Brain tissue segmentation using Multi-atlas and Deep learning approaches

Muhammad Kabir Hamzah
MSc. in Medical Imaging and Applications
University of Girona
Girona, Spain
u1992383@campus.udg.edu

Faisal Farhan
MSc. in Medical Imaging and Applications
University of Girona
Girona, Spain
u1992382@campus.udg.edu

Diptan Regmi
MSc. in Medical Imaging and Applications
University of Girona
Girona, Spain
u1992786@campus.udg.edu

Abstract

This report presents the development and evaluation of brain tissue segmentation approaches for MRI scans, focusing on cerebrospinal fluid (CSF), grey matter (GM), and white matter (WM). Using the IBSR 18 dataset, multiple segmentation frameworks were implemented, including classical intensity-based methods, atlas-based techniques, and deep learning models. Preprocessing steps addressed heterogeneity in pixel sizes and intensity distributions. Classical approaches such as Expectation-Maximization (EM) algorithms achieved average Dice scores of 0.5586–0.8730 across tissues, with improved performance when incorporating label propagation and anatomical priors. Multi-atlas methods outperformed classical approaches, achieving average Dice scores between 0.8177–0.8541, demonstrating the effectiveness of leveraging prior spatial information. Deep learning models, particularly the 2.5D U-Net, delivered superior performance with average Dice scores ranging from 0.8892–0.9261, surpassing both classical and atlas-based approaches. The ensemble of top-performing models further improved accuracy, achieving the highest average Dice score of 0.9215 across all tissue types. The results highlight the robustness of the 2.5D U-Net for brain tissue segmentation, especially in handling the challenges posed by CSF segmentation due to intensity overlaps. This study underscores the potential of combining deep learning architectures with ensemble strategies for enhanced performance in medical image segmentation tasks.

Keywords: *Image Segmentation, Brain MRI, Atlas, Deep Learning*

1. INTRODUCTION

Brain tissue segmentation is a widely used technique that facilitates the detection of abnormalities such as early tumor diagnosis, multiple sclerosis, Alzheimer's disease, dementia, schizophrenia, and other common diseases [1]. It is a commonly used approach for extracting tissues such as White Matter (WM), Gray Matter (GM), and Cerebrospinal Fluid (CSF) from magnetic resonance (MR) images to enable quantitative brain analysis. It also aids in identifying brain disorders and conducting comprehensive analyses of traumatic brain injuries [2]–[5]. Over the years, many popular brain tissue segmentation methods have been proposed by researchers. These methods have been effectively utilized for disease diagnosis and treatment planning. However, in clinical evaluation and neuroscience research, it remains a significant challenge due to the presence of various artifacts in medical images, including intensity inhomogeneity (IIH), noise, and abnormal tissues with heterogeneous signal intensities. Moreover, the effectiveness of brain tissue segmentation methods is influenced by various factors, including the location, size, shape, texture of tissues, and indistinct tissue boundaries, which are intrinsic to the imaging modalities used for acquisition [6]–[10]. Currently, one of the most widely recognized approaches for enhancing segmentation outcomes in medical imaging involves utilizing prior information from the images through an atlas. An atlas, as the term implies, is a map that represents the anatomical structure of multiple individuals combined into a single image space.

For this project, a subset of the IBSR (Internet Brain Segmentation Repository) database, consisting of brain magnetic resonance imaging (MRI) scans manually segmented by experts, was used. The dataset was divided into 10 images for training, 5 for validation, and 3 for testing. The main goal of this project was to use segmentation techniques to segment the 3 different tissues: CSF, White Matter (WM), and Gray Matter (GM). We implemented and assessed intensity-based, atlas-based, multi-atlas-based, and deep learning (2.5D and 3D) segmentation frameworks for brain MRI tissue segmentation, focusing on White Matter (WM), Gray Matter (GM), and Cerebrospinal Fluid (CSF) for our dataset. The next sections of this report will explain the methodology and the analysis of the results obtained of the Brain tissue segmentation project.

2. MATERIALS AND METHODS

2.1. Dataset description

The MRI dataset used for this project, *IBSR 18*, consists of 18 skull-stripped T1-w MRI images with different spatial resolutions (pixel spacing), intensity ranges but with same number of slices. Moreover, there is also a heterogeneity in image intensities which hinder segmentation. The dataset is provided as follows:

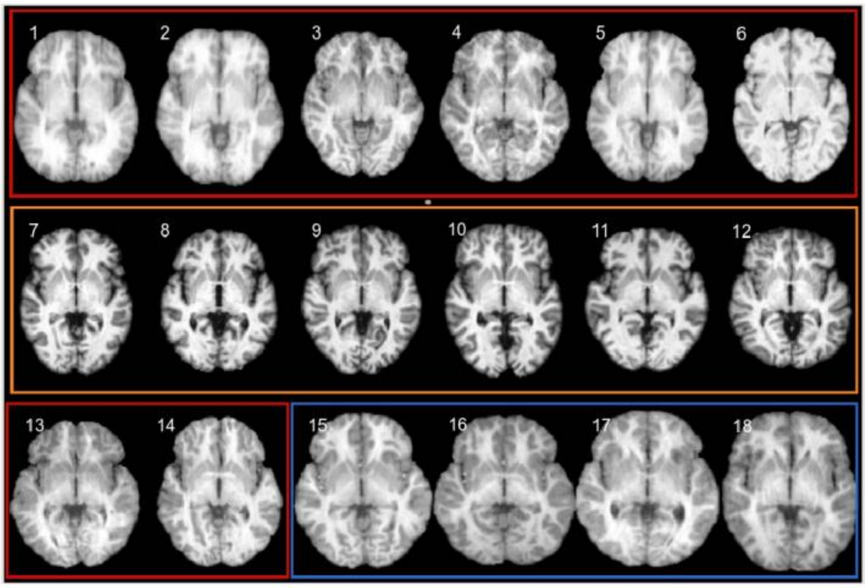


Fig 1. IBSR Data images

The 18 images were split into training, validation and test sets. The tables below summarize the splits as well as each image's pixel information.

Train Set		Validation Set		Test Set	
ID	Pixel Size	ID	Pixel Size	ID	Pixel Size
01	(0.9375, 1.5, 0.9375)	11	(1.0, 1.5, 1.0)	02	(0.9375, 1.5, 0.9375)
03	(0.9375, 1.5, 0.9375)	12	(1.0, 1.5, 1.0)	10	(1.0, 1.5, 1.0)
04	(0.9375, 1.5, 0.9375)	13	(0.9375, 1.5, 0.9375)	15	(0.8371, 1.5, 0.8371)
05	(0.9375, 1.5, 0.9375)	14	(0.9375, 1.5, 0.9375)		
06	(0.9375, 1.5, 0.9375)	17	(0.8371, 1.5, 0.8371)		
07	(1.0, 1.5, 1.0)				
08	(1.0, 1.5, 1.0)				
09	(1.0, 1.5, 1.0)				
16	(0.8371, 1.5, 0.8371)				
18	(0.8371, 1.5, 0.8371)				

Table 1. Dataset description

The observation was that the different image types were present in all sets (stratified split) which is good because it will ensure that any algorithm, we implement will generalize across all the types.

Next, we investigate the intensity distribution across all the set. As a visual example, we present the intensity distribution for all sets in the figures below.

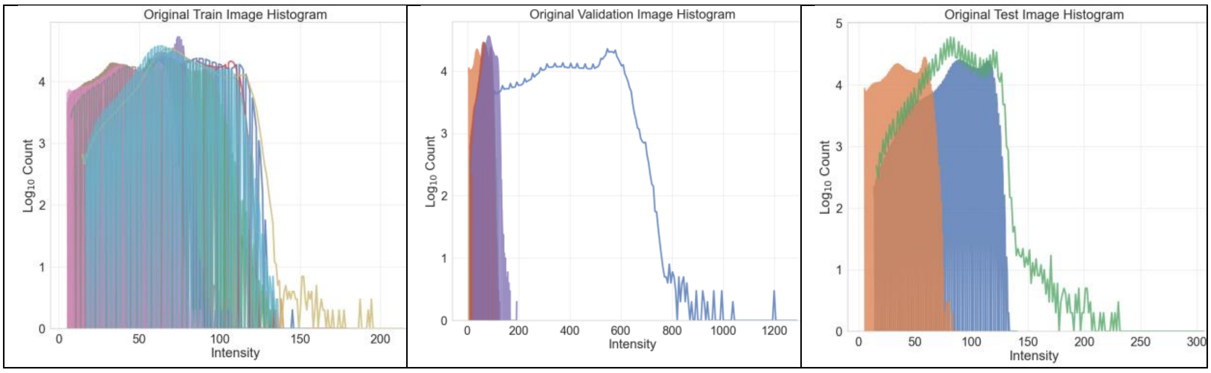


Fig 2. Intensity distribution

We observed the intensity differences across all the sets, and this gave us a quick view of the preprocessing steps to consider.

2.2 Proposal Analysis

Our workflow involves two main steps: preprocessing and segmentation.

2.3 Preprocessing Steps

The usual preprocessing step for any brain tissue segmentation task would involve the following:

a) Skull Stripping
b) Bias-Field Correction
c) Intensity Normalization
d) An optionally, Pixel Size Harmonization.

