an expert at identifying such tumors and will reduce the human error factor which will improve the mortality rate of the disease. Here we describe a multitask segmentation network for tumor segmentation from 3D MRIs and additional quantitative features.

1. Introduction

Gliomas are the most common type of primary brain tumors that originate from brain cells. These type of tumors are separated into two main categories: low-grade (LGG) and high-grade (HGG). High-grade gliomas are aggresive tumors that grow rapidly in size and cause fatal problems in human brain. Magnetic Resonance Imaging (MRI) is commonly used to analyze, diagnose and monitor these tumors. There are variety of 3D MRI scans available, namely: T1, T1 with contrast agent (T1c), T2 and Fluid Attenuation Inversion Recover (FLAIR). Each of these scans are meant to emphasize different tissue properties and areas of the tumors.

For the survival task we have the age of each patient as well as their extent of resection. Extent of resection has three classes: GTR, STR and NA. Gross total resection (GTR) is defined as the removal of all tumors, as gauged by magnetic resonance imaging. Subtotal resection (STR) is partly removing the tumors. Some patients did not have surgery at all which are denoted as NA. Number of days that patients have survived given in the dataset in terms of days.

Automating the segmentation process of the MRI scans can save physicians time and it can provide highly accurate details that experts may miss. This task is part of MICCAI BraTS 2020 [1]. This challenge aims to evaluate state-of-the-art methods for the segmentation of brain tumors by providing a 3D MRI dataset with ground truth tumor segmentation labels annotated by physicians.

Predicting survival of a patient using only statistical features is a difficult task. There are many variables that are vital to survival of a patient, handpicked features to predict survival of a patient is not reliable in most cases. Nie et al. [8] proposed a 3D multimodal network to predict survival time of a patient. Instead of relying only on 3D CNN model to extract features, we used important statistiscal features such as age of a patient, whether they are performed a resection etc. to predict survival time of a patient.

In this work we will review and modify a model called 3D U-NET [2]. We plan to modify the network in a way that it is able to solve both segmentation and regression tasks simultaneously. Regression task is predicting how long a patient will survive, given their MRI scans. We propose a pipeline which segments given 4 different modalities T1, T1ce, T2 and Flair, model simultaneously uses segmentation outputs as visual features and statistical features such as patients age, detected tumor area for every MRI image.

In 2017, Kamnitsas et al. [4] proposed an ensemble model for segmentation. In particular this model combined DeepMedic [5] FCN and U-net models and combined their predictions. in 2018 Isensee et al. [6] showed that a regular U-net architecture with only few modifications can achieve competitive performance. The authors used batch size of 2 and crop size of 128x128x128. The authors also used additional dataset which was allowed throughout the competition. At last Myronenko [7] proposed an additional autoencoder branch to regularize model. 3D MRI model with additional autoencoder branch won the 1s place in the BraTS 2018 challenge. Zhou et al. [3] proposed a cross task guided attention model and won the 3rd place in the BraTS 2018 challenge. In 2019 Lachinov et al. [9] proposed a knowledge distillation for brain tumor segmentation, they have used a cascaded U-Net based architecture to segment MRI scans from BraTS 2019 challenge.

Our proposed pipeline model combines feature extraction success of CNN based models to give visual insights

about the segmentation outputs and important statistical features such as age, resection. Pipeline works as follows, first we extract the segmentation outputs using pretrained U-Net model, then extracted segmentation maps are fed to visual feature extractor model, then visual features and statistical features are combined together and fed to fully connected layers and regression task is performed at the last layer.

2. Method

Since the main objective of this project is a task that is part of BraTS challange, firstly we reviewed the past works and models of the previous BraTS challenges to get a feel for the state-of-the-art. Then we focused on subtask which is survival time prediction of a patient.

