Medical Image Segmentation and Applications:
Final Project

Leonardo Pestana Legori[1]
leonardopl@outlook.com

Lisle Faray de Paiva[1]
lisle@nca.ufma.br

[1]Student at University of Girona in the Medical Imaging and Applications MSc program

***Abstract*—The goal of this project was to develop and explore methods for the segmentation of tissues of brain MRI images into cerebrospinal fluid (CSF), gray matter (GM) and white matter (WM). We accomplished that with the use of a multi-atlas approach, as well as with multiple different deep learning-based methods. The method with the highest prediction score was segmentation with nn-Unet semantic segmentation framework, which works by automatically creating a deep-learning pipeline optimized to our dataset characteristics. For the validation cases, we achieved a mean DSC score of 0.942 ± 0.02, a mean HD score of 8.0003 ± 2.4 and a mean AVD score of 0.0372 ± 0.04.**

***Index Terms*—segmentation, MRI, brain, medical imaging, multi-atlas, deep learning**

1 INTRODUCTION AND PROBLEM DEFINITION

The aim of this project was to develop and explore methods for the segmentation of tissues of brain MRI images into cerebrospinal fluid (CSF), gray matter (GM) and white matter (WM). To do that two main types of methods were pursued: multi-atlas segmentation and deep learning-based segmentation.

The IBSR18 dataset was used to develop the algorithms and train the deep-learning models, as well as to evaluate the results.

The results were evaluated and compared using Dice similarity coefficient (DSC), Hausdorff distance (HD) and average volumetric difference (AVD) scores.

2 ALGORITHM ANALYSIS, DESIGN, AND IMPLEMENTATION OF THE PROPOSED SOLUTION

2.1 Multi-atlas

To perform segmentation using a multi-atlas approach, the following steps were taken:

1. Preprocessing of training images.
2. Registration of each training image to each of the validation images.
3. Transformation of the training labels.
4. Label fusion using weighted voting.

For the preprocessing, we resampled each training image and corresponding ground-truth label to have the same spacing, direction, origin, and pixel type as the target image. A very important parameter to set was the type of interpolator. For the images we used a B-spline interpolator and for the labels a k-Nearest Neighbor interpolator. This way, we could ensure that the range for the labels did not change during the resample operation. Next, we matched the intensity distribution of the training images to the target images, doing a histogram matching operation, which was a crucial step for improving the accuracy and robustness of the image registration process, especially when dealing with images from different sources.

To register the images, we used elastix [1][2], with parameter Par0010 obtained from elastix's Model Zoo [3], which included affine and B-spline transformations files using mutual information (MI) as metric.

The transformation of the labels was done using transformix using the output parameter maps of the registrations.

For the label fusion, the first step was to calculate the metrics to be used as weights. For each pair of training and validation images, we calculated the MI, the mean squared error (MSE), the peak-signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) between them. Next, for each validation case, we proceed to generate what is in essence a probabilistic atlas made from the training propagated labels. To do that, for each brain tissue label, we do a weighted average operation with the weights we calculated before. That means that the contribution to the final probabilistic atlas depends on the training image similarity with the validation target image. This operation needed to be repeated for each metric we are using as weight. Additionally, we also created an atlas using equal weights for each training image, for comparison. The last remaining step was to get the final segmentation masks, one for each validation case and metric pair. That was simply done by doing an argmax operation using the calculated probabilities for each voxel.

2.2 Deep learning

For the deep learning-based segmentation approached we designed four experiments, each using a different model or method:

1. Using SegResNetDS.
2. Using SwinUNETR-V2.
3. Using Auto3DSeg.
4. Using nnU-Net.

For all experiments, we used MONAI framework [4] to implement the preprocessing pipeline, to instantiate the models and to interface with Auto3DSeg and nnU-Net. To implement the training process PyTorch was used.

2.2.1 Segmentation with SegResNetDS

For this experiment we used the SegResNetDS model implemented by MONAI based on the work by [5], which describes a 3D segmentation network based on encoder-decoder architecture. This specific MONAI implementation contains some improvements upon the original work, including deep supervision (hence the DS in the model’s name) and non-isotropic kernel support.

As part of the preprocessing and dataset loading pipeline for the training and validation sets, we z-score normalize the images and resample them, so they have a voxel spacing of (1.0, 1.0, 1.0). Next, only for the training set, we crop the image based on the foreground pixels, with the output image having a dimension divisible by our region of interest (ROI) size, which was set to (128, 128, 128). Following that we perform a random cropping of the images and labels to the ROI size, where the center is adjusted to be based on the label classes ratio and each image is cropped four times, generating four patches per image. Next, we perform the following random operations:

- random flipping of the image in all axes, with a probability of 0.2.
- random 90 degrees rotation of the image, with a probability of 0.2.
- random scale of intensity by factor of 0.1, with a probability of 0.1.
- random shift of intensity by factor of 0.1, with a probability of 0.1.

