# Brain Tumor Segmentation with Self-ensembled, Deeply-Supervised 3D U-Net Neural Networks: A BraTS 2020 Challenge Solution

**Chapter** · March 2021

DOI: 10.1007/978-3-030-72084-1_30

CITATIONS
84

READS
241

**7 authors**, including:

Théophraste Henry
Institut de Cancérologie Gustave Roussy
**31** PUBLICATIONS **217** CITATIONS
SEE PROFILE

Alexandre Carre
Institut de Cancérologie Gustave Roussy
**35** PUBLICATIONS **463** CITATIONS
SEE PROFILE

Marvin Lerousseau
CentraleSupélec
**44** PUBLICATIONS **386** CITATIONS
SEE PROFILE

Théo Estienne
CentraleSupélec
**36** PUBLICATIONS **1,424** CITATIONS
SEE PROFILE

# Brain tumor segmentation with self-ensembled, deeply-supervised 3D U-net neural networks: a BraTS 2020 challenge solution.

Théophraste Henry[1] *, Alexandre Carré[1] *, Marvin Lerousseau[1,2], Théo Estienne[1,2], Charlotte Robert[1,3], Nikos Paragios[4], and Eric Deutsch[1,3]

[1] Université Paris-Saclay, Institut Gustave Roussy, Inserm, Radiothérapie Moléculaire et Innovation Thérapeutique, F-94805, Villejuif, France.
[2] Université Paris-Saclay, CentraleSuplec, 91190, Gif-sur-Yvette, France
[3] Gustave Roussy, Département d'oncologie-radiothérapie, F-94805, Villejuif, France
[4] Therapanacea, Paris, France

**Abstract.** Brain tumor segmentation is a critical task for patient's disease management. In order to automate and standardize this task, we trained multiple U-net like neural networks, mainly with deep supervision and stochastic weight averaging, on the Multimodal Brain Tumor Segmentation Challenge (BraTS) 2020 training dataset. Two independent ensembles of models from two different training pipelines were trained, and each produced a brain tumor segmentation map. These two labelmaps per patient were then merged, taking into account the performance of each ensemble for specific tumor subregions. Our performance on the online validation dataset with test time augmentation were as follows: Dice of 0.81, 0.91 and 0.85; Hausdorff (95%) of 20.6, 4,3, 5.7 mm for the enhancing tumor, whole tumor and tumor core, respectively. Similarly, our solution achieved a Dice of 0.79, 0.89 and 0.84, as well as Hausdorff (95%) of 20.4, 6.7 and 19.5mm on the final test dataset, ranking us among the top ten teams. More complicated training schemes and neural network architectures were investigated without significant performance gain at the cost of greatly increased training time. Overall, our approach yielded good and balanced performance for each tumor subregion. Our solution is open sourced at https://github.com/lescientifik/open_brats2020

**Keywords:** Deep Learning · Brain Tumor · Semantic Segmentation

## 1 Introduction

### 1.1 Clinical overview

Gliomas are the most frequent primitive brain tumors in adult patients and exhibit various degrees of aggressiveness and prognosis. Magnetic Resonance Imaging (MRI) is required to fully assess tumor heterogeneity, and the following

* equally contributing authors

sequences are conventionally used: T1 weighted sequence (T1), T1-weighted contrast enhanced sequence using gadolinium contrast agents (T1Gd), T2 weighted sequence (T2), and fluid attenuated inversion recovery (FLAIR) sequence.

Four distinct tumoral subregions can be defined from MRI: the "enhancing tumor" (ET) which corresponds to area of relative hyperintensity in the T1Gd with respect to the T1 sequence; the "non enhancing tumor" (NET) and the "necrotic tumor" (NCR) which are both hypo-intense in T1-Gd when compared to T1; and finally the "peritumoral edema" (ED) which is hyper-intense in FLAIR sequence. These almost homogeneous subregions can be clustered together to compose three "semantically" meaningful tumor subparts: ET is the first cluster, addition of ET, NET and NCR represents the "tumor core" (TC) region, and addition of ED to TC represents the "whole tumor" (WT). Example of each sequence and tumor subvolumes is provided in Figure 1 using 3D Slicer [10].