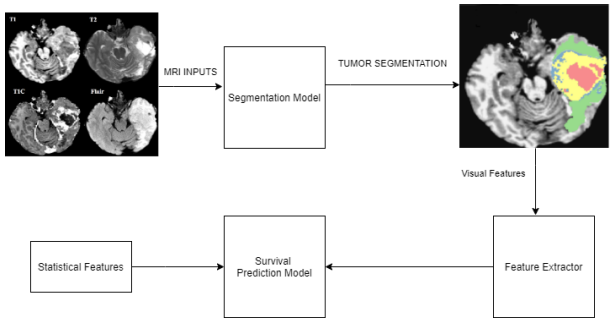

Figure 1: Proposed Pipeline Model

We propose a pipeline model (Figure 1) to combine segmentation task of a MRI scan and survival time prediction of a patient. We believe survival prediction models can benefit from segmentation output masks of a patient. Additionally just like previous works, we have used carefully handpicked features that we think model can benefit from.

2.1. Segmentation Model

We have used the aforementioned U-Net (Figure 2) based knowledge distillation network. This model performed 0.90 dice score on whole tumor, 0.73 dice score on enchanced tumor and 0.83 dice score on tumor core. This model works as follows:

First MRI scans are cropped so that only smallest possible nonzero subregion left as input image. Then each modality runs at parallel and return three different channels, whole-tumor, enchanced tumor and tumor core. During training a combination of Dice Loss and Binary Cross Entropy is used for each of the channels. For all of the input MRI scans corresponding segmentation output mask is saved in a directory and passed to the visual feature extractor which is next model of the pipeline.

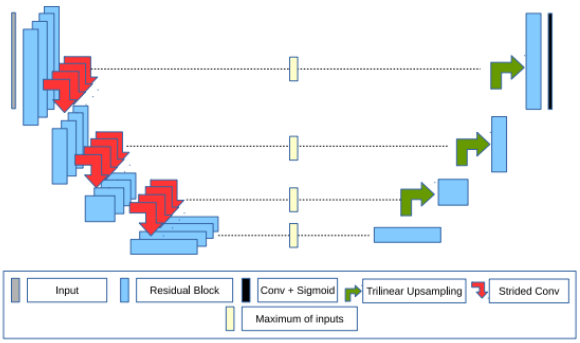

Figure 2: U-Net Based Segmentation Model

2.2. Visual Feature Extractor

We designed a 3D CNN based architecture to give visual features from given segmentation output mask to regression model. This model works as follows:

Each segmentation output mask are in shape of (240, 240, 155), using this output mask as input image, lead to computational expensive architecture, we thought that if we only used segmentation output mask in every W window, we could still preserve the information and have a fast and accurate model. This decreased the average time per epoch by 250%. We used the CNN architecture mention on Table 2 to extract features and Fully connected architecture mention on Table 1 to predict survival time of a patient.

Convolutional Layers				
Layer	in channels	out channels	kernel size	stride
Conv1	24	64	7	2
Max Pool1	64	64	2	2
Conv2	64	64	5	2
Max Pool2	64	64	2	2
Conv3	64	128	3	1
Conv4	128	128	3	1
Max Pool3	128	128	2	2

Table 1: Convolutional Layers

Convolutional Layers		
Layer	in features	out features
FC1	2048	1024
FC2	1024	1024
FC3	1024	1

Table 2: FC Layers

2.3. Additional Quantitative Features

In the previous works, feature extraction is done by using either by 3D CNN or handpicked selected features. But we believe additional information such as a persons age and whether resection is done on a patient can be very beneficial for a regression model.

We can see from the figure 3, a patient's age highly decreases survival time of that patient.

We grouped patients as young if they are younger than 30 years old, as adult if they are younger than 60 years old, as old if they are older than 60 years old. We can see on the figure 4 that almost all the time gross total resection is performed on young patients.

