# Medical Image Segmentation and Applications

## Final Project: Brain tissue segmentation

**Alejandro Cortina Uribe**

**Vladyslav Zalevskyi**

January 9th, 2023

## Introduction and problem definition

Brain tissue segmentation plays an important role in brain image analysis. Partition of the brain into white matter (WM), gray matter (GM), and cerebrospinal fluid (CSF) is one of the basic analysis in neuroscience and consists of splitting of the image into disjoint sets of pixels with shared intensity characteristics belonging to one of the tissue types. Such segmentation is useful for the analysis and detection of diseases like dementia, schizophrenia, traumatic injury, and brain tumors. However, this partition is highly dependent on the image quality, contrast, and modality of the image.

In this final project, we explored and implemented several available approaches for brain segmentation. Our goal was to produce segmentation maps with WM, GM, and CSF labels from the given set of images from the IBSR18 dataset. IBSR18 is one of the standard datasets for tissue quantification and segmentation evaluation. It consists of 18 MRI volumes that are split into training (ten images), validation (five images), and testing (three images) sets by the challenge organizers. This is quite a complicated dataset since the images have a different (often low) signal-to-noise ratio and variable resolution and contrast. Furthermore, the segmentation problem we were tackling in this challenge is unbalanced, since the minority class CSF occupied less than 2% of total GT segmentation volumes (in part due to the way GT were created (ignoring extracerebral CSF) and because of the anatomical properties of the brain).

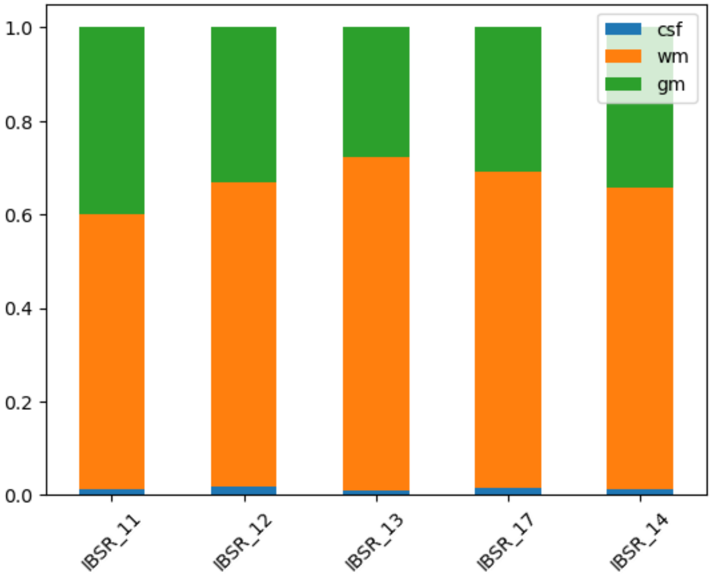


*Figure 1. Percentage of CSF, GM, and WM pixels in the GT segmentation masks for the validation cases.*

For the training and validation images, we had the corresponding ground truth (GT), while for the testing set, they were not available. We used Dice (DSC), Hausdorff distance (HD), and average volumetric difference (AVD) to quantitatively assess the performance of explored models.

## Algorithm analysis and implementation

We decided to implement and test various algorithms to assess their benefits, limitations, and performance on our dataset. As a baseline approach, we implemented a multi-atlas algorithm with majority and weighted voting combination strategies. Then, we used the deep learning model SynthSeg [1] to directly obtain segmentations and evaluate them, without any fine-tuning of the model. Finally, we developed our own deep learning model using a 2D U-Net architecture trained with image patches, testing its performance when a single or two channels (anatomical prior) are used for the input.

To evaluate the performance of our algorithms we used the dice similarity coefficient (DSC) (↑) for each label and the average of them; the 95th percentile of Hausdorff distance (HD) (↓) for each label and the average of them, and the average volume difference (AVD) (↓) for the three labels.

$$DSC = \frac{2 * |A \cap B|}{|A| + |B|} \tag{1}$$

$$d_H(X, Y) = max\{d_{XY}, d_{YX}\} = max\left\{\underset{x \in X}{max}\,\underset{y \in Y}{min}\, d(x, y), \underset{y \in Y}{max}\,\underset{x \in X}{min}\, d(x, y)\right\} \tag{2}$$

$$AVD_{norm} = \frac{\frac{\sum_{i=1}^{3} |V_{seg,i} - V_{GT,i}|}{3}}{\frac{\sum_{i=1}^{3} V_{GT,i}}{3}} \tag{3}$$

### I. Segmentation using multi-atlas label propagation

The premise of this approach was to use our training images as individual atlases, then register them to the target image (validation image) and propagate their segmentation labels. Having all atlases in the target image space, they will be combined with different strategies (equation 4) to obtain a final volume with the predicted labels.

$$S(x) = \underset{c}{argmax} \sum_{i=1}^{P} w_i(x) \cdot f(\pi_i^L(\hat{\tau}_i(x)), c) \qquad f(\pi_i^L(\hat{\tau}_i(x)), c) = \begin{cases} 1 & : \pi_i^L(\hat{\tau}_i(x)) = c \\ 0 & : \pi_i^L(\hat{\tau}_i(x)) \neq c \end{cases} \tag{4}$$

Where $P$ is the number of training images, $c$ is the class label (1, 2, or 3), $w_i$ is a weighting factor, and $f$ is a function of the propagated label with respect to $c$.

### Preprocessing

Before registering the training images to each of the validation set, we preprocessed them in the following sequence:

1. Bias field correction of all images.
2. Resampling moving images to the target image space.
3. Histogram matching with 7 points of each training (moving) image to each target (fixed) image.

We then performed the registration with Elastix's default non-rigid transformation (that consisted of euclidean + affine + b-spline), using advanced mattes mutual information as similarity metric, with a 4-resolutions image pyramid.

After we propagated the labels with the registration transformation map, we computed other similarity metrics between the moving transformed and target images that we later transformed into voting weights:

1. Mattes mutual information.
2. Joint histogram mutual information.
3. Correlation.
4. Mean square error.

Finally, using equation 4 and the computed metrics, we performed simple average (or majority voting) and global weighted average of the ten propagated segmentation images for each of the validation images. By running our experiments, we noticed that these weights were not necessarily ponderating the average, so we raised them to the power $p$. See Appendix A for the results by case.

### Results

We can see that this approach yielded a maximum average DSC across all tissues and patients of around 0.78 (see figure 2 for more results). However CSF segmentation was the worst of the three classes, and this can be seen in both DSC and HD values for all cases of multi-atlas combinations.

Looking deeper into the GT segmentations, we realized that they have many errors, such as labeling voxels as tissue that are out of the brain, anatomically incorrect segmentation of CSF, or noisy voxels scattered in the background that would interfere with both atlas propagation and segmentation output and with segmentation evaluation. See Appendix B for examples.

More importantly, we see that the combination strategies for multi-atlas approaches produce very similar results. Except for using joint histogram mutual information registration error as weighting factor we get very similar results for other weight sources. There is no combination strategy that positively stands out from the others for all cases, even though we have raised the weights to the power of *p* (p=3 set empirically) to increase the difference between the weights.