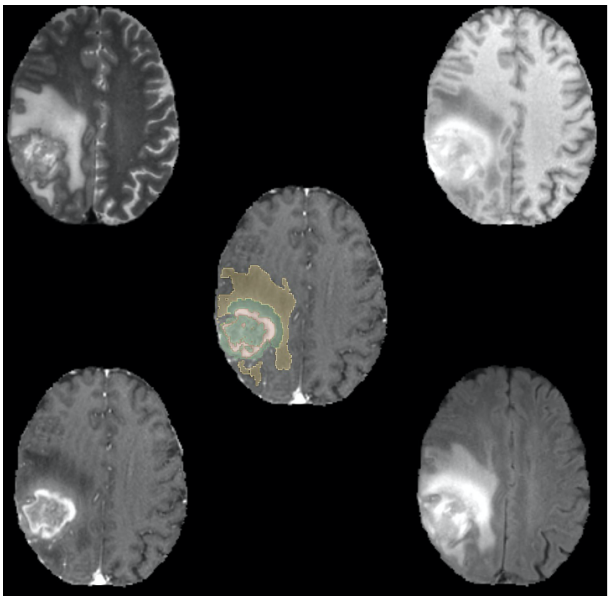

**Fig. 1.** Example of a brain tumor from the BraTS 2020 training dataset. **Red**: enhancing tumor (ET), **Green**: non enhancing tumor/ necrotic tumor (NET/NCR), **Yellow**: peritumoral edema (ED). Upper Left: T2 weighted sequence, Upper Right: T1 weighted sequence, Lower Left: T1-weighted contrast enhanced sequence, Lower Right: FLAIR sequence Middle: T1-weighted contrast enhanced sequence with labelmap overlay

Accurate delineation of each tumor subregion is critical to patient's disease management, especially in a post-surgical context. Indeed, the radiation oncologist is required to segment the tumor, including the surgical resection cavity, the residual enhancing tumor and surrounding edema according to the Radiation

Therapy Oncology Group (RTOG) [22]. Correct segmentation could also unveil prognostic factors through the use of radiomics or deep-learning based approach [9].

### 1.2 Multimodal Brain Tumor Segmentation challenge 2020

The Multimodal Brain Tumor Segmentation Challenge 2020 [19,3,4,6,5] was split in three different tasks: segmentation of the different tumor sub-regions, prediction of patient overall survival (OS) from pre-operative MRI scans, and evaluation of uncertainty measures in segmentation. The Segmentation challenge consisted in accurately delineating the ET, TC and WT part of the tumor. The main evaluation metrics were an overlap measure and a distance metric. The commonly used Dice Similarity Coefficient (DSC) measures the overlap between two sets. In the context of ground truth comparison, it can be defined as follows:

$$DSC = \frac{2TP}{2TP + FP + FN} \tag{1}$$

with TP the true positives (number of correctly classified voxels), FP the false positives and FN the false negatives. It is interesting to note that this metric is insensitive to the extent of the background in the image. The Hausdorff distance [15] is complementary to the Dice metric, as it measures the maximal distance between the margin of the two contours. It greatly penalizes outliers: a prediction could exhibits almost voxel-perfect overlap, but if a single voxel is far away from the reference segmentation, the Hausdorff distance will be high. As such, this metric can seem noisier than the Dice index, but is very handy to evaluate the clinical relevance of a segmentation. As an example, if a tumor segmentation encompasses distant healthy brain tissue, it would require manual correction from the radiation oncologist to prevent disastrous consequences for the patient, even if the overall overlap as measured by the Dice metric is good enough.

## 2 Methods

Two independent training pipelines were designed, with a common neural network architecture based on the 3D U-Net with minor variations (described below). These two different training approaches were kept separate in order to promote network predictions' diversity. The specific details of each pipeline will be described below, and referred to as pipeline A and pipeline B.

### 2.1 Neural network architecture

After neural network architecture exploration, the chosen network used an encoder decoder architecture, heavily inspired by the 3D U-Net architecture from Çiçek et al [35]. The architecture used is displayed in Figure 2.