We have set a batch size of 8 for training and 4 for inference. The maximum number of epochs was set to 1000 with a validation interval of 1 epoch, and a patience of 20 epochs for early stopping based on the average DSC. The model was instantiated with the number of downsample blocks in each layer set to [1, 2, 2, 4, 4] and a deep supervision depth of 1.

We used the weighted sum of the Dice loss and Cross Entropy Loss as the loss function. For the optimizer AdamW was chosen, with a learning rate of 2^{-4} and weight decay of 1^{-5}. We also made use of a linear warmup and then cosine decay learning rate scheduler.

2.2.2 Segmentation with SwinUNETR-V2

SwinUNETR-V2, as implemented in MONAI, is based on the work of [6] and [7]. The model implements a U-shaped network with a Swin transformer as the encoder which is connected to a CNN-based decoder of different resolutions via skip connections [6]. The version 2 of this model improves on the original design by adding a residual convolution block at the beginning of each Swin stage [7].

In this experiment, we keep most of the pipeline and hyperparameters values of the last experiment. The only change is in the ROI for the patches, which was set to (96, 96, 96). This decrease was needed due to the fact this model is more computationally expensive.

Only on this experiment we manually stopped the training at epoch 172 since we did not see improvement of the metric in many epochs.

2.2.3 Segmentation with Auto3DSeg

This experiment made use of MONAI's Auto3DSeg [8], a unified framework for 3D medical image segmentation with minimal user input. It works by analyzing the dataset and, based on the global information (intensity, data size, data spacing etc.), it generates a segmentation pipeline optimized for it [9]. Based on the input data, it can train up to four different models: DiNTS [10], 2D SegResNet, 3D SegResNet and SwinUNETR. The trained checkpoints are then ranked and the models with the best results are used in an ensemble to get the final predictions.

Using it in our project was a simple operation. The only required inputs were the modality of the images, a JSON file with the filenames of the cases, and the dataset directory. Since we already have the dataset split in train, validation, and test cases, we had to manually set it to train using only one-fold with the original data split.

Based on our data, the framework decided to make use of the DiNTS, 3D SegResNet and SwinUNETR models. At the end of the training, it decided to use only 3D SegResNet for the prediction of the test cases, instead of doing an ensemble.

2.2.4 Segmentation with nn-Unet

The final experiment made use of nn-Unet [11], which, like Auto3DSeg, is an automated framework enabling semantic segmentation based on the dataset and works much like the same. Based on the input dataset, it creates several optimized U-Net configurations, which can be any of the following:

- *2d* – a 2D U-Net for 2D and 3D datasets.
- *3d_fullres* – a 3D U-Net using high resolution images for training, only for 3D datasets.
- *3d_lowres/3d_cascade_fullres* – a 3D U-Net cascade operating first on low resolution images and then on high resolution for refinement of predictions, for 3D datasets with large images only.

There is already an interface for using nn-Unet in MONAI, so we decided to make use of that for this project.

To use it, we had to provide a JSON file with the filenames and the cases, the modality type, the dataset directory, and the working directory.

Differently from Auto3DSeg, we could not easily specify the splits for the training, so we let the nn-Unet algorithm decide the splits using only the training set cases. With that, it configured a 5-fold cross-validation splits based only on the training set, with each fold containing 8 training cases and 2 validation cases, which can be seen in table 1. Our original validation set was only used for final inference, together with the test set.

Table 1: Fold split configuration for nn-Unet

Fold	Training cases	Validation cases
0	3, 4, 5, 6, 7, 8, 16, 18	1, 9
1	1, 3, 4, 6, 7, 8, 9, 16	5, 18
2	1, 3, 4, 5, 7, 9, 16, 18	6, 8
3	1, 4, 5, 6, 7, 8, 9, 18	3, 16
4	1, 3, 5, 6, 8, 9, 16, 18	4, 7

For our dataset, the algorithm decided to generate two U-Net configurations: a 2D (*2d*) and a 3D one using high resolution images (*3d_fullres*).

For the final prediction of the validation and test cases, the model chose to use an ensemble of both trained models.

3 Experiments and Result Analysis

All the code was run and benchmarked on a laptop equipped with a processor Intel Core i7-11800H, 64GB of RAM and a NVIDIA 3080 graphics card with 16GB of VRAM.

3.1 Multi-atlas

First, we will report on the average scores across all validation cases and all brain tissue types using each metric used for weighting during label fusion, which can be seen on table 2.

Table 2: Average scores per weighting metric for all tissue types across all validation cases

	DSC	HD	AVD
MI	0.8024 ± 0.02	12.6650 ± 4.62	0.1331 ± 0.07
MSE	0.8019 ± 0.02	12.4538 ± 4.62	0.1313 ± 0.07
PSNR	0.8024 ± 0.02	12.6073 ± 4.54	0.1325 ± 0.06
SSIM	0.8024 ± 0.02	12.6073 ± 4.54	0.1325 ± 0.07
Equal weights	0.8024 ± 0.02	12.9179 ± 4.64	0.1530 ± 0.08

There was no real impact on the segmentation results based on the metric type used as weight during the label fusion step. The results are nearly identical, even when using equal weights for all atlases. Weighing with MSE produced the best HD and AVD results while being the only metric with slightly lower dice score.