In this task the first preprocessing step (Skull Stripping) has been done for us, but the dataset still poses two main challenges: different intensity ranges and pixel sizes. Thus, we intend to solve these two challenges in our proposed solution.

2.4 Classic Segmentation Methods

Given our experience during this course with **classical approaches** for brain tissue segmentation, we turned our attention to these methods first.

We decided to reuse most of the previous models we obtained in our lab sessions and previous experiments. As a baseline, we implement:

Intensity-based

a) Tissue Models
b) Expectation-maximization (EM) algorithm with tissue model initialization.
c) EM algorithm with label propagation initialization.
d) EM algorithm with label propagation and tissue models (into and a-posteriori).

Based on our experience, **atlas-based** segmentation usually yields promising results and as required in this task, we implement several **multi-atlas-based** methods as follows:

Multi-atlas

a) Most similar atlas
b) Majority voting or Mean atlas
c) Weighted voting atlas
d) Top atlases

2.5 Deep Learning Segmentation Models

For the deep learning segmentation method, we decided to explore several well-known models that have proven to perform well on brain tissue segmentation and compare their performance with those obtained with our well-known classic algorithms. It was also a good opportunity for us to compare the performance of **3D and 2.5D** networks so each of the models we proposed were trained with 3D and 2.5D inputs.

The proposed deep learning models include:

a) The Basic Unet
b) Unet Transformer
c) ResUnet
d) Swin-Unet

3. Design and implementation of the proposed solutions

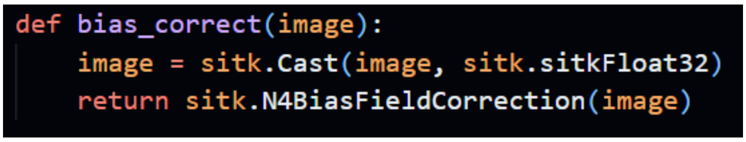
3.1 Preprocessing

The preprocessing methods are discussed as follows.

3.1.1. Bias field Correction

Since the dataset provided was skull stripped, the first part of our preprocessing pipeline was the bias field correction.

We achieved this by using the ***N4BiasFieldCorrection*** filter that comes with ***SimpleITK***. This filter is applied directly on the image and returns a corrected version of the image. A code snippet is provided below.



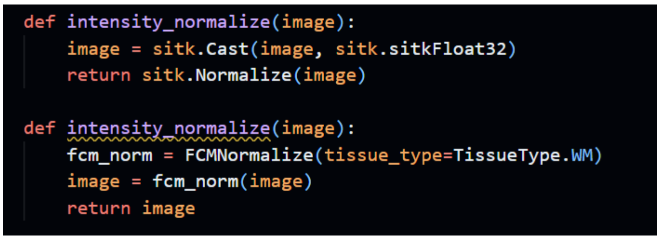
```
def bias_correct(image):
    image = sitk.Cast(image, sitk.sitkFloat32)
    return sitk.N4BiasFieldCorrection(image)
```

Fig 3. Code snippet for Bias Field Correction

Although we applied this for both the baseline methods and the multi-atlas methods we proposed, we did not apply it in our deep learning preprocessing step.

3.1.2. Normalization

We tried two normalization methods for our baseline and multi-atlas methods. One was with ***SimpleITK*** and the other was by a Python library called ***intensity-normalization*** specifically designed for brain MRI imaging normalization. This library contains several normalization methods, but we attempted the Fuzzy C-means (FCM)-based tissue-based mean normalization. We visualized the result of both normalization strategies and based on this result we decided to proceed with the ***SimpleITK*** normalization.



```
def intensity_normalize(image):
    image = sitk.Cast(image, sitk.sitkFloat32)
    return sitk.Normalize(image)

def intensity_normalize(image):
    fcm_norm = FCMNormalize(tissue_type=TissueType.WM)
    image = fcm_norm(image)
    return image
```

Fig 4. Code snippet for intensity normalization

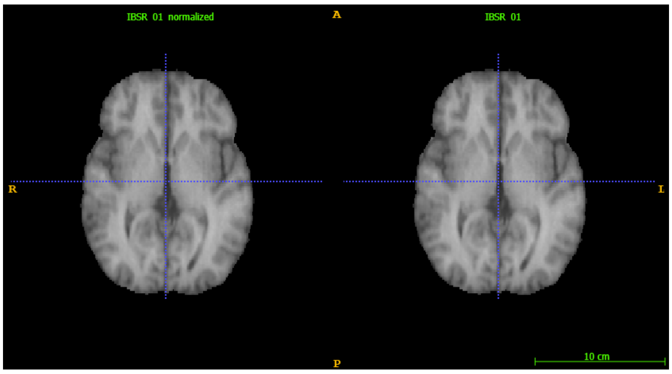


Fig 5. Intensity normalized images(SITK)