In the following description, a stage is defined as an arbitrary number of convolutions that does not change the spatial dimensions of the feature maps. All

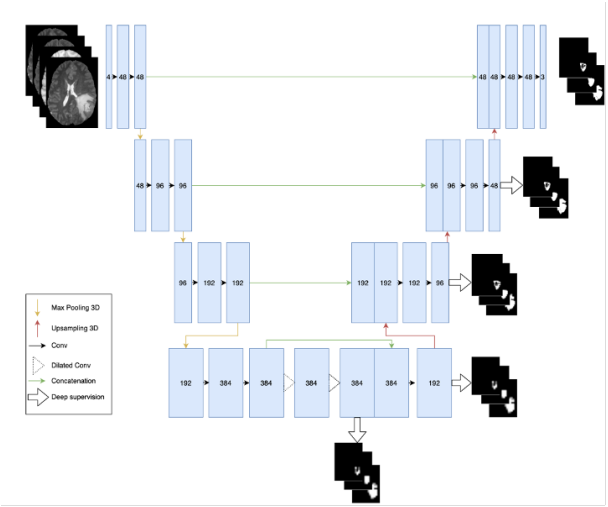


**Fig. 2.** Neural Network Architecture: 3D U-Net [35] with minor modifications

convolutions were followed by a normalization layer and a nonlinear activation (ReLU layer [21]). Group normalization [32] (A) and Instance normalization [29] (B) were used as a replacement for Batch Normalization [16] due to a small batch size during training and good theoretical performance on non-medical datasets.

The encoder had four stages. Each stage consisted of two 3x3x3 convolutions. The first convolution increased the number of filters to the predefined value for the stage (48 for stage 1), while the second one kept the number of output channels unchanged. Between each stage, spatial downsampling was performed by a MaxPool layer with a kernel size of 2x2x2 with stride 2. After each spatial downsampling, the number of filters was doubled. After the last stage, two 3x3x3 dilated convolutions with a dilation rate of 2 were performed, and then concatenated with the last stage output.

The decoder part of the network was almost symmetrical to the encoder. Between each stage, spatial upsampling was performed using a trilinear interpolation. Shortcut connections between encoder and decoder stages that shared the same spatial sizes were performed by concatenation. The decoder stage performing at the lowest spatial resolution was made up of only one 3x3x3 convolution. Last convolutional layer used a 1x1x1 kernel with 3 output channels and a sigmoid activation.

The previous winner of the Brats challenge [1] limited their downsampling steps to 3. We hypothesized that further downsampling of the features maps, given the limited size of the input (128x128x128), would lead to irreversible loss of spatial information. As the last stage of the encoder takes much less GPU memory than the first, the dilation trick [8] was used to perform a pseudo fifth stage at the same spatial resolution as the fourth stage.

3D attention U-Nets were also trained, using the Convolutional Block Attention Module [31] added at the end of each encoder stage.

### 2.2 Loss Function

Inspired by the conciseness of the 2019 winning solution [1], the neural network was trained using only the Dice Loss [20] (A). The loss L is computed batch-wise and channel-wise, without weighting:

$$DSC = 1 - \frac{1}{N} \sum_{n} \frac{S_n * R_n + \varepsilon}{S_n^2 + R_n^2 + \varepsilon} \tag{2}$$

with n the number of output channels, S the output of the neural network after sigmoid activation, R the ground truth label and $\epsilon$ a smoothing factor (set to 1 in our experiment). For diversity, the pipeline B used a slightly different formulation of the Dice Loss, without squaring the terms of the denominator. Similarly, optimization was made directly on the final tumor regions to predict (ET, TC and WT) and not on their components (ET, NET-NCR, ED). The neural network output was a 3-channel volume, each channel representing the probability map for each tumor region.

Deep supervision [30] was performed after the dilated convolutions, and after each stage of the decoder (except the last) as in [23]. Deep supervision was achieved by adding an extra 1x1x1 convolution with sigmoid activation and trilinear upsampling. Like the main output, each of this additional convolution resulted in a 3-channel volume, each channel representing the probability map for each tumor region (ET, TC and WT). The final loss was the unweighted sum of the main output loss, and the four auxiliary losses.

### 2.3 Image pre-processing

Since MRI intensities vary depending on manufacturers, acquisition parameters, and sequences, input images needed to be standardized. Min-max scaling of each MRI sequence was performed separately, after clipping all intensity values to the 1 and 99 percentiles of the non-zero voxels distribution of the volume (A). Pipeline B performed a z-score normalization of the non-zero voxels of each IRM sequence independently.