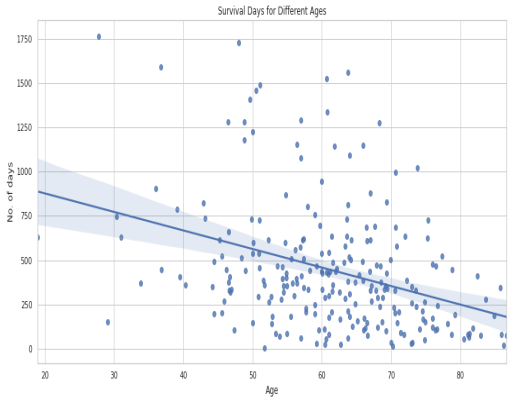

Figure 3: Survival Times vs Age Distribution

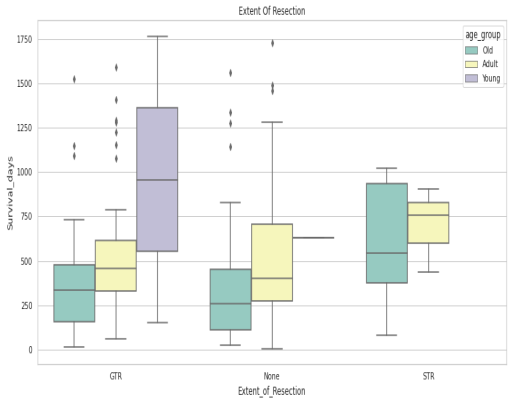

Figure 4: Survival Days by Different Resection Types

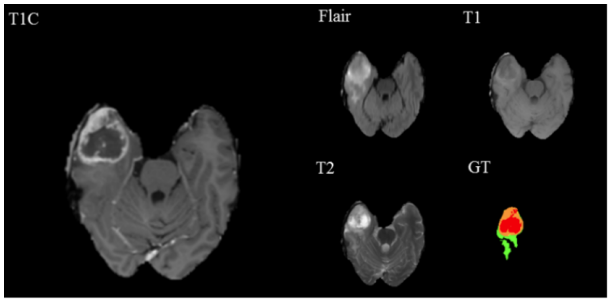

Figure 5: T1, T1C, FLAIR, T2, GT images

3. Experimental Settings

3.1. Dataset

The Brain Tumor Segmentation 2020 (BraTS2020) provided 369 subjects with four different type of MRI scans, T1, T1ce, T2, FLAIR and additional ground truth scan annotated by experienced neuro-radiologists. However for survival task, csv file only contains 236 examples which is a subset of the original dataset. Different type of MRI scans shows different features, the FLAIR and T2 scans are useful for detecting edema, while T1ce can be used to detect inner regions of the tumor and T1 can assist in the seperation of the tumor core from edema.

Glioma has two different type, low grade glioma and high grade glioma. Different grade of the glioma tumor cells look different under microscope "low grade" meaning that the tumor cells look as if they are dividing more slowly under the microscope, or "high-grade," meaning that the cells look more aggressive under the microscope. BraTS 2020, 293 of 369 subjects have high grade glioma, and 76 of the 369 subjects have low grade glioma.

Annotations are enhancing tumor (ET) labeled as 4, the whole tumor labeled as 2, and the necrotic and non enhancing tumor core (TC) labeled as 1. All images are skull stripped and resampled to 1 mm resolution.

Survival data contains patient ID, age, extent of resection and the number of survived days. In the first set of experiments we only used survival days as label and ignored the additional handpicked data. For the second set, we fed additional handpicked data to our network before different fully connected layers.

Average Class Frequency				
	0	1	2	4
Brain	0.96	0.002	0.024	0.006

Table 3: Distribution of the classes

3.2. Model Pipeline

We used a pre-trained version of U-NET architecture to generate the inferences of the dataset. These inferences have dimensions of 155x240x240, meaning that they have 155 channels, these channels represents the timestamp of the MRI scan. Our CNN architecture implements 2D convolutional layers and using 155 channels even in small batch sizes like 8 completely overflows the memory.

Instead we randomly sampled a slice from every W slice which effectively creates a hyperparameter called $window\ size$. We think that this is a good way to sample because we preserve the information in a semantic way by choosing form every W slice instead of randomly selecting from the all possible 155 slices. Also inferences consists of label values like 0, 1, 2 and 4. Since nonexistince of a tumor also captures 0 label, We instead use three channel for three label 1, 2, 4.We converted these values to RGB color space and normalized them. Then we fed the inferences to our convolutional layers to extract the features. Finally we fed these extracted features along with the survival features (age and extent of resection, area of segmentation labels) to fully connected layers for regression. See the table 1 and table 2 for details of the CNN architecture.

3.3. Experiments

We have two major parameters in this work, window size and initial learning rate of the CNN model. We first tried to change the window size for the learning rate of 0.001. We tried for values of 10, 20 and 40. Note that this will require us to change the number of input channels for the first convolutional layer.