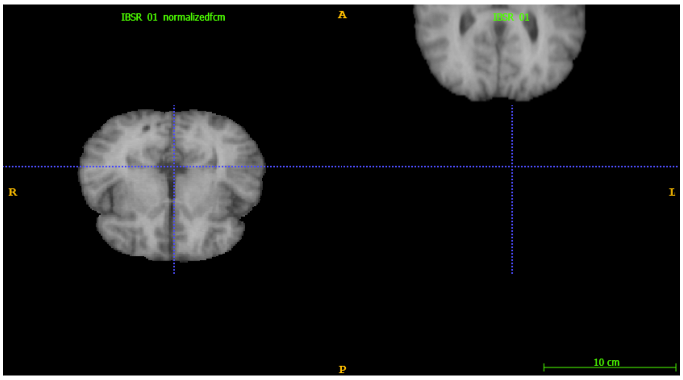
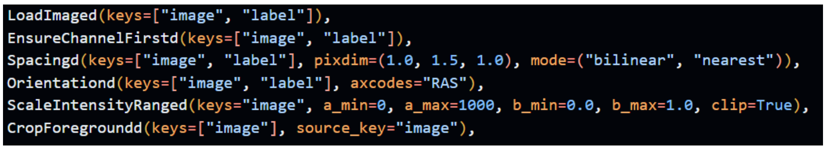


Fig 6. Intensity normalized images(intensity-normalization)

For our deep learning pipeline, we use the ***ScaleIntensityRange*** provided by ***MONAI.***

3.1.3. Pixel Size Harmonization

Here we tried to harmonize the pixel size of our images for all our methods using the ***Spacingd*** provided by ***MONAI*** but we faced multiple bugs in the baseline and multi-atlas approach. Hence the pixel size harmonization was only applied in the deep learning phase. Overall, here is what our deep learning data preprocessing looks like:



```
LoadImaged(keys=["image", "label"]),
EnsureChannelFirstd(keys=["image", "label"]),
Spacingd(keys=["image", "label"], pixdim=(1.0, 1.5, 1.0), mode=("bilinear", "nearest")),
Orientationd(keys=["image", "label"], axcodes="RAS"),
ScaleIntensityRanged(keys="image", a_min=0, a_max=1000, b_min=0.0, b_max=1.0, clip=True),
CropForegroundd(keys=["image"], source_key="image"),
```

Fig 7. Code snippet for deep learning data preprocessing

3.2 Classic Segmentation Approaches

The classic segmentation approaches are discussed in this section.

3.2.1 Intensity-based

Prior to this final project, we had explored several intensity-based segmentation strategies. The knowledge of that was helpful in developing a baseline solution for this task. The implementation of each of these strategies can be found in the ***baselines.ipynb*** notebook.

a) Tissue Models

One way to implement the tissue models was by using the registered images along with the topological atlas but the easiest way and the one which we used was using the original images and their corresponding labels just as we did in the previous lab. For each image and label, the likelihoods for each voxel (p(x|wk)) are computed. These likelihoods represent the conditional probabilities of a voxel's intensity (x) given a tissue class (wk). We created a tissue model from all our training data and used it to segment our validation data to compare the results with individual ground truths provided. An Example of the segmentation result is as shown below.

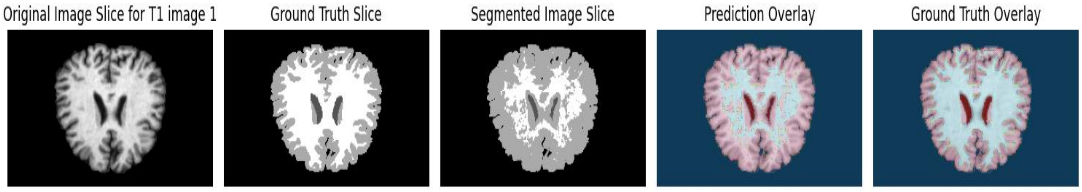


Fig 8. Segmentation results

We observed a decline in performance compared to what we achieved in the previous labs with the same method. This decline could be attributed to several factors but mostly because of the variations in the data.

b) **EM algorithm with tissue model initialization**.

One of the best methods from our previous lab experiment was the initialization of an EM with the tissue model, and thus we attempted the same experiment here. Instead of initializing our EM algorithm with precomputed parameters (mean and variance), we use precomputed tissue model (intensity information) as initial weights for our algorithm to estimate the model parameters. This procedure led to the following results.