In tables 3, 4 and 5 we provide the DSC, HD, and AVD results, respectively, for each brain tissue type (CSF, GM, and WM).

Table 3: DSC average scores per weighting metric for each tissue type across all validation cases

	CSF	GM	WM
MI	0.8026 ± 0.04	0.8123 ± 0.03	0.7922 ± 0.01
MSE	0.8012 ± 0.04	0.8125 ± 0.03	0.7918 ± 0.01
PSNR	0.8021 ± 0.04	0.8126 ± 0.03	0.7924 ± 0.01
SSIM	0.8029 ± 0.04	0.8123 ± 0.03	0.792 ± 0.01
Equal weights	0.8007 ± 0.04	0.8168 ± 0.03	0.7896 ± 0.01

Table 4: HD average scores per weighting metric for each tissue type across all validation cases

	CSF	GM	WM
MI	19.1250 ± 4.1	9.3822 ± 1.72	9.4880 ± 0.99
MSE	18.9517 ± 4.01	9.2603 ± 1.52	9.1493 ± 0.93
PSNR	18.9517 ± 4.01	9.3822 ± 1.72	9.4880 ± 0.99
SSIM	18.9517 ± 4.01	9.3822 ± 1.72	9.4880 ± 0.99
Equal weights	19.4017 ± 4.24	9.3822 ± 1.72	9.9698 ± 1.12

Table 5: AVD average scores per weighting metric for each tissue type across all validation cases

	CSF	GM	WM
MI	0.1227 ± 0.03	0.1993 ± 0.11	0.0774 ± 0.05
MSE	0.1187 ± 0.04	0.2014 ± 0.11	0.0740 ± 0.05
PSNR	0.1214 ± 0.04	0.2002 ± 0.11	0.0758 ± 0.05
SSIM	0.1203 ± 0.04	0.1997 ± 0.11	0.0776 ± 0.05
Equal weights	0.1473 ± 0.09	0.2340 ± 0.12	0.0776 ± 0.07

The multi-atlas approached produced consistent DSC results, with little variation across tissue types. The same cannot be said when looking at the HD scores, with the CSF tissue having on average double the HD when compared to GM and WM. For AVD, the best results were with WM, the second best with CSF and the worst with GM tissues.

3.2 Deep learning

3.2.1 Segmentation with SegResNetDS

The results of the segmentation using SegResNetDS can be seen in tables 6, 7 and 8.

Table 6: DSC scores for each tissue type (SegResNetDS)

Case	CSF	GM	WM	Total per case
11	0.8782	0.9330	0.9473	**0.9195 ± 0.04**
12	0.9005	0.9176	0.9297	**0.9159 ± 0.01**
13	0.8764	0.9323	0.9075	**0.9054 ± 0.03**
14	0.9192	0.9472	0.9429	**0.9364 ± 0.02**
17	0.9281	0.9403	0.9185	**0.9290 ± 0.01**
Total per tissue	**0.9005 ± 0.02**	**0.9341 ± 0.01**	**0.9292 ± 0.02**	**0.9212 ± 0.02**

Table 7: HD scores for each tissue type (SegResNetDS)

Case	CSF	GM	WM	Total per case
11	10.0995	7.1414	7.5498	**8.2636 ± 1.60**
12	16.2788	9.2736	8.0623	**11.2049 ± 4.44**
13	14.8997	11.0000	8.2462	**11.3820 ± 3.34**
14	35.9166	8.6023	6.5574	**17.0254 ± 16.39**
17	14.2478	8.6603	9.4868	**10.7983 ± 3.02**
Total per tissue	**18.2885 ± 10.12**	**8.9355 ± 1.39**	**7.9805 ± 1.07**	**11.7348 ± 7.3**

Table 8: AVD scores for each tissue type (SegResNetDS)

Case	CSF	GM	WM	Total per case
11	0.0329	0.0012	0.0462	**0.0268 ± 0.02**
12	0.1264	0.0218	0.0802	**0.0761 ± 0.05**
13	0.0757	0.0554	0.1004	**0.0771 ± 0.02**
14	0.0080	0.0169	0.0040	**0.0096 ± 0.01**
17	0.0032	0.0239	0.0481	**0.025 ± 0.02**
Total per tissue	**0.0492 ± 0.05**	**0.0238 ± 0.02**	**0.0558 ± 0.04**	**0.0429 ± 0.04**

The results obtained with this method were better, compared with the multi-atlas segmentation. On average, the DSC scores and the HD scores are 15% and 9% better, respectively. The AVD score did see a more substantial increase, with 72% better results. This improvement is also seen for all types of brain tissues (CSF, GM, and WM).

3.2.2 Segmentation with SwinUNETR-V2

The results of the segmentation using SwinUNETR-V2 can be seen in tables 9, 10 and 11.