Since the best results came from window size 20 and 40, we conducted learning rate experiments with only these window sizes.

We repeated the same set of experiments for the additional survival features age, extend of resection, statistical features such as nonzero area of every MRI image area of tumor core, area of whole tumor, area of enchanced tumor these feature are tested separately as a feature vector and their mean and standard deviation values are also tested. We added these features before different fully connected layers, we observed that if these features are added before the first fully connected layer, the model performs the most.

4. Experimental Results

4.1. Loss

We tried Mean Squared Error loss and L1 loss as a loss function. As we can see in the figure 4, our dataset has a lot of outlier values. L1 loss is fundamentally more robust to outliers. Thus we have used L1 loss for future experiments.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

$$L1 = \sum_{i=1}^{n}|y_{true} - y_{pred}|$$

Loss Function	Train Error	Validation Error
L1	292.76	236.06
MSE	338.60	239.22

Table 4: Different Loss Functions (Learning rate = 0.001, Windowsize = 40)

Loss Function	Train Error	Validation Error
L1	197.94	187.50
MSE	290.83	250.81

Table 5: Different Loss Functions (Learning rate = 0.001, Windowsize = 20)

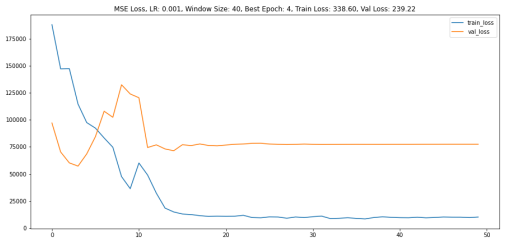

Figure 6: Training and Validation error with window size 40 and MSE loss

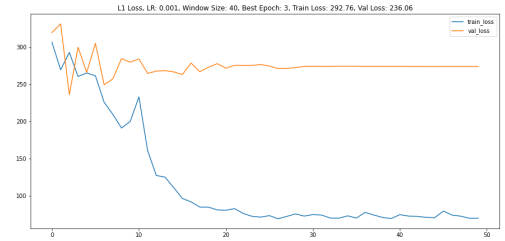

Figure 7: Training and Validation error with window size 40 and L1 loss

4.2. Window Size

We believe that lower window size should impact the model in a positive way. But for window size 10 and 20

this seems to be incorrect. By taking every W image from a MRI scan, we believe in some cases we might miss relevant information and this can be the reason window size 20 is better on this experiment setting.

Window Size	Train Error	Validation Error
10	199.88	210.16
20	197.94	187.50
40	292.76	236.06

Table 6: Different Window Sizes : Error is measured as L1 loss in Days. (Learning Rate = 0.001)

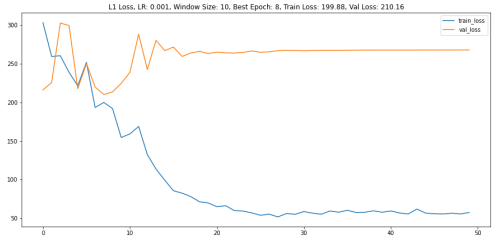

Figure 8: Training and Validation Error for window size of 10

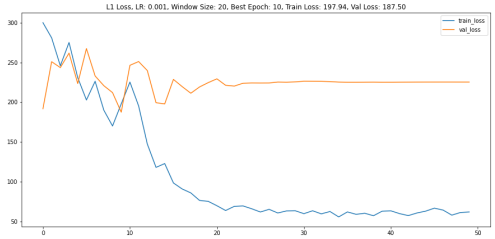

Figure 9: Training and Validation Error for window size of 20

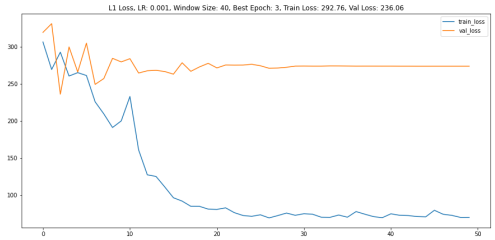

Figure 10: Training and Validation Error for window size of 40

4.3. Learning Rate

We tested different learning rates for window size 20 and 40, in all settings we have used reduce on plateau scheduler with patience value 2 and learning rate is reduced by half at each step. Learning rate 0.001 is the most successful on validation set for window size 20 where as Learning rate of 0.01 is most successful on validation set for window size 40.