Images were then cropped to a variable size using the smallest bounding box containing the whole brain, and randomly re-cropped to a fixed patch size of 128x128x128. This allowed to remove most of the useless background that was present in the original volume, and to learn from an almost complete view of each brain tumor.

### 2.4 Data augmentation techniques

To prevent overfitting, on-the-fly data augmentation techniques were applied in both pipelines, according to a predefined probability. The augmentations and their respective probability of application were:

- input channel rescaling: multiplying each voxel by a factor uniformly sampled between 0.9 and 1.1 (A: 80% probability, B: 20%).
- input channel intensity shift: Adding each voxel a constant uniformly sampled between -0.1 and 0.1 (A: not performed, B: 20% probability).
- additive gaussian noise, using a centered normal distribution with a standard deviation of 0.1.
- input channel dropping: all voxel values of one of the input channels were randomly set to zero (A: 16% probability, B: not performed).
- random flip along each spatial axis (A: 80% probability, B: 50%).

### 2.5 Training details

Models were produced by a five-fold cross-validation. The validation set was only used to monitor the network performance during training, and to benchmark its performance at the end of the training procedure.

**Pipeline A:** For each fold, the neural network was trained for 200 epochs with an initial learning rate of 1e-4, progressively reduced by a cosine decay after 100 epochs [12]. A batch size of 1 and the Ranger optimizer [18,34,33] were used. After 200 epochs, we performed a training scheme inspired from the fast stochastic weight averaging procedure [2]. The initial learning rate was restored to half of its initial value (5e-5), and training was done for another 30 epochs with cosine decay. Every 3 epochs, the model weights were saved. This procedure was repeated 5 times for a total of 150 additional epochs. At the end, the saved weights were averaged, effectively creating a new "self-ensembled" model. The Adam optimizer [17] was used without weight decay for the stochastic weight averaging procedure.

**Pipeline B:** The maximum number of training iterations was set to 400. The best model kept was the one with the lowest loss value on the validation set. A batch size of 3 and Adam optimizer with an initial learning of 1e-4 and no weight decay. Cosine annealing scheduler was used.

**Common:** In order to train a bigger neural network, float 16 precision (FP16) was used, which reduced memory consumption, accelerated the training procedure, and may lead to extra performance [12].

The neural network was built and trained using Pytorch v1.6 (which has native FP16 training capability) on Python 3.7. The model could fit on one graphic card (GPU).

### 2.6 Inference

Inference was performed in a two-steps fashion. First, models available from each pipeline were ensembled separately, by simple predictions averaging. Consequently, two labelmaps per case, one for each pipeline, were created. Three

different models per fold (except one fold due to time constraint) were available for pipeline A: a 3D attention U-net version, a U-net version trained on an unfiltered version of the training dataset, and a U-net version trained on a filtered subset of the training dataset. The filtering process was based on previous training runs: cases with high training loss at the end of the training procedure were flagged as potentially wrong and removed from the complete training set, thus creating a "cleaned" version of the training dataset. The top two performing models per fold were chosen for ensembling (A). For Pipeline B, the five cross-validated models (one per fold) were ensembled. Then, the two labelmaps are merged based on the individual performance of each ensemble on the online validation set, as described below.

**First step** For each pipeline, the initial volume was preprocessed like the training data, then cropped to the minimal brain extent, and finally zero-padded to have each of the spatial dimensions divisible by 8. Test time augmentation (TTA) was done using 16 different augmentations for each of the models generated by the cross-validation, for a total of 80 predictions per sample. We used flips, and 90-180-270 rotations only in the axial plane, as rotation in other planes led to worse performance on the local validation set. Final prediction was made by averaging the predictions, using a threshold of 0.5 to binarize the prediction. Labelmap reconstruction was then performed in a straightforward manner: ET prediction was left untouched, the NET-NEC region of the tumor was deduced from a boolean operation between the ET label and the TC label, and similarly for the edema between the TC and the WT label ($NonEnhanching = TC–ET$ ; $edema = WT–TC$).