Table 9: DSC scores for each tissue type (SwinUNETR-V2)

Case	CSF	GM	WM	Total per case
11	0.8760	0.7477	0.8892	**0.8376 ± 0.08**
12	0.9056	0.7379	0.8781	**0.8405 ± 0.09**
13	0.8695	0.8328	0.8599	**0.8541 ± 0.02**
14	0.9088	0.9039	0.9183	**0.9103 ± 0.01**
17	0.9181	0.9006	0.9201	**0.9129 ± 0.01**
Total per tissue	**0.8956 ± 0.02**	**0.8246 ± 0.08**	**0.8931 ± 0.03**	**0.8711 ± 0.06**

Table 10: HD scores for each tissue type (SwinUNETR-V2)

Case	CSF	GM	WM	Total per case
11	135.7792	130.7020	127.7263	**131.4025 ± 4.07**
12	139.7891	133.9963	132.3669	**135.3841 ± 3.90**
13	134.5548	129.2633	130.5795	**131.4659 ± 2.75**
14	13.6382	129.7074	124.1169	**89.1541 ± 65.46**
17	18.1384	119.5324	116.5333	**84.7347 ± 57.69**
Total per tissue	**88.3799 ± 66.22**	**128.6403 ± 5.42**	**126.2646 ± 6.27**	**114.4283 ± 40.46**

Table 11: AVD scores for each tissue type (SwinUNETR-V2)

Case	CSF	GM	WM	Total per case
11	0.0504	0.4479	0.1121	**0.2035 ± 0.21**
12	0.0573	0.4753	0.1940	**0.2422 ± 0.21**
13	0.0172	0.1193	0.2082	**0.1149 ± 0.10**
14	0.1122	0.0920	0.0284	**0.0775 ± 0.04**
17	0.0385	0.0356	0.0057	**0.0266 ± 0.02**
Total per tissue	**0.0551 ± 0.04**	**0.2340 ± 0.21**	**0.1097 ± 0.09**	**0.1329 ± 0.15**

The results are worse than with SegResNetDS. The DSC scores were, on average, 0.0501 lower compared to the previous experiment. On the other hand, the HD and AVD were significantly worse, being an order of magnitude higher than previous results.

These findings suggest that our trained SwinUNETR-V2 model is less accurate, both in terms of the location and the size of the segmented areas, compared to SegResNetDS. This could be due to various factors such as differences in model architecture, or the need for more suitable parameter tuning and preprocessing steps for this model. As an improvement for future work, we may require revisiting these aspects to understand why the performance has degraded and to identify potential areas for improvement.

3.2.3 *Segmentation with Auto3DSeg*

The results of the segmentation using Auto3DSeg can be seen in tables 12, 13 and 14.

Table 12: DSC scores for each tissue type (Auto3DSeg)

Case	CSF	GM	WM	Total per case
11	0.8990	0.9485	0.9582	**0.9352 ± 0.03**
12	0.9060	0.9332	0.9333	**0.9242 ± 0.02**
13	0.8989	0.9428	0.9166	**0.9194 ± 0.02**
14	0.9225	0.9567	0.9513	**0.9435 ± 0.02**
17	0.9381	0.9527	0.9399	**0.9436 ± 0.01**
Total per tissue	**0.9129 ± 0.02**	**0.9468 ± 0.01**	**0.9399 ± 0.02**	**0.9332 ± 0.02**

Table 13: HD scores for each tissue type (Auto3DSeg)

Case	CSF	GM	WM	Total per case
11	8.7750	7.0711	6.4807	**7.4423 ± 1.19**
12	9.4868	7.0000	10.0499	**8.8456 ± 1.62**
13	11.2250	8.0000	12.9615	**10.7288 ± 2.52**
14	6.3246	7.5498	6.4031	**6.7592 ± 0.69**
17	11.1803	6.1644	8.6603	**8.6683 ± 2.51**
Total per tissue	**9.3983 ± 2.02**	**7.1571 ± 0.69**	**8.9111 ± 2.74**	**8.4888 ± 2.11**

Table 14: AVD scores for each tissue type (Auto3DSeg)

Case	CSF	GM	WM	Total per case
11	0.0279	0.0164	0.0143	**0.0195 ± 0.01**
12	0.1291	0.0234	0.0941	**0.0822 ± 0.05**
13	0.0149	0.0555	0.1128	**0.0611 ± 0.05**
14	0.1002	0.0172	0.0137	**0.0437 ± 0.05**
17	0.0256	0.0078	0.0139	**0.0157 ± 0.01**
Total per tissue	**0.0595 ± 0.05**	**0.0241 ± 0.02**	**0.0498 ± 0.05**	**0.0445 ± 0.04**

The results are better than in previous experiments. We have achieved a higher DSC in all brain tissues. HD scores also have improved, especially for CSF, where we got a 48.6% decrease compared with the previous best result. The AVD results remained in line with the previous best results, being only slightly worse, with a difference of 0.0016 on the total average.

3.2.4 *Segmentation with nn-Unet*

The results of the segmentation using nn-Unet can be seen in tables 15, 16 and 17.