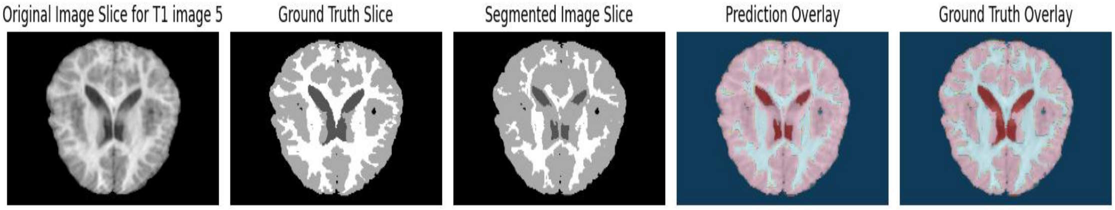


Fig 9. Segmentation results

c) EM algorithm with label propagation initialization.

Next, we attempted label propagation initialization. To do this we had to do several things as summarized below:

1. Determine the most similar atlas.
2. Register all train images to the most similar atlas.
3. Create probabilistic, topological and mean atlas from the train set.
4. Use the obtained atlases to register and create probabilistic atlases for our validation set.
5. Finally, use the probabilistic atlases to initialize the EM for each validation data.

This procedure led to the following results.

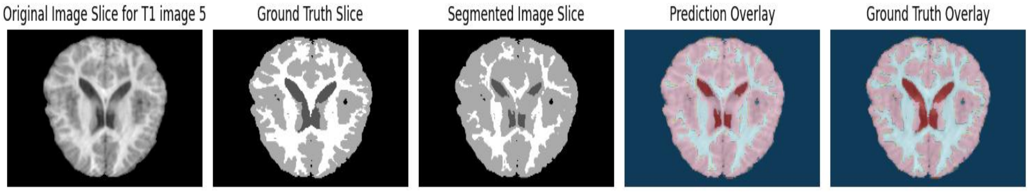
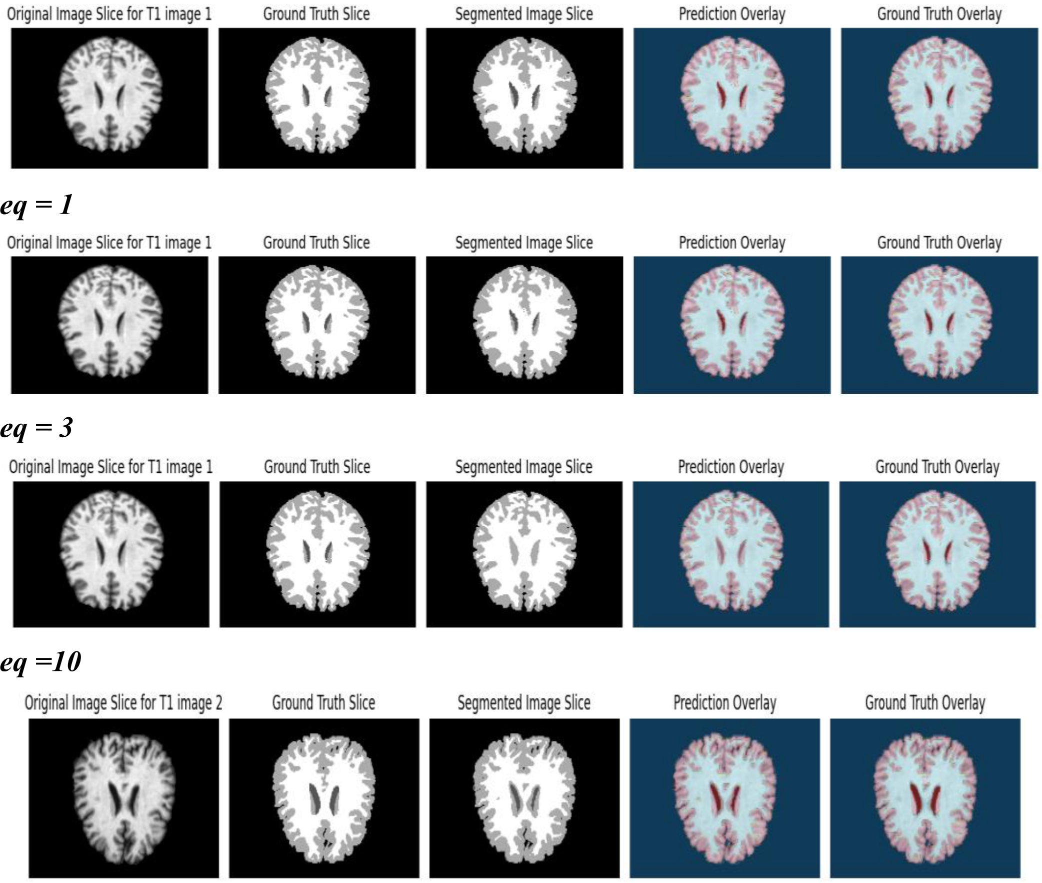


Fig 10. Segmentation results

d) **EM algorithm with label propagation and tissue models (into and a-posteriori).** Our baseline experiment concluded with the combination of tissue models and label propagation as initialization strategies for our EM. Although we did this in several ways. First the aposteriori, then with weight update into the EM with frequencies one, three and ten. We provide an example result of each experiment below.