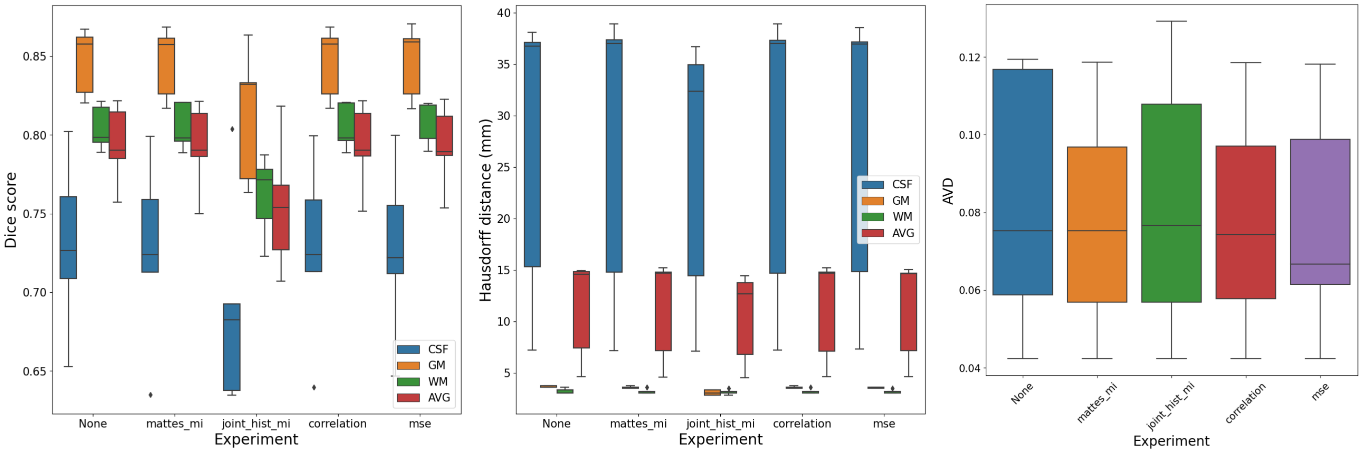


*Figure 2. Multi-atlas results by combination strategy. Weighted by None refers to the simple average (majority voting) of the propagated labels.*

In figure 3 we see a case example of this approach. In the top row, we see an intensity image from the training set (a) and GT segmentation (b), the preprocessing result (c), the registration result (d), and the propagated labels (e). In the bottom row, we see the validation case IBSR_11 intensity image (bias field corrected) (f), GT segmentation (g), and multi-atlas result (h).

## II. SynthSeg inference

Next we moved on to deep learning models to assess their benefits with our dataset. It was of our particular interest to test a state-of-the-art contrast-agnostic model since its main goal is to generalize to different domains, populations, modalities, and data types.

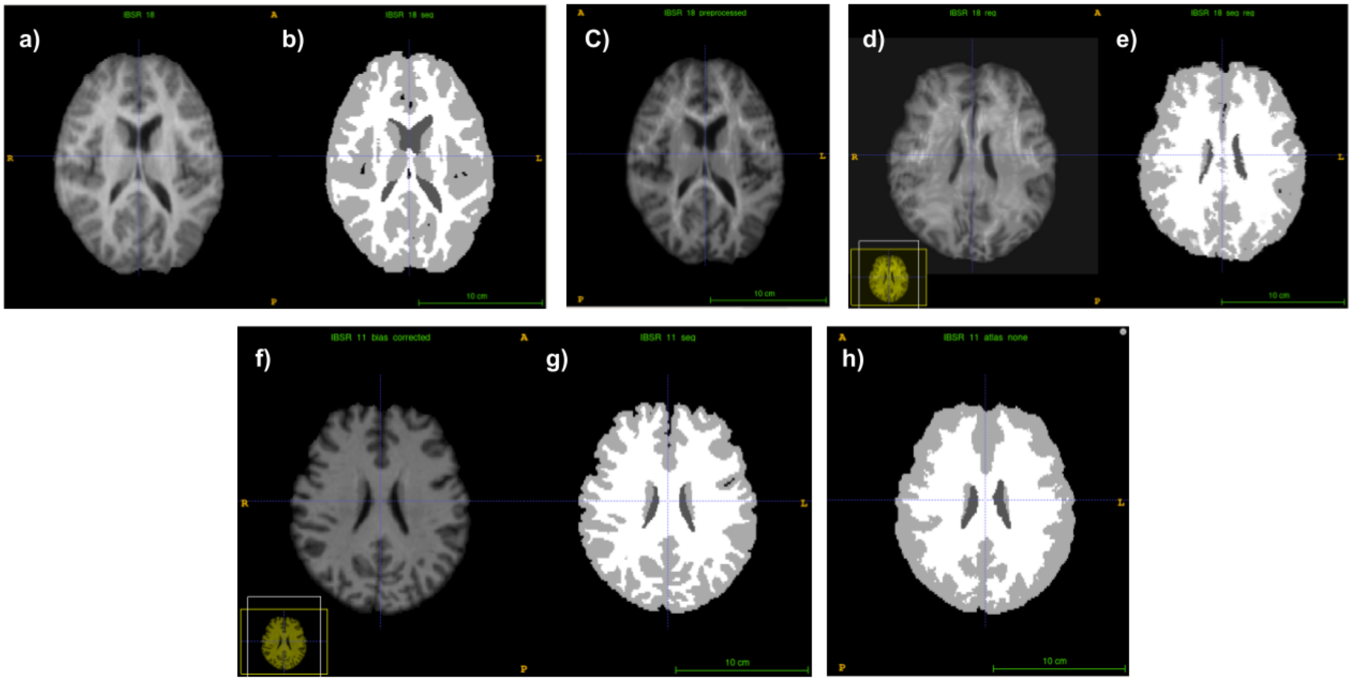


*Figure 3. Multi-atlas label propagation results (h) for validation case 11.*

In [1] they developed a generator that creates randomized synthetic data based on labels, in order to train with it a 3D U-Net. In this way, they successfully solve the famous problem of DL of annotated data scarcity and also have direct control of all variables to generate it (contrast, resolution, morphology, artifacts, noise, etc.) (figure 4). It allowed them to train a segmentation model invariant to contrast and resolution that works out-of-the-box. Considering the challenges of the presented dataset we thought it would be a good idea to validate their model and see how it performs in our scenario without doing any fine-tuning of the model or data. More importantly, since the model was not trained on any real images, its use has no conflicts with our challenge.

Considering the specifications of the SynthSeg model, we were able to use our dataset without any preprocessing. We only needed to perform some postprocessing as the obtained segmented images had isotropic resolution (1x1x1 mm) and 32 labels (including subcortical structures).

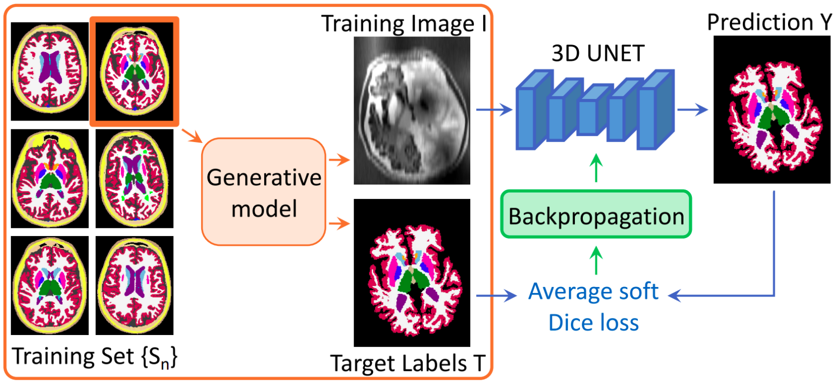


*Figure 4. Overview of a training step of SynthSeg. At each mini-batch, they randomly select a 3D label map from a training set {S n} and sample a pair {I, T} from the generative model. The obtained image is then run through the network, and its prediction Y is used to compute the average soft Dice loss, that is back propagated to update the weights of the network. Figure and caption retrieved from original paper [1].*

### Results

For us to calculate the performance metrics, we post-process the model's output by resampling back to each original validation image dimensions and voxel spacing. Also, we merged the 32 labels into the 3 target labels, for proper calculation. Once again, we noticed that the CSF GT segmentation looks anatomically unrealistic since in the GT segmentation there are barely any CSF pixels around the cortex (figure 5). We confirmed this once again with the first set of SynthSeg results (figure 6) having a poor performance at CSF segmentation.