Table 15: DSC scores for each tissue type (nn-Unet)

Case	CSF	GM	WM	Total per case
11	0.9125	0.9545	0.9622	**0.9431 ± 0.03**
12	0.8998	0.9470	0.9511	**0.9326 ± 0.03**
13	0.9069	0.9463	0.9312	**0.9281 ± 0.02**
14	0.9376	0.9652	0.9601	**0.9543 ± 0.01**
17	0.9530	0.9565	0.9464	**0.9520 ± 0.01**
Total per tissue	**0.9220 ± 0.02**	**0.9539 ± 0.01**	**0.9502 ± 0.01**	**0.9420 ± 0.02**

Table 16: HD scores for each tissue type (nn-Unet)

Case	CSF	GM	WM	Total per case
11	10.0499	7.8740	5.7446	**7.8895 ± 2.15**
12	10.2470	6.7082	5.3852	**7.4468 ± 2.51**
13	10.4403	11.0454	8.6023	**10.0293 ± 1.27**
14	4.2426	5.0990	6.0000	**5.1139 ± 0.88**
17	11.9164	8.7750	7.8740	**9.5218 ± 2.12**
Total per tissue	**9.3792 ± 2.96**	**7.9003 ± 2.23**	**6.7212 ± 1.43**	**8.0003 ± 2.40**

Table 17: AVD scores for each tissue type (nn-Unet)

Case	CSF	GM	WM	Total per case
11	0.0325	0.0142	0.0054	**0.0174 ± 0.01**
12	0.1413	0.0082	0.0465	**0.0653 ± 0.07**
13	0.0209	0.0573	0.0547	**0.0443 ± 0.02**
14	0.0777	0.0128	0.0378	**0.0428 ± 0.03**
17	0.0083	0.0001	0.0402	**0.0162 ± 0.02**
Total per tissue	**0.0561 ± 0.05**	**0.0185 ± 0.02**	**0.0369 ± 0.02**	**0.0372 ± 0.04**

The results were the best we achieved in this project. This time, looking at the total average results, we improved in all metrics. For the DSC score we got an increase in all the types of tissues.

We hypothesize that this increase is due to this being the only method where we trained multiple models with a 5-fold cross-validation strategy and used their prediction in an assembly. Of course, the automatic auto-tunning of the model to the specific attributes of our dataset was also a critical factor to achieve such good performance.

The only limiting factor of this approach was the high computational time that was needed to train all the 10 models (two for each fold). It took approximately 262 hours, or about 11 days to complete the training.

3.3 Qualitative results

Due to space constraints, we show the qualitative results of all the methods employed in the Appendix A. For the multi-atlas method, we only show the best segmentation results, which were obtained with MSE as the weighting factor for the label fusion.

4 CONCLUSION

In this project coursework we set to develop methods to segment brain MRI images into CSF, GM, and WM. We developed and tested multiple methods to accomplish that. Two main types of methods were pursued: multi-atlas-based segmentation and deep learning-based segmentation.

Our best results using a multi-atlas approach were produced using MSE metric for weighted label fusion. With that, we managed to get prediction scores of 0.8019 ± 0.02, 12.4538 ± 4.62, and 0.1313 ± 0.07, using DSC, HD and AVD metrics, respectively. Although that was the best result with a multi-atlas approach, it was basically the same as using other metrics and using equal weights for label fusion. For possible future improvements of this approach, we may investigate more effective approaches for label fusion, such as using a non-local means approach.

For the deep leaning-based segmentation approach we tried two models with different architectures (SegResNetDS and SwinUNETR-V2) and two automatic segmentation frameworks (AutoSeg3D and nn-Unet), which in turn, made use of many other models. After our experiments, we achieved the best results with nn-Unet, with which we obtained prediction scores of 0.9420 ± 0.02, 8.0003 ± 2.40 and 0.0372 ± 0.04, using DSC, HD and AVD metrics, respectively. The results show the remarkable potential of such automated frameworks. We hypothesize that such strong result may be especially due to the ensemble prediction made of all the models trained with cross-validation and also due to the optimization of the network and the preprocessing of the data to account for the particularities of our dataset.

5 REFERENCES

[1] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. W. Pluim, “Elastix: A toolbox for intensity-based medical image registration,” *IEEE Trans Med Imaging*, vol. 29, no. 1, pp. 196–205, Jan. 2010, doi: 10.1109/TMI.2009.2035616.

[2] D. P. Shamonin, E. E. Bron, B. P. F. Lelieveldt, M. Smits, S. Klein, and M. Staring, “Fast parallel image registration on CPU and GPU for diagnostic classification of Alzheimer’s disease,” *Front Neuroinform*, vol. 7, no. JAN, p. 66699, Jan. 2014, doi: 10.3389/FNINF.2013.00050/ABSTRACT.

[3] “Model Zoo Elastix.” Accessed: Jan. 12, 2024. [Online]. Available: https://elastix.lumc.nl/modelzoo/

[4] M. Jorge Cardoso *et al.*, “MONAI: An open-source framework for deep learning in healthcare,” 2022.