A posteriori



Freq = 1

Freq = 3

Freq =10

Fig 11. Segmentation results for different frequencies

3.2.2. Multi-Atlas

Multi-atlas segmentation methods leverage prior spatial information from similar images with known segmentation labels (individual atlases) and combine these labels through a label fusion process. The effectiveness of this approach depends on two key factors: the accuracy of the segmentation labels in the individual atlases and the quality of image registration. Before implementing any multi-atlas method, it is necessary to individually register all 10 training images to the target (validation) images.

3.2.3. Image registration

We attempted two methods to register our images. The first was by using our SimpleElastix function and parameters from our previous experiment and the second was with the VoxelMorph library using a pre-trained deep learning model for brain MRI T1-w image registration (*dense-brain-T1-3d-mse-32feat*) found in the VoxelMorph Github repository. By the end of the experiment, we determined that SimpleElastix delivered better results, and we chose to proceed with it. The implementation of each of the strategies discussed below can be found in the ***multi-atlas.ipynb*** notebook.

a) Mean Atlas

This is the simplest label fusion method for multiple atlases. It involves summing the probabilities of all registered individual atlases and dividing the total by the number of atlases. Alternatively, it can be viewed as a weighted voting approach, where all weights are equal, w=1/n, with n being the total number of individual atlases.

b) Most Similar Atlas

When multiple atlases are registered to the same target image (validation image), an easy segmentation approach is to select the atlas with the highest similarity metric. This involves relying solely on the label propagation from the atlas most similar to the target image. For the metric used to measure the similarity, we chose Mattes Mutual Information (MMI). This metric is commonly employed by registration tools such as Elastix. For each validation image, the atlas with the best MMI score was selected, and its propagated labels were used as segmentation masks.

c) Weighted Voting Atlas (global)

This approach assigns different weights to each individual atlas. We used the similarity metric as the weight, as it provides a numerical representation of how similar each individual atlas is to the target image. This ensures greater importance is given to atlases that closely resemble the target image.

d) Weighted Voting Atlas (local)

The previous method assigns weights at a global level i.e. we have one weight associated to each atlas. Here we assign weights at a local level, i.e. to each individual voxel in an atlas. We used the same MMI similarity method to assign these weights.

e) Top Atlas

The analysis of the most similar atlas revealed that certain individual atlases were significantly less similar to our validation images. Including these in the label fusion process, even with weighting, could negatively impact segmentation performance. To address this, we decided to select only the top three most similar individual atlases for fusion. Additionally, we introduced a condition where an atlas would be included in the fusion only if the similarity metric difference was less than 0.05. This ensures that all selected atlases maintain optimal similarity to the target image. Given that the top atlases already exhibit high similarity with the target, we simplified the label fusion process by averaging their labels.

3.2.4. Deep learning Methods

One of the most well-known architectures in the encoder-decoder paradigm is the U-Net model, introduced in 2015. It has become a standard for medical image segmentation due to its effectiveness in capturing both low-level and high-level features. In this project, we variants of the U-Net architecture, including Residual U-Net, UNETR, and SwinUNETR. Each network was trained as a 2.5D and 3D model. The 2.5D and 3D architectures were built using ***MONAI*** and are basically the same with the only difference being in the number of spatial dimensions and the data preprocessing steps. The implementation of both can be found in the ***MISA_DEEPL2.5D.ipynb*** and ***MISA_DEEPL3D.ipynb*** respectively.

Given the limited size of our dataset, we addressed this constraint by training our 3D models using a patch-based approach. This technique involves dividing the input images into smaller, manageable patches, allowing the model to focus on localized features while reducing memory requirements. To further enhance model performance and mitigate overfitting, we incorporated data augmentation techniques. These augmentations introduced variability to the training data by applying transformations such as rotations, flipping, scaling, and intensity variations, ensuring the model's robustness to different image conditions.

3.2.4.1. Basic U-Net

The U-Net architecture is designed with a symmetric encoder-decoder structure and incorporates skip connections to tackle the limitations of standard convolutional neural networks (CNNs) in medical image segmentation. The encoder path extracts hierarchical features through a series of convolutional and max-pooling layers, while the decoder path reconstructs the segmentation mask using transposed convolutions. Skip connections bridge the encoder and decoder paths, ensuring that spatial information from earlier layers is preserved, leading to more accurate segmentation of fine details. Its simplicity and effectiveness make it a widely adopted model in medical imaging tasks.