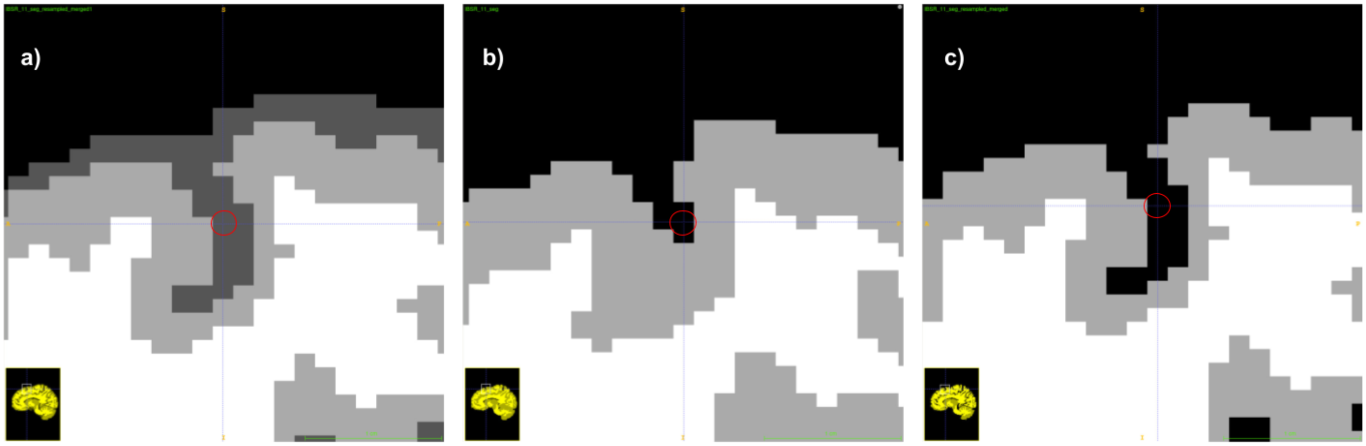


*Figure 5. Synthseg case 11 segmentation result, zoomed in. a) result after merging the 32 labels into 3, b) GT segmentation, c) result after setting to 0 the extra-cerebral CSF. The red circles indicate the same pixel position in both images.*

To eliminate the problem of SynthSeg segmenting extracerebral CSF that is anatomically correct but that did not align with our GT, we decided to map the Synthseg label corresponding to extracerebral CSF to background (value 0), thus reducing the false positives (figure 5 c). In figure 6 (bottom row), we see the improvement in performance.

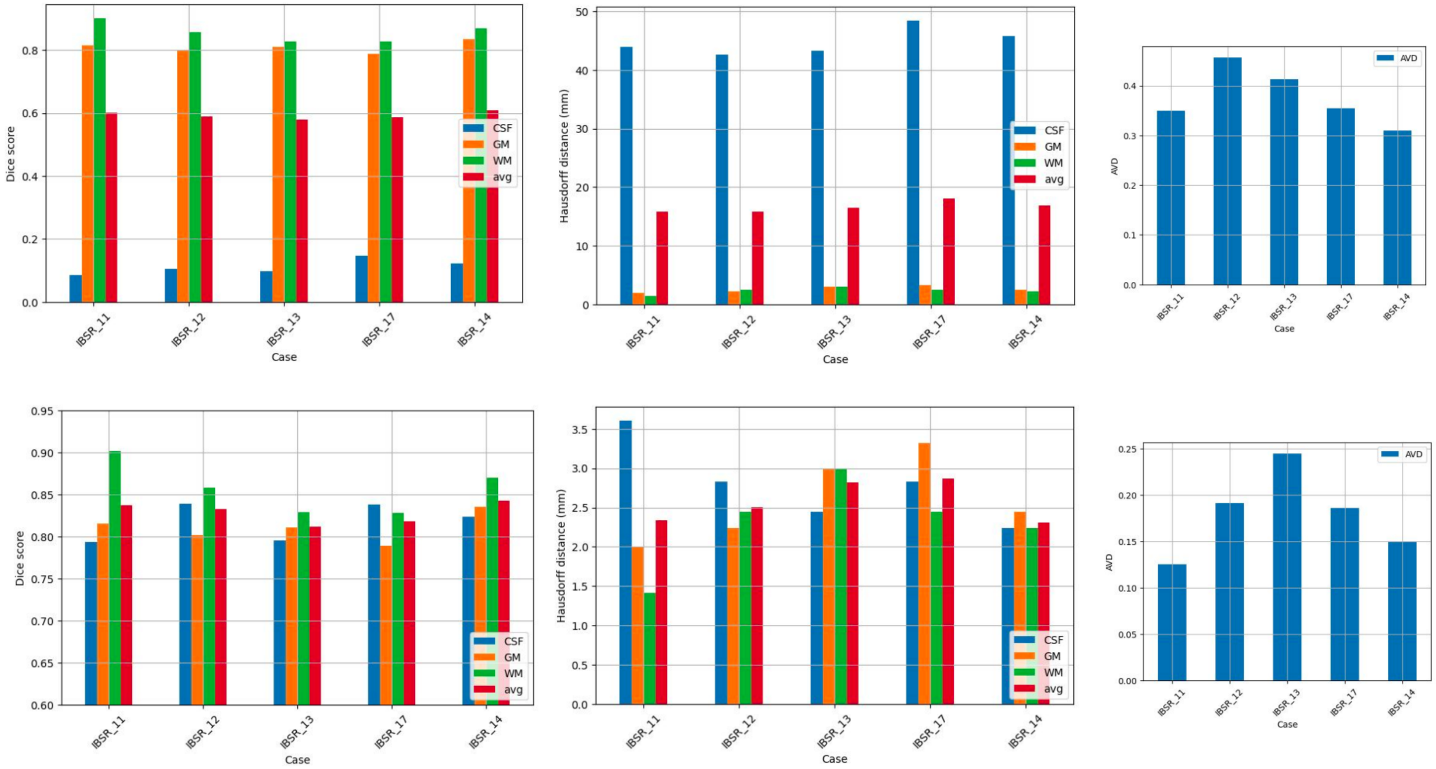


*Figure 6. Synthseg results after direct tissue merging. Top, considering direct mapping. Bottom, considering extracerebral CSF as background.*

From the result plots, we can see an improvement in all performance metrics with respect to the multi-atlas results. For all metrics, the scores are less variable for all images meaning that the model is more robust to our dataset inter-patient variations, as it was expected. The only metric that got worse was AVD, since our multi-altas approach gives a coarser segmentation while SynthSeg's is more detailed between the different tissues, ending up in a volume reduction for all labels.

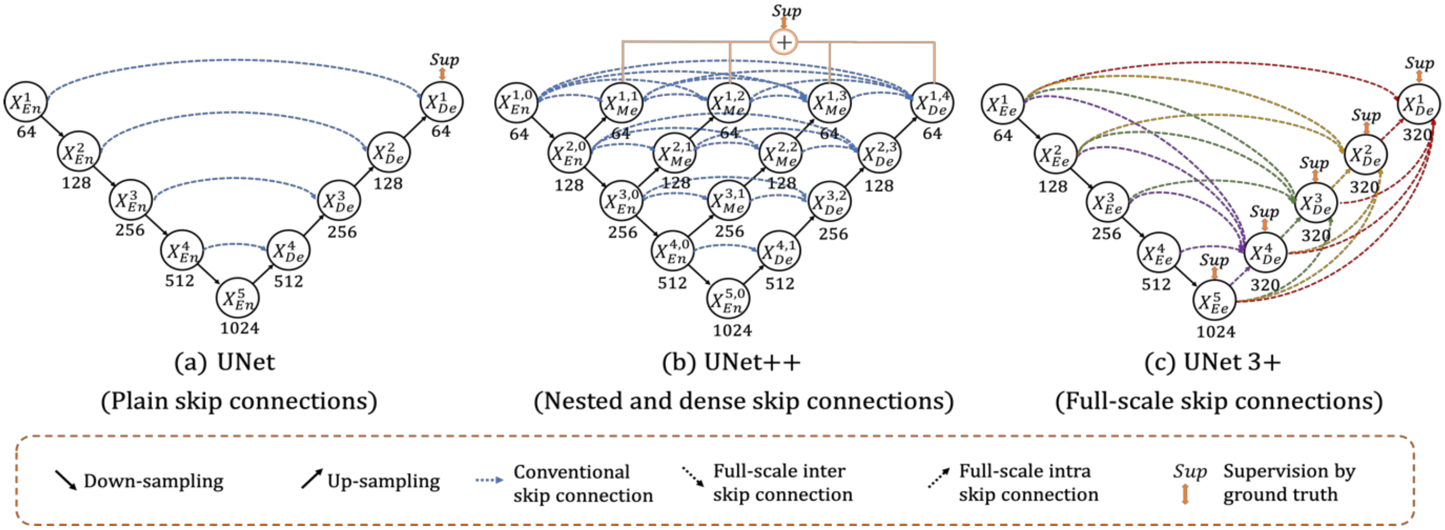
Finally, the most drastic improvement between the SynthSeg remapping changes can be seen in HD, with a reduction of one order of magnitude, and this is due to the fact that having a lot of CSF-labeled pixels around the brain negatively impacts the distance-based metrics, and when removed, this impact is reverted.