**Second step** The first step gave two labelmaps per case. Based on the online validation dataset, the mean whole tumour dice metric of the pipeline B's ensemble was consistently higher than that of the pipeline A's ensemble. We hypothesized that models from pipeline B were better for predicting edema. To keep the score intact on ET and TC from models A, ET and NET/NCR predicted labels had to be left untouched. If A predicted background or edema and B predicted edema or background respectively, B predicted labels were kept. The merging procedure is shown in table 1

### 2.7 Ablation Study for Pipeline A

Experiments with and without dataset filtering and attention block were produced for pipeline A. Cross-validated results can be found in Table 2. There was no clear benefit of either strategy, hence we decided to keep the two best available models for each fold for this pipeline.

**Table 1.** Merging procedure of the two labelmaps. 0: background, 1: necrotic and non-enhancing tumor core (NET), 2: peri-tumoral edema (ED), 4:enhancing tumor (ET)

| | | Model A | | | |
|---|---|---|---|---|---|
| | | ***0*** | ***1*** | ***2*** | ***4*** |
| model B | ***0*** | 0 | 1 | 0 | 4 |
| | ***1*** | 0 | 1 | **2** | 4 |
| | ***2*** | **2** | 1 | 2 | 4 |
| | ***4*** | 0 | 1 | 2 | 4 |

**Table 2.** Ablation study: results from cross-validation on the training set.

| Dice: mean(std) | ET | WT | TC |
|---|---|---|---|
| U-Net like | 0.8077 (0.011) | 0.9070 (0.006) | 0.8705 (0.013) |
| + Patients removal | 0.8126 (0.019) | 0.9043 (0.005) | 0.8686 (0.012) |
| + Attention block | 0.8144 (0.022) | 0.9037 (0.008) | 0.8701 (0.018) |

## 3 Results

### 3.1 Online Validation dataset

Table 3 displays the results for the online validation data. Our models produced a Dice metric greater than 0.8. for each tumor region. Our two-pass merging strategy had no impact on the ET and TC segmentation performance of the pipeline A's ensemble, while greatly improving WT segmentation. Single pass strategy already yielded good performance for all three tumor regions. Larger value of Hausdorff distance for ET compared to other tumor subregions is explained by the absence of the ET label for some cases. Consequently, predicting even one voxel of ET would lead to a major penalty for this metric. Example of segmented tumor from the online validation set is displayed in Figure 3. It is hard to visually discriminate best from the average result, based on the mean dice score per patient (average across the three tumor sub-regions). However, our worst generated mask showed obvious error: contrast enhanced arteries were mislabeled as enhancing tumor.

**Table 3.** Performance on the complete BraTs'20 Online Validation Data for the merging strategy, unless otherwise specified.

| Metric (mean) | ET | WT | TC |
|---|---|---|---|
| Dice (Pipeline A alone) | 0.80585 | 0.89518 | 0.85415 |
| Dice (Pipeline B alone) | 0.72738 | 0.91123 | 0.84921 |
| Dice | 0.80585 | 0.91148 | 0.85416 |
| Sensitivity | 0.81488 | 0.91938 | 0.84485 |
| Specificity | 0.99970 | 0.99915 | 0.99963 |
| Hausdorff (95%) | 20.55756 | 4.30103 | 5.69298 |

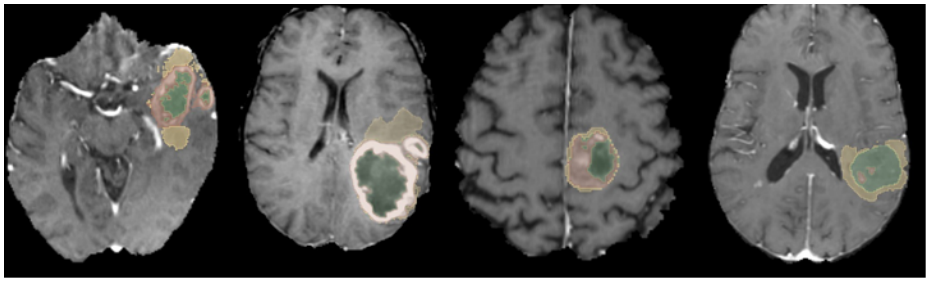

**Fig. 3.** From left to right: ground truth example from the training set, and generated segmentations from our solution for three patients among the online validation set; respectively: best mean dice score (ET:0.95, WT:0.96, TC:0.98), average mean dice score (ET:0.73, WT:0.92, TC:0.93), and worst mean dice score (ET:0.23, WT:0.95, TC:0.13). **Red**: enhancing tumor (ET), **Green**: non enhancing tumor/ necrotic tumor (NET/NCR), **Yellow**: peritumoral edema (ED)