Learning Rate	Train Error	Validation Error
0.01	267.08	192.96
0.001	197.94	187.50
0.0001	184.93	238.90

Table 7: Different Learning Rates for window size of 20

Learning Rate	Train Error	Validation Error
0.01	305.29	187.41
0.001	292.76	236.06
0.0001	146.46	204.36

Table 8: Different Learning Rates for window size of 40

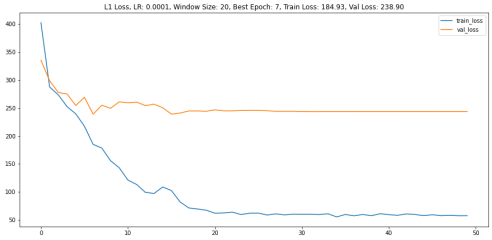

Figure 11: Training and Validation error with Learning rate 0.0001 with window size of 20

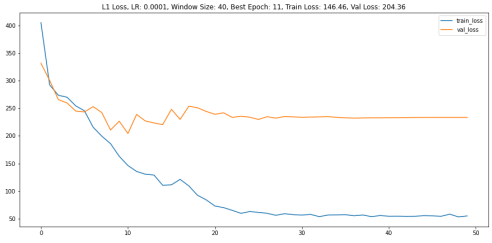

Figure 12: Training and Validation error with learning rate 0.0001 with window size of 40

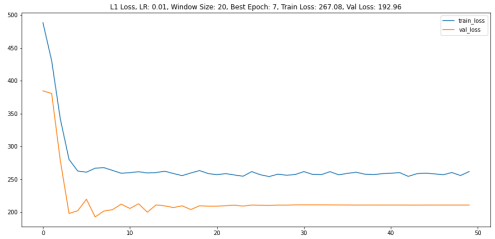

Figure 13: Training and Validation error with Learning rate 0.01 with window size of 20

4.4. Additional Features

We have conducted two more experiments with additional features we mentioned for window sizes of 20 and 40. Learning rate is fixed to 0.001. Adding these features slightly improved the train and validation errors as we can see from the figures 14 and 15.

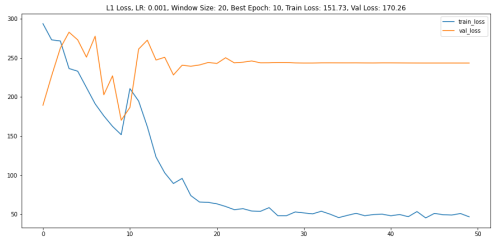

Figure 14: Training and Validation error with additional Features with windows size of 20

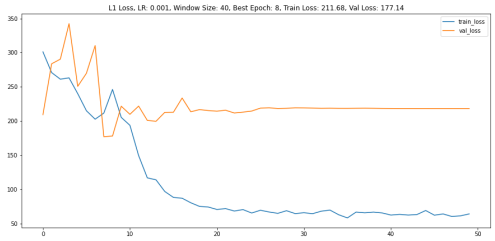

Figure 15: Training and Validation error with additional Features with windows size of 40

5. Discussion and Conclusion

From figure 16 we can see that best model on the test set is model without extra features whereas best model on validation set is model with extra features. Also from the both figure we can see that model unsuccessful predictions

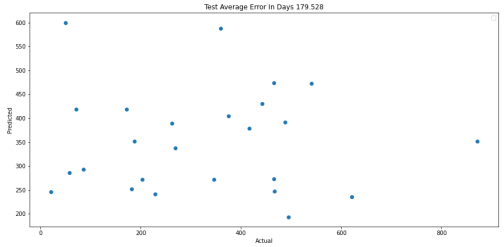

Figure 16: Model Without Extended Features Test Results

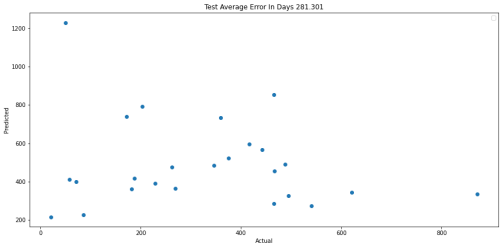

Figure 17: Model With Extended Features Test Results

are mostly outliers of the dataset. Best model predicts with average L1 error of ± 180 days.