[5] A. Myronenko, “3D MRI brain tumor segmentation using autoencoder regularization,” Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.11654

[6] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu, “Swin UNETR: Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images”, Accessed: Jan. 11, 2024. [Online]. Available: https://monai.io/research/swin-unetr

[7] Y. He, V. Nath, D. Yang, Y. Tang, A. Myronenko, and D. Xu, “SwinUNETR-V2: Stronger Swin Transformers with Stagewise Convolutions for 3D Medical Image Segmentation,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14223 LNCS, pp. 416–426, 2023, doi: 10.1007/978-3-031-43901-8_40/TABLES/5.

[8] “tutorials/auto3dseg at main · Project-MONAI/tutorials.” Accessed: Jan. 12, 2024. [Online]. Available: https://github.com/Project-MONAI/tutorials/tree/main/auto3dseg

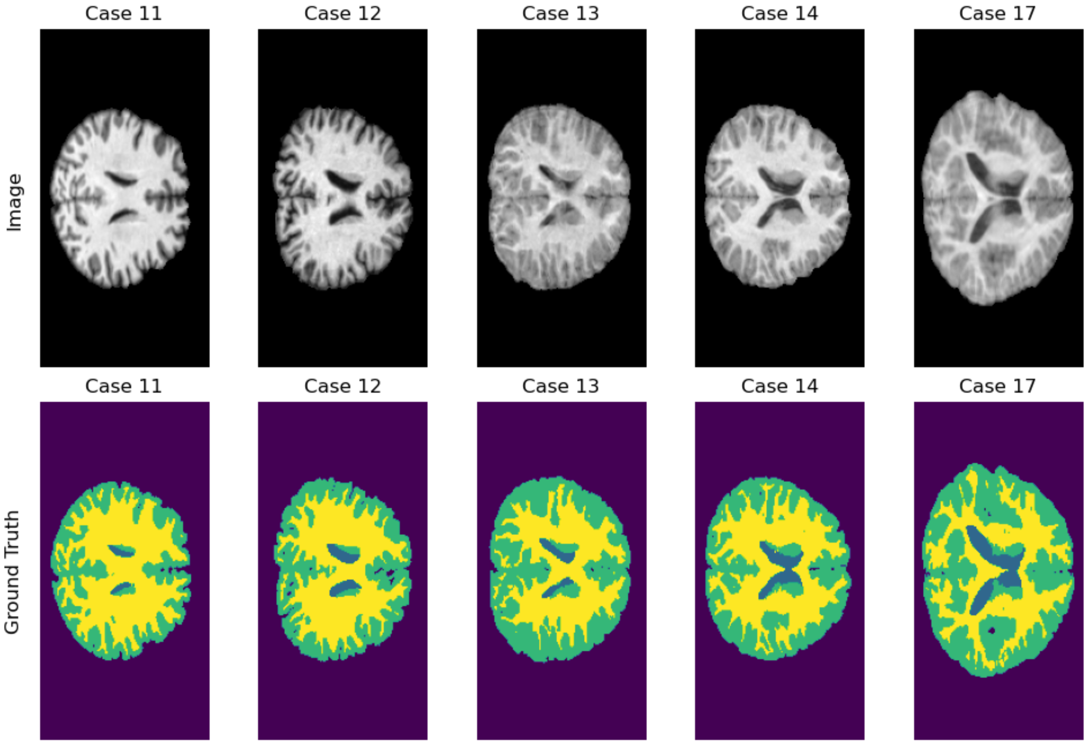
[9] “tutorials/auto3dseg/docs/algorithm_generation.md at main · Project-MONAI/tutorials.” Accessed: Jan. 12, 2024. [Online]. Available: https://github.com/Project-MONAI/tutorials/blob/main/auto3dseg/docs/algorithm_generation.md

[10] Y. He, D. Yang, H. Roth, C. Zhao, and D. Xu, “DiNTS: Differentiable Neural Network Topology Search for 3D Medical Image Segmentation”.