3.2.4.2. UNETR

The UNETR (U-Net with Transformers) architecture introduces transformers into the U-Net framework, utilizing self-attention mechanisms to capture long-range dependencies within the input data. Unlike the traditional U-Net, which relies solely on convolutional operations, UNETR employs a transformer-based encoder to process the entire image contextually before feeding it into the decoder. This approach is particularly effective for complex medical imaging tasks, as it combines the strengths of both local feature extraction (from the decoder) and global context understanding (from the transformer encoder). The enhanced feature representation enables improved segmentation performance, especially for anatomically intricate structures.

3.2.4.3. Residual U-Net (ResUNet)

The ResUNet architecture extends the standard U-Net by incorporating residual connections, which mitigate the vanishing gradient problem and improve feature propagation during training. Each convolutional block in ResUNet includes skip connections that add the input of the block to its output, allowing the network to learn identity mappings alongside the residual mappings. This design enhances the network's ability to capture complex features while maintaining computational efficiency. ResUNet is particularly suitable for medical image segmentation, as it provides robustness against variations in image intensity and texture.

3.2.4.4. SwinUNETR

The SwinUNETR architecture combines the strengths of U-Net, transformers, and Swin Transformer blocks. SwinUNETR uses a hierarchical transformer-based encoder that employs shifted windows to partition the input into non-overlapping regions, enabling efficient computation and better modeling of long-range dependencies. The architecture leverages these Swin Transformer blocks to enhance feature extraction while maintaining fine-grained spatial information. This makes SwinUNETR highly effective for medical image segmentation tasks, particularly those involving high-resolution images and complex anatomical structures. By integrating the Swin Transformer with the U-Net framework, SwinUNETR achieves superior segmentation accuracy and computational efficiency.

3.2.5. Model Training Parameters

In this section, we discuss the key parameters and hyperparameters used to train our models.

3.2.5.1.Optimizer

An optimizer is a machine learning algorithm used to update a model's parameters based on the training data. The objective is to find the optimal set of parameters that minimize the loss function, which measures the discrepancy between the model's predictions and the true labels.

- **AdamW (Adaptive Moment Estimation with Weight Decay):** This optimizer is an enhanced version of the Adam optimizer, which incorporates weight decay for better regularization. AdamW combines the benefits of adaptive learning rates and momentum, making it robust and efficient for deep learning tasks. It adjusts learning rates individually for each parameter and ensures better generalization by penalizing large weights.

3.2.5.2.LossFunction

A loss function quantifies the error between the predicted output of the model and the actual target output. It guides the optimization process by determining how to update the model's parameters to minimize the error.

DiceCELoss:

This is a composite loss function that combines the Dice Loss and Cross-Entropy Loss. Dice Loss measures the overlap between predicted and true segmentation masks, while Cross-Entropy Loss evaluates the pixel-wise classification accuracy. The combination ensures a balance between optimizing segmentation overlap and classification accuracy, making it particularly suitable for medical image segmentation tasks.

3.2.5.3.Metrics

Metrics are used to evaluate model performance on validation or test sets, allowing for comparison between different models or configurations.

- **DiceSimilarityCoefficient(DSC):**
  The DSC is a statistical metric used to assess the similarity between two sets, typically a predicted segmentation mask and a ground-truth mask. It is calculated as:
  $$DSC(A,B) = 2(A \cap B)/(A + B)$$
  where A is the predicted mask and B is the ground truth. A higher DSC indicates better segmentation accuracy.

Model	**2.5D**	**3D**
Optimizer	AdamW	AdamW
Loss	DiceCE loss	DiceCE loss
Metric	Dice Similarity Coefficient	Dice Similarity Coefficient
Number of Epochs	150	100
Batch Size	64-128	4
Patch size	-	(96,96,96)
Early Stopping (Patience)	15	15

Table 2. Summary of training parameters and hyperparameters

These parameters and hyperparameters were carefully selected to optimize the model's performance while balancing computational efficiency.

4. Experimental section and results analysis

The results demonstrate varying performance across different segmentation frameworks, emphasizing the strengths and weaknesses of classical and modern techniques.

4.1. CSF Segmentation

We observed that segmenting the CSF is the most challenging task among the three brain tissues. The primary difficulty arises because the intensity range of CSF is very narrow and overlaps considerably with that of GM, making it hard for intensity-based models to distinguish between the two. From our analysis, we noticed that the tissue models, and the two initialization strategies of our EM methods, had severely failed to segment some of the CSF effectively. This likely stems from low variability between CSF and GM intensities, causing the algorithm to struggle with distinguishing these tissues as its convergence depends on the distinctiveness of the intensity distributions. When we introduced a single atlas into the segmentation process, we observed a significant improvement in the results, with the most significant improvement achieved when we introduced the atlas (Label prop + Tissue model) into the EM with frequency equals one.