Overall, SynthSeg was able to produce good results on the given dataset with an average DSC across 3 tissues of 82.8 for the validation set. We can see that this is better than the best multi-atlas segmentation results, however, we needed to do some post-processing to account for the GT specifications. However, we do not consider this to be a big mistake of the SynthSeg since it did produce anatomically plausible segmentations and the low metric scores can be accounted for by the poor GT quality.

## III. Patch-based 2D U-Net

Finally, we decided to implement and train our own deep learning model. The very popular U-Net has been used extensively to perform brain tissue segmentation, and it has gone through improvements and modifications such as the U-Net++ and U-Net3+ [2]. The latter was proposed to take advantage of full-scale skip connections and deep supervisions (figure 7). It is one of the many variants of the classic U-Net available on the internet with available code in Pytorch that we could update and use.

To simplify our experiments, we decided to use the U-Net 3+ architecture without deep supervisions, which provides fewer parameters than its ancestors and yields a more accurate position-aware and boundary-enhanced segmentation map. We have used [3] implementation for the U-Net 3+ and focused on adjusting the model to our task and dataset and on training it from scratch.



(a) UNet (Plain skip connections) (b) UNet++ (Nested and dense skip connections) (c) UNet 3+ (Full-scale skip connections)

*Figure 7. Comparison of U-Net (a), U-Net++ (b) and U-Net 3+ (c). Figure retrieved from original paper [unet3+].*

To tackle the problem of our small training dataset (only 10 training images), we exploited a patch-based training scheme. We used fixed-size overlapping 2D patches extracted from the images. Among the created training patches we filtered ones that contained a certain amount of non-zero pixels (50%) and we also extracted these patches from slices of all three views of the image (sagittal, coronal and axial). This allowed us to drastically increase the number of training samples.

For the inference time, the same algorithm is used with the test image (except for the part of the filtering of patches with low brain percentage) and then the final segmentation is reconstructed from the predicted patches segmentations.

### Hyperparameters

We developed our experiments by varying different hyperparameters as can be seen below.

- Patch size:
  - 32, with stride 16
  - 64, with stride 32
  - 128, with stride 32
- Loss function:
  - Cross entropy loss
  - Focal loss
  - Dice loss
- Data augmentations:
  - Without
  - With augmentations like HorizontalFlip, VerticalFlip, Rotate, GaussianBlur, Downscale RandomBrightnessContrast (we chose such augmentations as they reproduce patch variability and contrast/resolution differences between images in the dataset)

Another important thing that we have tested is adding additional anatomical prior to the model input to aid the segmentation model. For example, in the SynthSeg+ [1] architecture, the authors have used a cascaded architecture in which they have few segmenting models (based on U-Net) where the latter model takes as input the image to be segmented and the segmentation output of the previous model. It allows the latter model to have some prior information about the segmentation and focus on refining it, instead of learning it from scratch. In the case of SynthSeg+ the earlier model was performing brain segmentation into 4 classes while the latter into 32 classes, so the input of the coarse 4 classes segmentation to the second model was supposed to serve as an anatomical guide to the tissue types present in the brain before the model learned to subdivide them into more regions.

In our case, since our final goal was three-class segmentations, the images were already skull-stripped and we did not have time and resources to train such cascaded architecture, we decided to use SynthSeg segmentations (from the section 2 of this report) as such prior. It would serve as a rough anatomical guideline of the tissues present in the images and thus the model would need to learn the refinement of this segmentation based on the given image. As shown in the section 2, SynthSeg results were anatomically plausible and correct, and since they did not require multiple time consuming registrations to compute and had higher DSC and other metrics, we decided to use them instead of the multi-atlas segmentations for our prior channel.

Therefore, for the experiments below (experiments 2-5) we used 2 channel 2D patch for the model input, in which one channel is a patch of image and the second channel is the corresponding patch of the SynthSeg segmented mask.

## Results

1. Augmentation test.
    a. No augmentation
    b. Augmentation

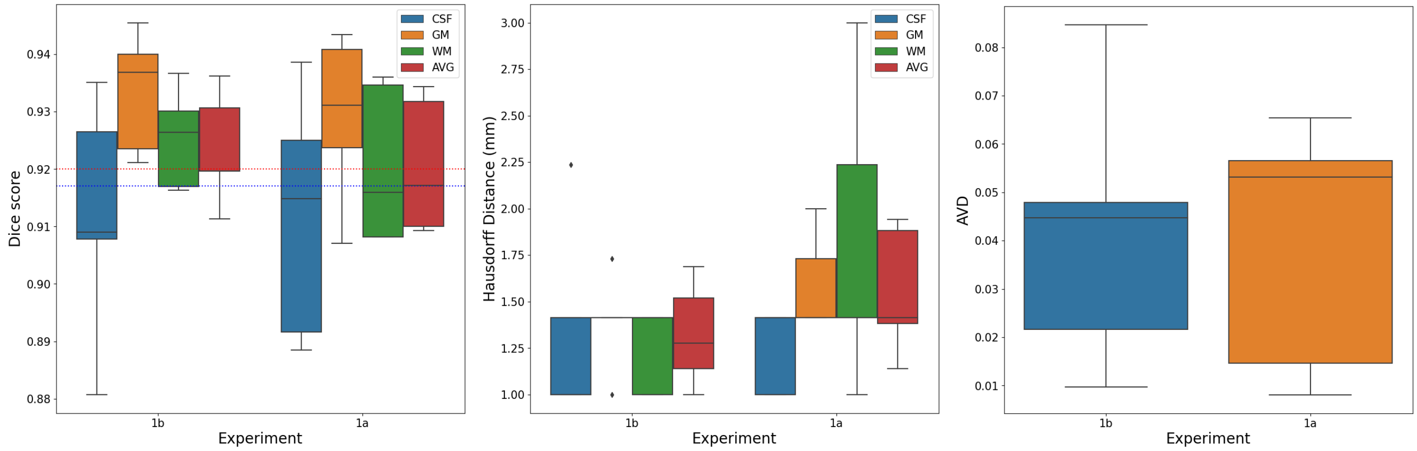


*Figure 8. Boxplots of results for the data augmentation experiments 1a and 1b.*

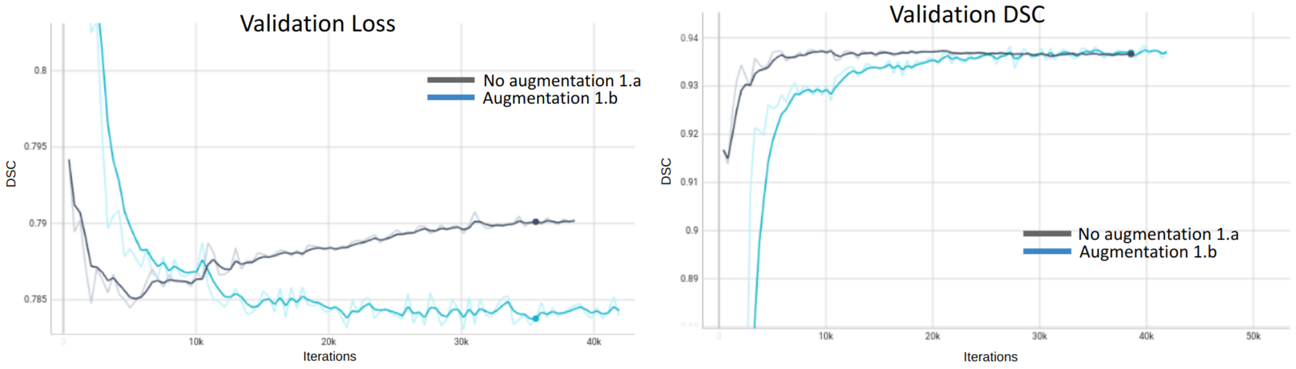


*Figure 9. Training curves for the augmentation experiments. All hyperparameters were equal in both experiments except for the use of augmentation.*

The use of augmentation improved the results on all metrics, as can be seen in figure 8 but also allowed us to tackle the overfitting problem as can be seen in figure 9. The only drawback of using augmentations is that we need more epochs to reach the same performance for the model, due to higher variability in data.