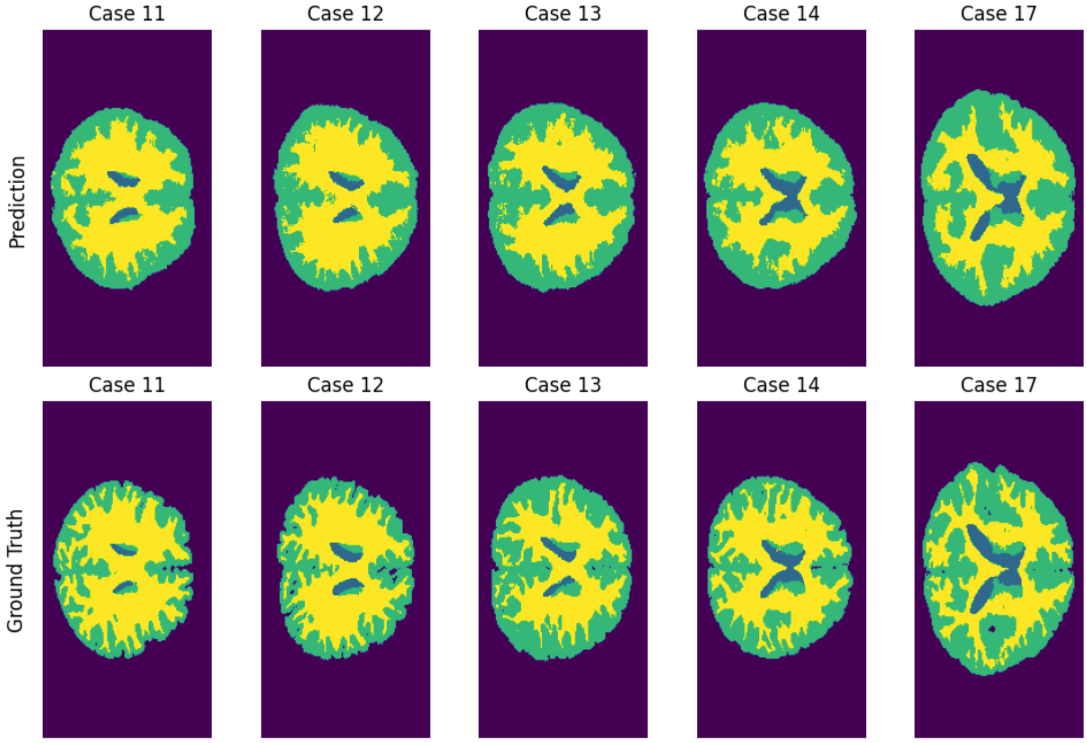
[11] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, “nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation,” *Nature Methods 2020 18:2*, vol. 18, no. 2, pp. 203–211, Dec. 2020, doi: 10.1038/s41592-020-01008-z.

APPENDIX A.

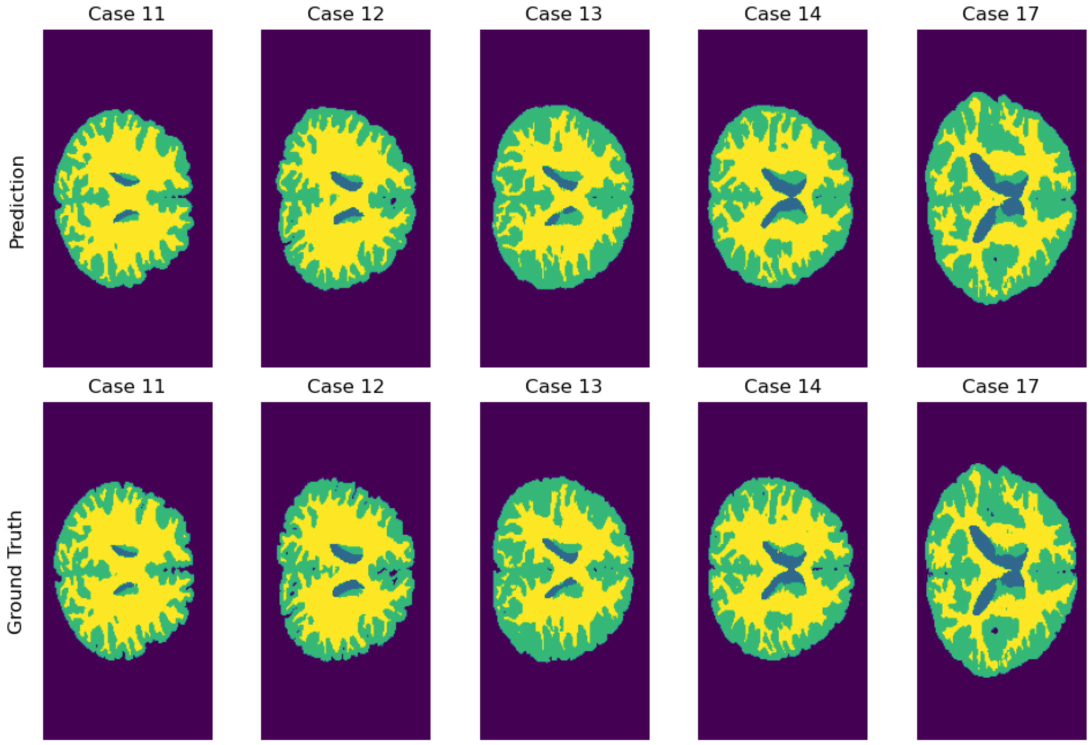
1. **Figure showing the validation images and the corresponding ground truth labels at the slice 150**