Additional features that we have added did not improve the model like we expected.We added age of a patient which in theory should improve the model because survival time of a patient is very dependent on age of that patient. We added resection of a patient which again did not improve. We also added quantitative features from segmentation output masks such as mean and standard deviation of nonzero area, predicted whole tumor area, enchanced tumor area, tumor core area.

Even though model with extended features performed better on train and validation set, it performed poorly on test set when compared with the without features model. We believe additional features increases overfitting but fails on the test set. This could be due to the fact that our test set only contains 25 images and it's distribution might be different than the validation set.

We believe that complexity of regression model, which is a neural network, causes problem to overfit to training set and since the dataset we have is very small compared to other natural image tasks, model overfits the training set and it is not successful as expected as on the validation set.

We have tried to regularize the network and smaller architectures, different learning rates to stop model from overfitting the train set but it did not work. We believe this can be solved when a traditional decision tree based model used instead of a neural network.

We also believe this work can be further improved if a

single model that utilizies encoder of a U-Net for visual embeddings from pretrained network instead of training a second model, then these visual embeddings can be used in a fully connected layer to predict survival time of a patient. An other improvement can be made if one of the previously mentioned better U-Net architectures [7] is used, we chose time complexity over accuracy, most of our model design choices are highly affected by computational complexity. The difficulties of this task are as follows:

- There are many outliers in the dataset which is caused by features that are outside of this dataset.

- Segmentation of a patient is highly important and as so best models are either cascading networks or ensemble methods. But using a ensemble network on this dataset requires very high computation power that we do not have access to we chose time complexity over accuracy which reduces prediction score.

- Brain tumor segmentation dataset (BraTS) is very small compared to other natural image problems. A lot of the variation depended problems can be solved if additional data is used.

References

[1] B. H. Menze et al., "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)," in IEEE Transactions on Medical Imaging, vol. 34, no. 10, pp. 1993-2024, Oct. 2015.

[2] Ronneberger, O., P.Fischer, Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). LNCS, vol. 9351, pp. 234–241. Springer (2015)

[3] Zhou, Chenhong Ding, Changxing Wang, Xinchao Lu, Zhentai Tao, Dacheng. (2019). One-pass Multi-task Networks with Cross-task Guided Attention for Brain Tumor Segmentation.

[4] Kamnitsas, K., W. Bai, E.F., McDonagh, S., Sinclair, M., Pawlowski, N., Rajchl, M., Lee, M., Kainz, B., Rueckert, D., Glocker, B.: Ensembles of multiple models and architectures for robust brain tumour segmentation. In: International Conf. on Medical Image Computing and Computer Assisted Intervention. Multimodal Brain Tumor Segmentation Challenge (MICCAI, 2017). LNCS

[5] Kamnitsas, K., Ledig, C., andJoanna P. Simpson, V.F.N., 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 (2016)

[6] Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., Maier-Hein, K.H.: No newnet. In: International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2018). Multimodal Brain Tumor Segmentation Challenge (BraTS 2018). BrainLes 2018 workshop. LNCS, Springer (2018)

[7] Myronenko, A.: 3d mri brain tumor segmentation using autoencoder regularization. BrainLes 2018, Springer LNCS 11384 (2019) 311–320

[8] Nie D, Zhang H, Adeli E, Liu L, Shen D. 3D Deep Learning for Multi-modal Imaging-Guided Survival Time Prediction of Brain Tumor Patients. Medical Image Computing and Computer-assisted Intervention : MICCAI ... International Conference on Medical Image Computing and Computer-assisted Intervention. 2016

[9] Lachinov D., Shipunova E., Turlapov V. (2020) Knowledge Distillation for Brain Tumor Segmentation. In: Crimi A., Bakas S. (eds) Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. BrainLes 2019. Lecture Notes in Computer Science, vol 11993. Springer, Cham