2. SynthSeg segmentation as a prior input
    a. Previous experiment 1b
    b. Previous experiment 1b with the second channel input

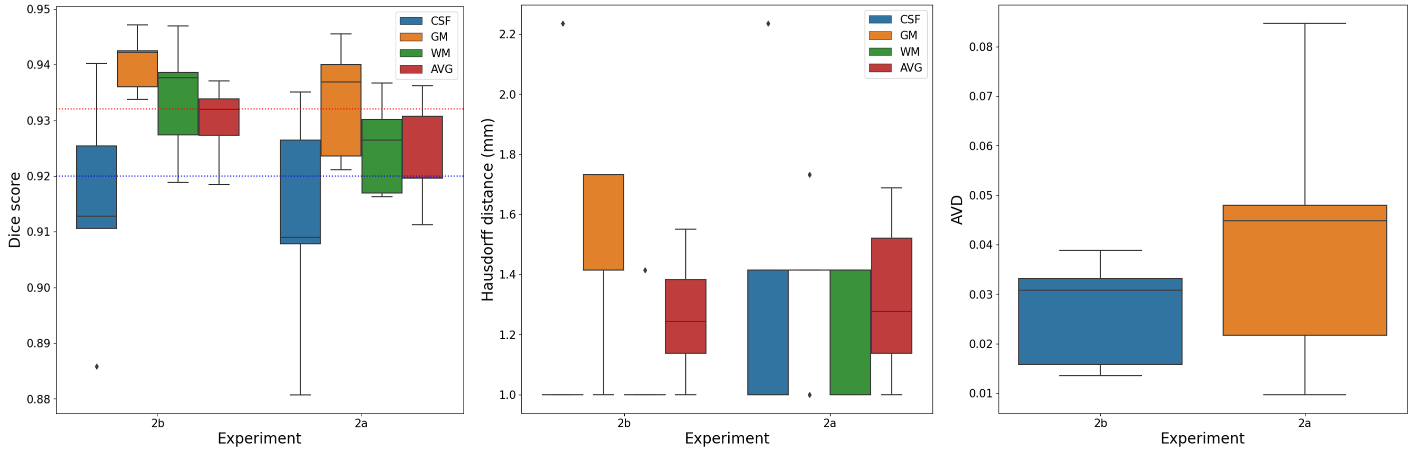


*Figure 10. Boxplots of results for experiments 2a and 2b.*

Having SynthSeg segmentation as a prior for the model further improved the results based on all metrics, which proved our hypothesis that it can be used as an anatomical guide for the model. It shows that this cascaded idea exploited in the original paper can be applied in other cases too. Important to note that for this experiment, SynthSeg's segmentation consisted of all 32 original labels.

3. Loss function test
    a. Previous experiment 2a (that used cross-entropy loss).
    b. Previous experiment 2a, with dice loss.
    c. Previous experiment 2a, with focal loss.

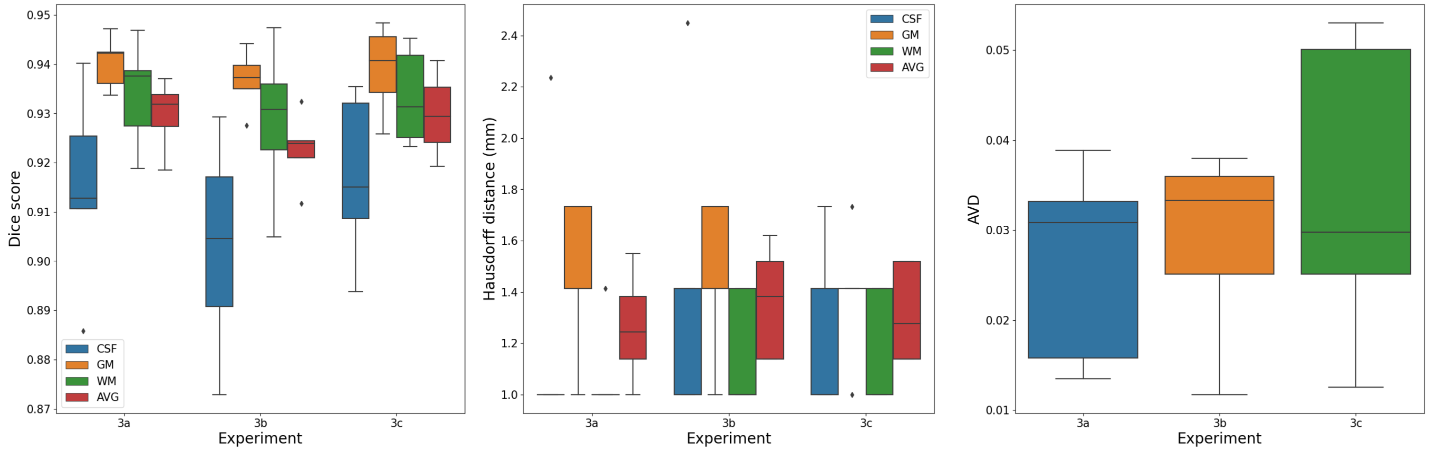


*Figure 11. Boxplots of results for experiments 3a, 3b, and 3c.*

We can see better results in experiment 3a, with the CE loss. Nevertheless, because of the data imbalance (figure 1), we decided to continue our experiments using focal loss, which is more suitable for imbalance segmentation problems and which had very similar results to cross entropy loss.

4. Patch size
   a. Patch size of 128x128 and stride of 32
   b. Patch size of 64x64  and stride of 32
   c. Patch size of 32x32 and stride of 16

Analyzing results for different patch sizes we noticed that for both patches of size 128 and 64 we get very similar results with some metrics being slightly higher for 128x128 patches (figure 12). However, CSF segmentation in them is considerably worse (see figure 13). This could be attributed to the fact that with patches of size 128 we had considerably less training data (despite reducing the stride to a quarter of the patch size) and therefore the model just didn't have enough data to reach the same performance.