### 3.2 Testing dataset

Our final results on the testing dataset are displayed in table 2. These results ranked us among the top 10 teams for the segmentation challenge. A significant discrepancy between validation and testing datasets for the TC Hausdorff distance was visible, while all other metrics showed small but limited overfit.

**Table 4.** Performance on the BraTs'20 Testing Data.

| Metric (mean) | ET | WT | TC |
|---|---|---|---|
| Dice | 0.78507 | 0.88595 | 0.84273 |
| Sensitivity | 0.81308 | 0.91690 | 0.85934 |
| Specificity | 0.99967 | 0.99905 | 0.99964 |
| Hausdorff (95%) | 20.36071 | 6.66665 | 19.54915 |

## 4 Discussion

Our solution to the BraTS'20 challenge is based on standard approaches carefully crafted together: we used U-net 3D neural networks, trained with on-the-fly data augmentations using the Dice Loss and deep supervision, and inferred using test time augmentation and models predictions ensembling.

Many modern "bells and whistles" were tried: short additive residual connections [11], dense blocks [14], more recent neural networks backbone based on inverted residual bottleneck [13], newer decoder structure like biFPN layer [26], or semi-supervised setting using brain dataset from the Medical Decathlon [24].

None of these refinements led to significant improvement on the local validation set. We hypothesize that this was probably due to GPU memory constraints. Indeed, while these layers improve the model accuracy at a relatively small parameter cost, it increases significantly the size of the activation maps of the model, forcing us to use smaller networks (reduction of the number of output channels per convolutional layer). Reducing the crop size of the patch was not an option as this would have most probably reduced the network performance due to the lack of context. Moreover, all of these additions led to a significant increase of the training time, reducing the searchable space in the limited timeframe of the challenge.

Stochastic weight averaging at the end of the training was the most notable refinement we used. This training scheme was a remnant from the mean teacher semi-supervised training [28]. We did not benchmark its real potential but expect it to produce a more generalizable model, to prevent from overfitting on the training set and to remember the noisy labels. Indeed, it has been shown that a high learning rate could prevent such behavior, and we expect that our training benefits from the multiple learning rate restarts [27].

Notably, while our results were not state of the art for the BraTS 2020 challenge, the segmentation performance of our method is in the usual range of inter-rater agreement for lesion segmentation [7,25] and could already be valuable for clinical use. As an example, Figure 4 zooms in the tumor segmentation of the first two annotations of Figure 3 (respectively manual ground truth annotations and best validation case).

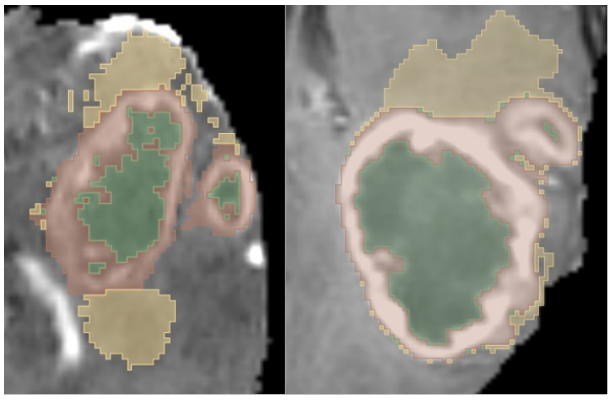

**Fig. 4.** Zoomed version of the first two vignettes of Figure 3 Left: ground truth example from the training set. Right: generated segmentations from our solution for the best mean dice score patient on the validation set. **Red**: enhancing tumor (ET), **Green**: non enhancing tumor/ necrotic tumor (NET/NCR), **Yellow**: peritumoral edema (ED). It is interesting to note that both exhibit the same pattern: central non enhancing tumor core with surrounding enhancing ring and diffuse peritumoral edema.