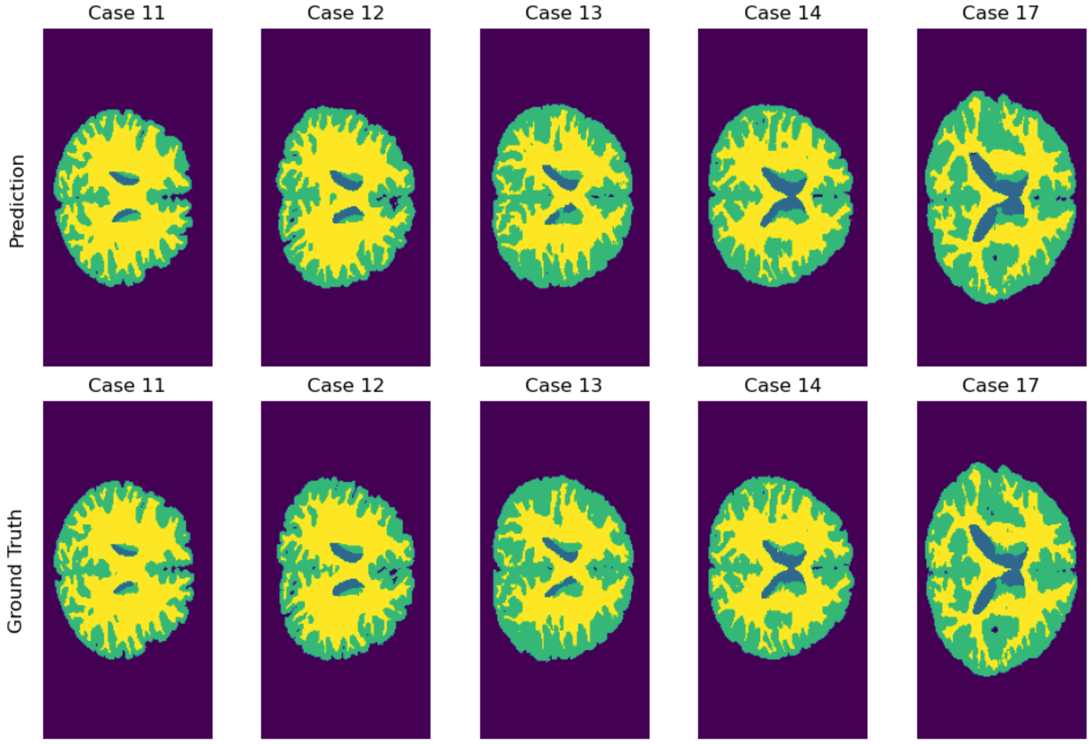
2. **Figure showing the predicted labels using multi-atlas (label fusion with MSE as weight) and the ground truth labels at the slice 150**



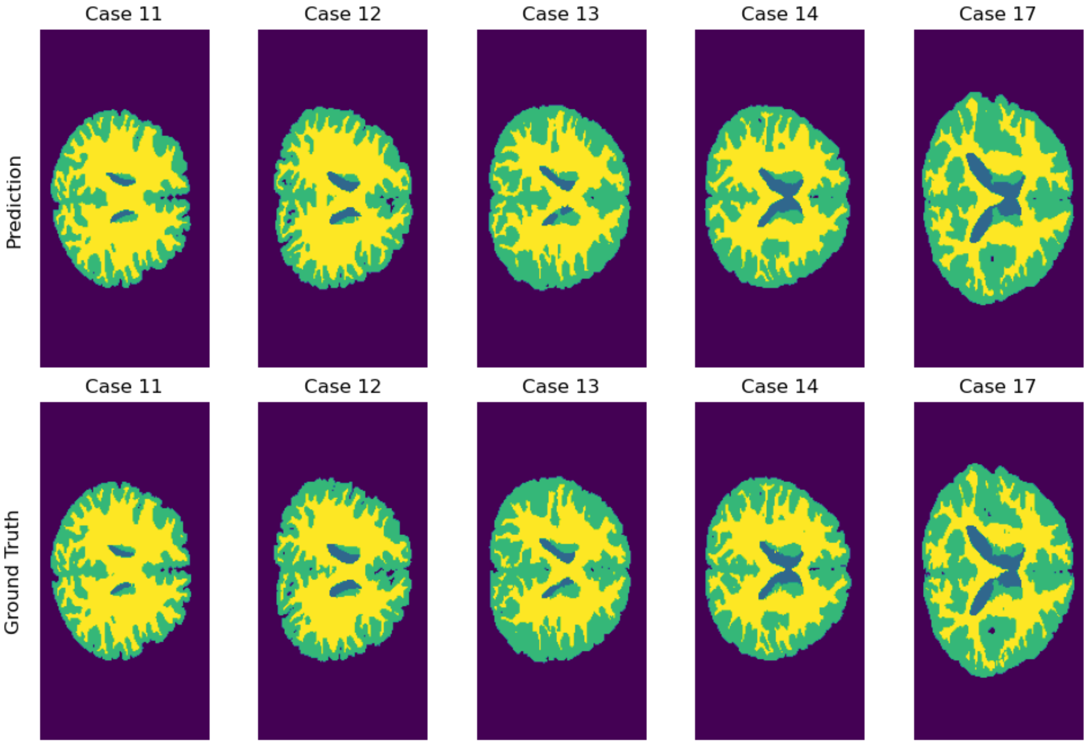
3. **Figure showing the predicted labels using SegResNetDS and the ground truth labels at the slice 150**



4. **Figure showing the predicted labels using SwinUNETR-V2 and the ground truth labels at the slice 150**



5. **Figure showing the predicted labels using Auto3DSeg and the ground truth labels at the slice 150**



6. **Figure showing the predicted labels using nn-Unet and the ground truth labels at the slice 150**