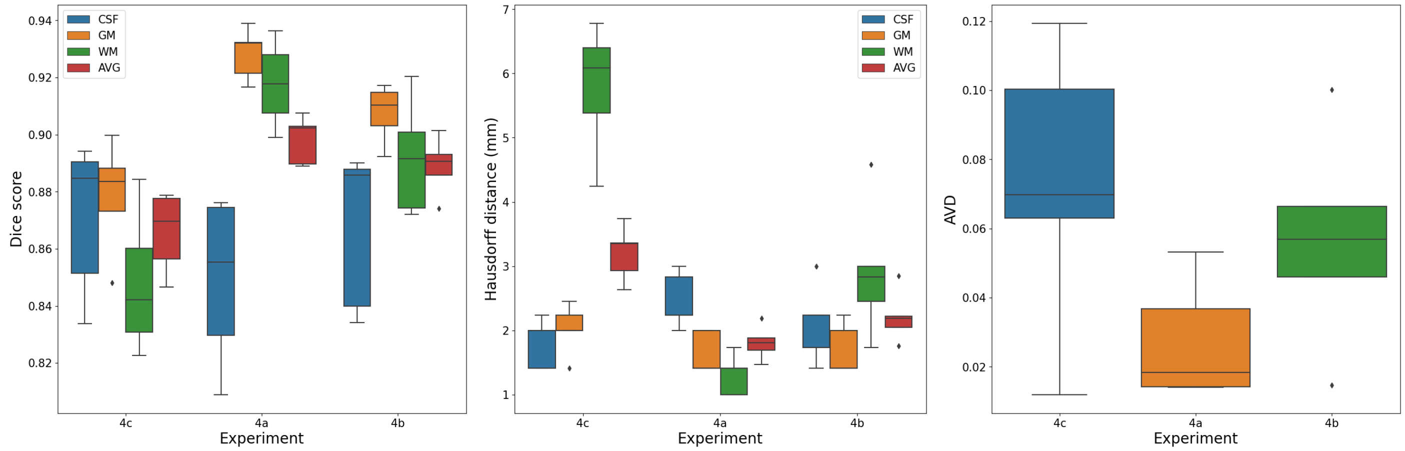


*Figure 12. Boxplots of results for experiments 4a, 4b, and 4c.*

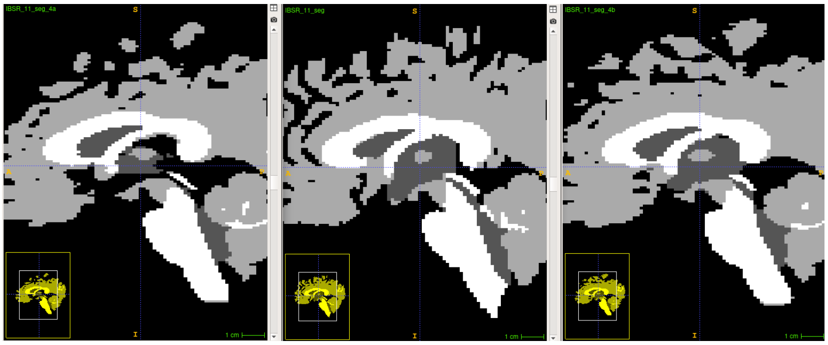


*Figure 13. Coronal slice of the segmentation masks from experiments 4.a (left), GT (middle) and 4.b (right). Darkest gray refers to CSF labels.*

5. Merging the labels for the SynthSeg prior segmentation
    a. Previous experiment 4b
    b. Previous experiment 4b with the prior input having 3 labels (only WM, GM, and CSF).

Finally, we have explored the idea of merging the labels from the original SynthSeg segmentation 32 to our 3. It would make the prior used for the model's training closer to the target segmentation and would allow faster and more efficient training (since the model would not need to learn first the mapping from 32 labels to 3 target ones). And as we can see in figure 14 this even further improved our results, proving our initial idea and reaching an average dice of 0.928839.

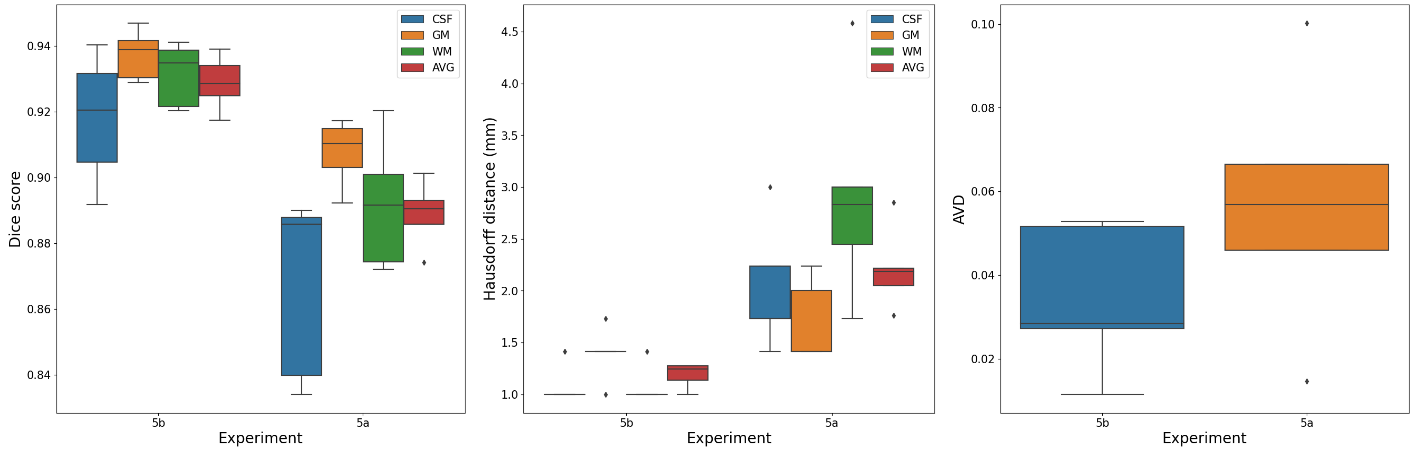


*Figure 14. Boxplots of results for experiments 5a and 5b.*

## Project management details (tasks, time estimations, and real dedication).

We equally contributed to the completion of this project, with an approximate working time of 20 hours each, considering school and housework. We worked collaboratively for the code implementation and experiments run.

We have to express our gratitude for the access to UdG servers provided for the duration of this lab that helped us considerably in the development of our solutions. One full training of the network for 100 epochs on one configuration took ~10 hours on a GPU, so having a few of them available was essential.

## Conclusions

In this project we have explored and developed a few approaches for the brain tissue segmentation of a particularly challenging dataset, that had low contrast and variable resolution images. Furthermore, we had an imbalance segmentation problem in which the minority class corresponding to CSF occupied ~1–1.7% of all GT-segmented pixels.

Our baseline approach was implemented through a multi-atlas weighted averaging model. For it, we explored different weights sets, based on a variety of registration similarity metrics and compared them to simple averaging of the propagated label maps. The results we obtained showed that there is no significant difference between averaging and weighted averaging (based on most of the similarity metrics). It shows that for this particular dataset even despite having groups of images of similar resolution this was not a definitive enough factor for preferring one atlas over others and with simple averaging we got the best result for this type of model with an average dice of 0.778.

Next, we tried one of the state-of-the-art models for brain segmentation that was specifically trained to deal with low-resolution and low signal-to-noise ratio images. We were interested to see how applicable it is in our case, considering that it was not trained on any real images. With a direct inference, we already obtained good results that were anatomically correct and after some post-processing close to our GT, we obtained an average dice of 0.828.

Finally, we have trained and explored a DL 2D patch-based U-Net style model for this task that produced the best results. We explored how the use of augmentations, different losses, and patch-slicing parameters could improve our segmentation results. Moreover, inspired by SynthSeg we tested a model trained not only on image patches but also with anatomical segmentation prior and showed that having this additional information helped the model to achieve better results. This approach proved to be the best one and reached an average dice across tissues and validation images of 0.928839.

## References

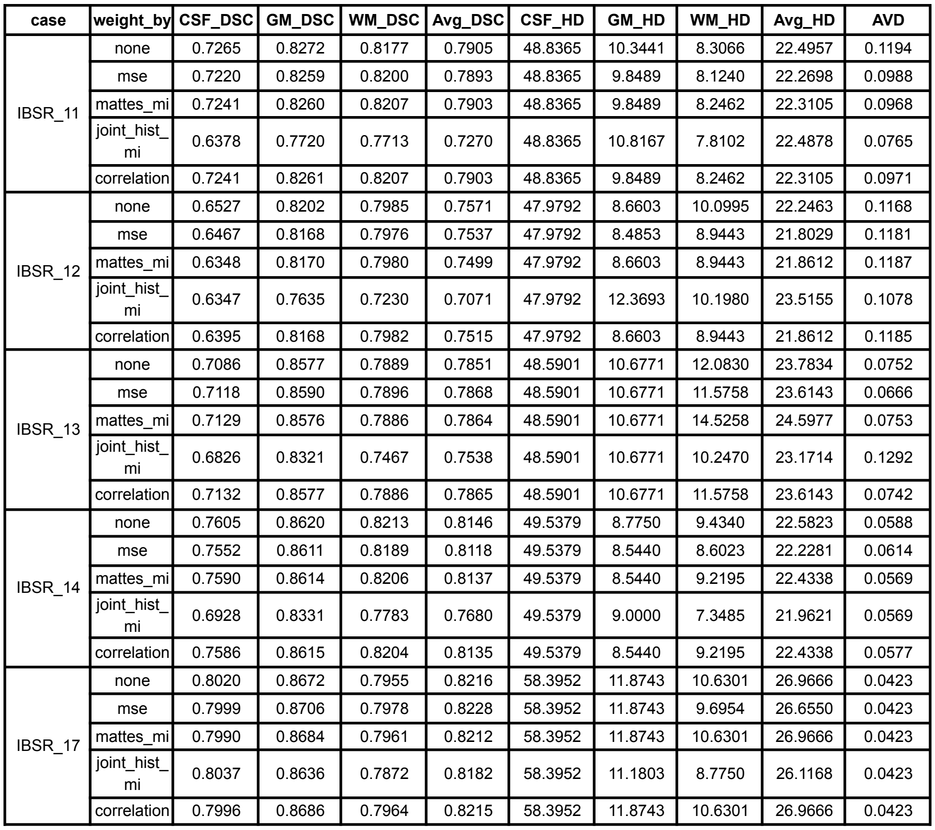
[1] Billot, B., Magdamo, C., Arnold, S. E., Das, S., & Iglesias, Juan. E. (2022). *Robust machine learning segmentation for large-scale analysis of heterogeneous clinical brain MRI datasets*. http://arxiv.org/abs/2209.02032

[2] H. Huang et al., "UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 1055-1059, doi: 10.1109/ICASSP40776.2020.9053405.