## 5 Conclusion

The task of brain tumor segmentation, while challenging, can be solved with good accuracy using 3D U-Net like neural network architecture, with a carefully crafted pre-processing, training and inference procedure. We open-sourced our training pipeline at https://github.com/lescientifik/open_brats2020, allowing future researchers to build upon our findings, and improve our segmentation performance.

## References


1. Two-Stage Cascaded U-Net: 1st Place Solution to BraTS Challenge 2019 Segmentation Task | SpringerLink, https://link.springer.com/chapter/10.1007/978-3-030-46640-4_22
2. Athiwaratkun, B., Finzi, M., Izmailov, P., Wilson, A.G.: There are many consistent explanations of unlabeled data: why you should average p. 22 (2019)
3. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.S., et al.: Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features. Scientific Data **4**, 170117 (2017). https://doi.org/10.1038/sdata.2017.117
4. Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., et al.: Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge. arXiv:1811.02629 [cs, stat] (Apr 2019), http://arxiv.org/abs/1811.02629, arXiv: 1811.02629
5. Bakas S, Akbari H, Sotiras A, Bilello M, Rozycki M, Kirby J, et al.: Segmentation Labels and Radiomic Features for the Pre-operative Scans of the TCGA-GBM collection. The Cancer Imaging Archive (2017). https://doi.org/10.7937/K9/TCIA.2017.GJQ7R0EF
6. Bakas S, Akbari H, Sotiras A, Bilello M, Rozycki M, Kirby J, et al.: Segmentation Labels and Radiomic Features for the Pre-operative Scans of the TCGA-LGG collection. The Cancer Imaging Archive (2017). https://doi.org/10.7937/K9/TCIA.2017.KLXWJJ1Q
7. Chassagnon, G., Vakalopoulou, M., Battistella, E., Christodoulidis, S., Hoang-Thi, T.N., Dangeard, S., et al.: AI-Driven quantification, staging and outcome prediction of COVID-19 pneumonia. Medical Image Analysis p. 101860 (Oct 2020). https://doi.org/10.1016/j.media.2020.101860, http://www.sciencedirect.com/science/article/pii/S1361841520302243
8. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. arXiv:1606.00915 [cs] (May 2017), http://arxiv.org/abs/1606.00915, arXiv: 1606.00915
9. Dercle, L., Henry, T., Carré, A., Paragios, N., Deutsch, E., Robert, C.: Reinventing radiation therapy with machine learning and imaging biomarkers (radiomics): State-of-the-art, challenges and perspectives. Methods (Jul 2020). https://doi.org/10.1016/j.ymeth.2020.07.003, http://www.sciencedirect.com/science/article/pii/S1046202319303184
10. Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J.C., Pujol, S., et al.: 3D Slicer as an Image Computing Platform for the Quantitative Imaging Network. Magnetic resonance imaging **30**(9), 1323–1341 (Nov 2012). https://doi.org/10.1016/j.mri.2012.05.001, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3466397/
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs] (Dec 2015), http://arxiv.org/abs/1512.03385, arXiv: 1512.03385
12. He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., Li, M.: Bag of Tricks for Image Classification with Convolutional Neural Networks. arXiv:1812.01187 [cs] (Dec 2018), http://arxiv.org/abs/1812.01187, arXiv: 1812.01187

13. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T.e.a.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861 [cs] (Apr 2017), http://arxiv.org/abs/1704.04861, arXiv: 1704.04861
14. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely Connected Convolutional Networks. arXiv:1608.06993 [cs] (Jan 2018), http://arxiv.org/abs/1608.06993, arXiv: 1608.06993
15. Huttenlocher, D., Klanderman, G., Rucklidge, W.: Comparing images using the Hausdorff distance. IEEE Transactions on Pattern Analysis and Machine Intelligence **15**(9), 850–863 (Sep 1993). https://doi.org/10.1109/34.232073, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence
16. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift (Feb 2015), https://arxiv.org/abs/1502.03167v3
17. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs] (Jan 2017), http://arxiv.org/abs/1412.6980, arXiv: 1412.6980
18. Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Han, J.: On the Variance of the Adaptive Learning Rate and Beyond. arXiv:1908.03265 [cs, stat] (Apr 2020), http://arxiv.org/abs/1908.03265, arXiv: 1908.03265
19. Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., et al.: The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). IEEE transactions on medical imaging **34**(10), 1993–2024 (Oct 2015). https://doi.org/10.1109/TMI.2014.2377694
20. Milletari, F., Navab, N., Ahmadi, S.A.: V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. arXiv:1606.04797 [cs] (Jun 2016), http://arxiv.org/abs/1606.04797, arXiv: 1606.04797
21. Nair, V., Hinton, G.E.: Rectified Linear Units Improve Restricted Boltzmann Machines p. 8
22. Niyazi, M., Brada, M., Chalmers, A.J., Combs, S.E., Erridge, S.C., Fiorentino, A., Grosu, A.L., et al.: Estro-acrop guideline "target delineation of glioblastomas". Radiotherapy and oncology **118**(1), 35–42 (2016)
23. Qin, X., Zhang, Z., Huang, C., Gao, C., Dehghan, M., Jagersand, M.: BASNet: Boundary-Aware Salient Object Detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7471–7481. IEEE, Long Beach, CA, USA (Jun 2019). https://doi.org/10.1109/CVPR.2019.00766, https://ieeexplore.ieee.org/document/8953756/
24. Simpson, A.L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., van Ginneken, B.e.a.: A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv:1902.09063 [cs, eess] (Feb 2019), http://arxiv.org/abs/1902.09063, arXiv: 1902.09063
25. Tacher, V., Lin, M., Chao, M., Gjesteby, L., Bhagat, N., Mahammedi, A., et al.: Semiautomatic Volumetric Tumor Segmentation for Hepatocellular Carcinoma: Comparison between C-arm Cone Beam Computed Tomography and MRI. Academic Radiology **20**(4), 446–452 (Apr 2013). https://doi.org/10.1016/j.acra.2012.11.009, http://www.sciencedirect.com/science/article/pii/S107663321200606X
26. Tan, M., Pang, R., Le, Q.V.: EfficientDet: Scalable and Efficient Object Detection. arXiv:1911.09070 [cs, eess] (Jul 2020), http://arxiv.org/abs/1911.09070, arXiv: 1911.09070
27. Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint Optimization Framework for Learning with Noisy Labels. arXiv:1803.11364 [cs, stat] (Mar 2018), http://arxiv.org/abs/1803.11364, arXiv: 1803.11364

28. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. arXiv:1703.01780 [cs, stat] (Apr 2018), http://arxiv.org/abs/1703.01780, arXiv: 1703.01780
29. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance Normalization: The Missing Ingredient for Fast Stylization. arXiv:1607.08022 [cs] (Nov 2017), http://arxiv.org/abs/1607.08022, arXiv: 1607.08022
30. Wang, L., Lee, C.Y., Tu, Z., Lazebnik, S.: Training Deeper Convolutional Networks with Deep Supervision. arXiv:1505.02496 [cs] (May 2015), http://arxiv.org/abs/1505.02496, arXiv: 1505.02496
31. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: CBAM: Convolutional Block Attention Module. arXiv:1807.06521 [cs] (Jul 2018), http://arxiv.org/abs/1807.06521, arXiv: 1807.06521
32. Wu, Y., He, K.: Group Normalization. arXiv:1803.08494 [cs] (Jun 2018), http://arxiv.org/abs/1803.08494, arXiv: 1803.08494
33. Yong, H., Huang, J., Hua, X., Zhang, L.: Gradient Centralization: A New Optimization Technique for Deep Neural Networks. arXiv:2004.01461 [cs] (Apr 2020), http://arxiv.org/abs/2004.01461, arXiv: 2004.01461 version: 2
34. Zhang, M.R., Lucas, J., Hinton, G., Ba, J.: Lookahead Optimizer: k steps forward, 1 step back. arXiv:1907.08610 [cs, stat] (Dec 2019), http://arxiv.org/abs/1907.08610, arXiv: 1907.08610
35. Çiçek, O., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. arXiv:1606.06650 [cs] (Jun 2016), http://arxiv.org/abs/1606.06650, arXiv: 1606.06650