[3] https://github.com/ZJUGiveLab/UNet-Version

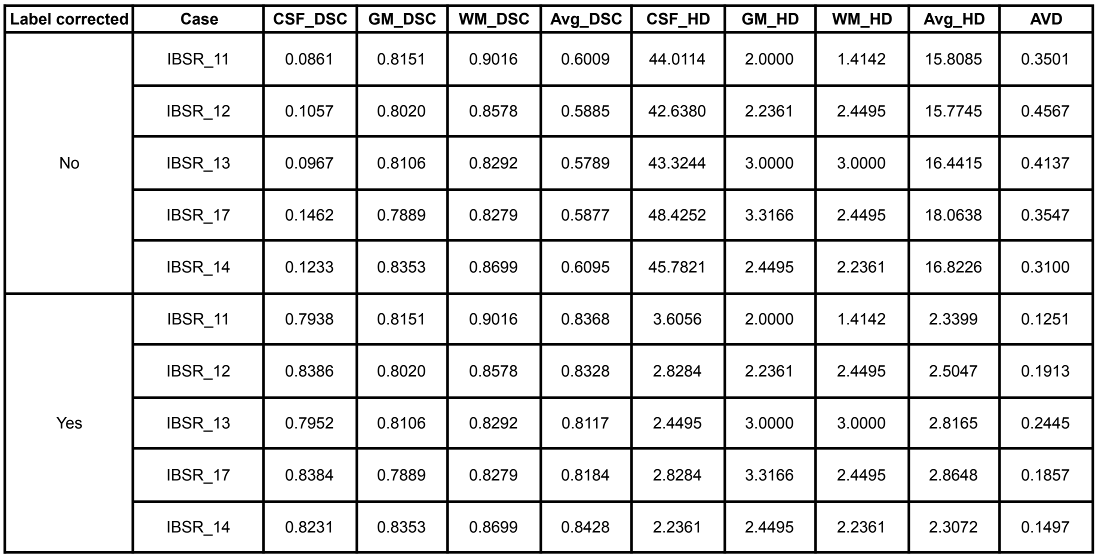
# Appendix A.

## A) Table with multi-atlas results



| case | weight_by | CSF_DSC | GM_DSC | WM_DSC | Avg_DSC | CSF_HD | GM_HD | WM_HD | Avg_HD | AVD |
|---|---|---|---|---|---|---|---|---|---|---|
| IBSR_11 | none | 0.7265 | 0.8272 | 0.8177 | 0.7905 | 48.8365 | 10.3441 | 8.3066 | 22.4957 | 0.1194 |
| | mse | 0.7220 | 0.8259 | 0.8200 | 0.7893 | 48.8365 | 9.8489 | 8.1240 | 22.2698 | 0.0988 |
| | mattes_mi | 0.7241 | 0.8260 | 0.8207 | 0.7903 | 48.8365 | 9.8489 | 8.2462 | 22.3105 | 0.0968 |
| | joint_hist_mi | 0.6378 | 0.7720 | 0.7713 | 0.7270 | 48.8365 | 10.8167 | 7.8102 | 22.4878 | 0.0765 |
| | correlation | 0.7241 | 0.8261 | 0.8207 | 0.7903 | 48.8365 | 9.8489 | 8.2462 | 22.3105 | 0.0971 |
| IBSR_12 | none | 0.6527 | 0.8202 | 0.7985 | 0.7571 | 47.9792 | 8.6603 | 10.0995 | 22.2463 | 0.1168 |
| | mse | 0.6467 | 0.8168 | 0.7976 | 0.7537 | 47.9792 | 8.4853 | 8.9443 | 21.8029 | 0.1181 |
| | mattes_mi | 0.6348 | 0.8170 | 0.7980 | 0.7499 | 47.9792 | 8.6603 | 8.9443 | 21.8612 | 0.1187 |
| | joint_hist_mi | 0.6347 | 0.7635 | 0.7230 | 0.7071 | 47.9792 | 12.3693 | 10.1980 | 23.5155 | 0.1078 |
| | correlation | 0.6395 | 0.8168 | 0.7982 | 0.7515 | 47.9792 | 8.6603 | 8.9443 | 21.8612 | 0.1185 |
| IBSR_13 | none | 0.7086 | 0.8577 | 0.7889 | 0.7851 | 48.5901 | 10.6771 | 12.0830 | 23.7834 | 0.0752 |
| | mse | 0.7118 | 0.8590 | 0.7896 | 0.7868 | 48.5901 | 10.6771 | 11.5758 | 23.6143 | 0.0666 |
| | mattes_mi | 0.7129 | 0.8576 | 0.7886 | 0.7864 | 48.5901 | 10.6771 | 14.5258 | 24.5977 | 0.0753 |
| | joint_hist_mi | 0.6826 | 0.8321 | 0.7467 | 0.7538 | 48.5901 | 10.6771 | 10.2470 | 23.1714 | 0.1292 |
| | correlation | 0.7132 | 0.8577 | 0.7886 | 0.7865 | 48.5901 | 10.6771 | 11.5758 | 23.6143 | 0.0742 |
| IBSR_14 | none | 0.7605 | 0.8620 | 0.8213 | 0.8146 | 49.5379 | 8.7750 | 9.4340 | 22.5823 | 0.0588 |
| | mse | 0.7552 | 0.8611 | 0.8189 | 0.8118 | 49.5379 | 8.5440 | 8.6023 | 22.2281 | 0.0614 |
| | mattes_mi | 0.7590 | 0.8614 | 0.8206 | 0.8137 | 49.5379 | 8.5440 | 9.2195 | 22.4338 | 0.0569 |
| | joint_hist_mi | 0.6928 | 0.8331 | 0.7783 | 0.7680 | 49.5379 | 9.0000 | 7.3485 | 21.9621 | 0.0569 |
| | correlation | 0.7586 | 0.8615 | 0.8204 | 0.8135 | 49.5379 | 8.5440 | 9.2195 | 22.4338 | 0.0577 |
| IBSR_17 | none | 0.8020 | 0.8672 | 0.7955 | 0.8216 | 58.3952 | 11.8743 | 10.6301 | 26.9666 | 0.0423 |
| | mse | 0.7999 | 0.8706 | 0.7978 | 0.8228 | 58.3952 | 11.8743 | 9.6954 | 26.6550 | 0.0423 |
| | mattes_mi | 0.7990 | 0.8684 | 0.7961 | 0.8212 | 58.3952 | 11.8743 | 10.6301 | 26.9666 | 0.0423 |
| | joint_hist_mi | 0.8037 | 0.8636 | 0.7872 | 0.8182 | 58.3952 | 11.1803 | 8.7750 | 26.1168 | 0.0423 |
| | correlation | 0.7996 | 0.8686 | 0.7964 | 0.8215 | 58.3952 | 11.8743 | 10.6301 | 26.9666 | 0.0423 |

## B) Table with SynthSeg results



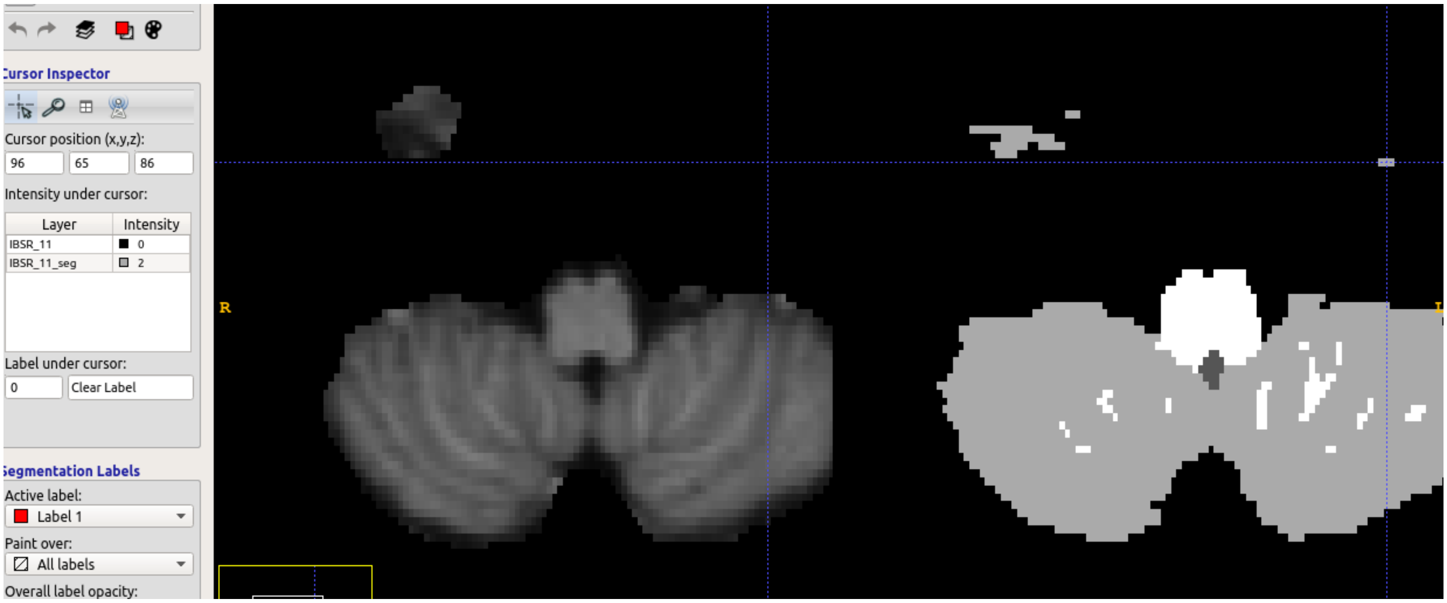
| Label corrected | Case | CSF_DSC | GM_DSC | WM_DSC | Avg_DSC | CSF_HD | GM_HD | WM_HD | Avg_HD | AVD |
|---|---|---|---|---|---|---|---|---|---|---|
| No | IBSR_11 | 0.0861 | 0.8151 | 0.9016 | 0.6009 | 44.0114 | 2.0000 | 1.4142 | 15.8085 | 0.3501 |
| | IBSR_12 | 0.1057 | 0.8020 | 0.8578 | 0.5885 | 42.6380 | 2.2361 | 2.4495 | 15.7745 | 0.4567 |
| | IBSR_13 | 0.0967 | 0.8106 | 0.8292 | 0.5789 | 43.3244 | 3.0000 | 3.0000 | 16.4415 | 0.4137 |
| | IBSR_17 | 0.1462 | 0.7889 | 0.8279 | 0.5877 | 48.4252 | 3.3166 | 2.4495 | 18.0638 | 0.3547 |
| | IBSR_14 | 0.1233 | 0.8353 | 0.8699 | 0.6095 | 45.7821 | 2.4495 | 2.2361 | 16.8226 | 0.3100 |
| Yes | IBSR_11 | 0.7938 | 0.8151 | 0.9016 | 0.8368 | 3.6056 | 2.0000 | 1.4142 | 2.3399 | 0.1251 |
| | IBSR_12 | 0.8386 | 0.8020 | 0.8578 | 0.8328 | 2.8284 | 2.2361 | 2.4495 | 2.5047 | 0.1913 |
| | IBSR_13 | 0.7952 | 0.8106 | 0.8292 | 0.8117 | 2.4495 | 3.0000 | 3.0000 | 2.8165 | 0.2445 |
| | IBSR_17 | 0.8384 | 0.7889 | 0.8279 | 0.8184 | 2.8284 | 3.3166 | 2.4495 | 2.8648 | 0.1857 |
| | IBSR_14 | 0.8231 | 0.8353 | 0.8699 | 0.8428 | 2.2361 | 2.4495 | 2.2361 | 2.3072 | 0.1497 |

C) Table with best U-Net 3+ results (experiment 5b).

| Experiment | Case | CSF_DSC | GM_DSC | WM_DSC | Avg_DSC | CSF_HD | GM_HD | WM_HD | Avg_HD | AVD |
|---|---|---|---|---|---|---|---|---|---|---|
| 5b | IBSR_11 | 0.9048 | 0.9289 | 0.9411 | 0.9249 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0516 |
| | IBSR_12 | 0.9205 | 0.9303 | 0.9349 | 0.9286 | 1.4142 | 1.4142 | 1.0000 | 1.2761 | 0.0285 |
| | IBSR_13 | 0.8919 | 0.9389 | 0.9217 | 0.9175 | 1.0000 | 1.7321 | 1.0000 | 1.2440 | 0.0528 |
| | IBSR_17 | 0.9316 | 0.9470 | 0.9388 | 0.9391 | 1.0000 | 1.4142 | 1.0000 | 1.1381 | 0.0272 |
| | IBSR_14 | 0.9404 | 0.9416 | 0.9203 | 0.9341 | 1.0000 | 1.4142 | 1.4142 | 1.2761 | 0.0115 |

# Appendix B.

Ground truth segmentations, examples of errors.



Cursor position (x,y,z): 96 65 86

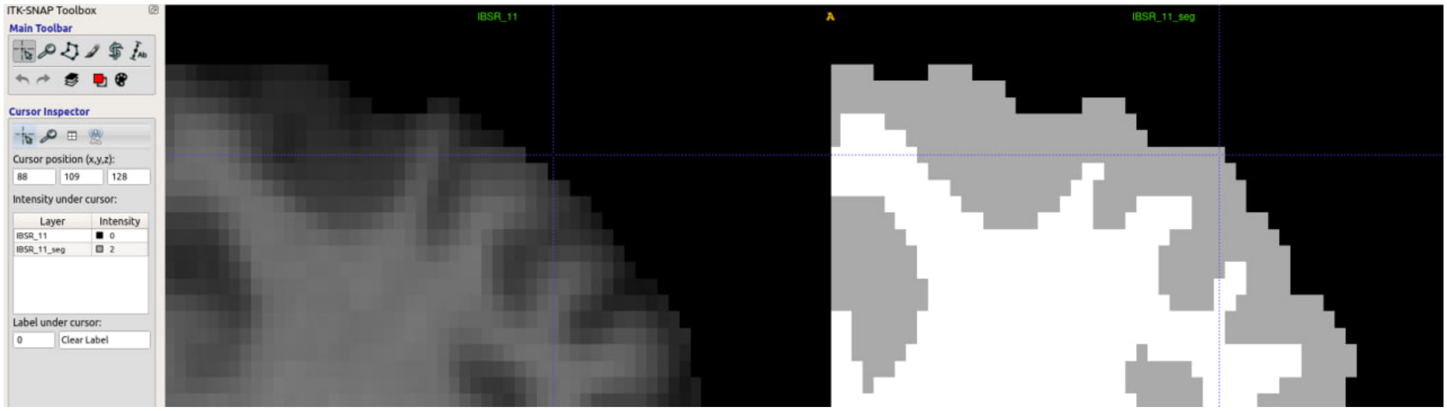
Intensity under cursor:

| Layer | Intensity |
|---|---|
| IBSR_11 | ■ 0 |
| IBSR_11_seg | ▨ 2 |

Label under cursor: 0 Clear Label

Active label: Label 1

Paint over: All labels



Cursor position (x,y,z): 88 109 128

Intensity under cursor:

| Layer | Intensity |
|---|---|
| IBSR_11 | ■ 0 |
| IBSR_11_seg | ▨ 2 |

Label under cursor: 0 Clear Label