In the case of the multi-atlas strategies, all methods evaluated significantly outperformed our baselines and the results were almost identical in terms of the mean dice scores.

In the case of the deep learning models, the first observation was that both the 2.5D and 3D models outperformed the mutli-atlas strategies. Which is as expected, but what we didn't expect was that the basic UNET will outperform the sophisticated models we implemented such as UNETR or SwinUNETR. Another observation that we didn't expect was that the 2.5D models would outperform the 3D models. Although the performance of the sophisticated models as well as the 3D models in general, could be attributed to the limitation of the datasets.

Overall, the 2.5D Basic UNET model proved to be the most effective at segmenting the CSF.

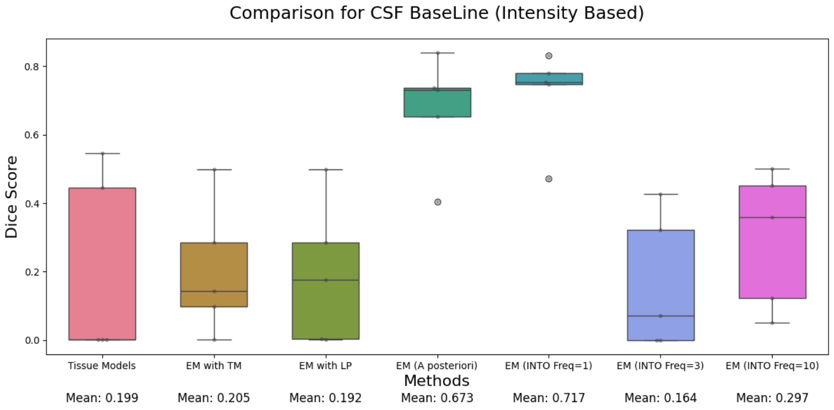
4.2. GM and WM Segmentation

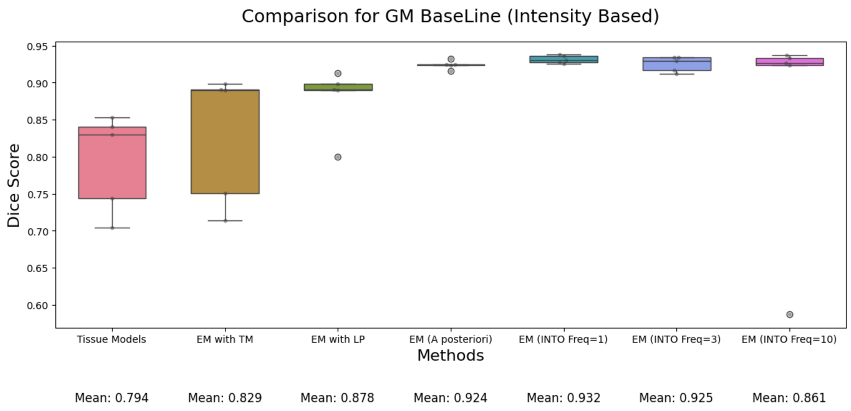
In contrast to the challenges faced with CSF segmentation, most of the methods we experimented with produced very good segmentation results for both grey matter (GM) and white matter (WM), with the only exception being the WM segmentation with tissue models.

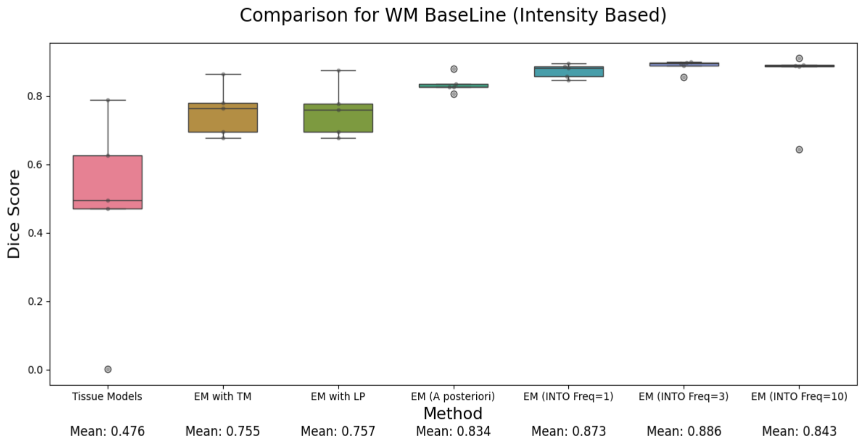
For our baseline approaches, again the atlas (Label prop + Tissue model) into the EM with frequency equals one outperforms every other method in its class. And for the multi-atlas strategies, all methods evaluated were almost identical in terms of the mean dice scores.

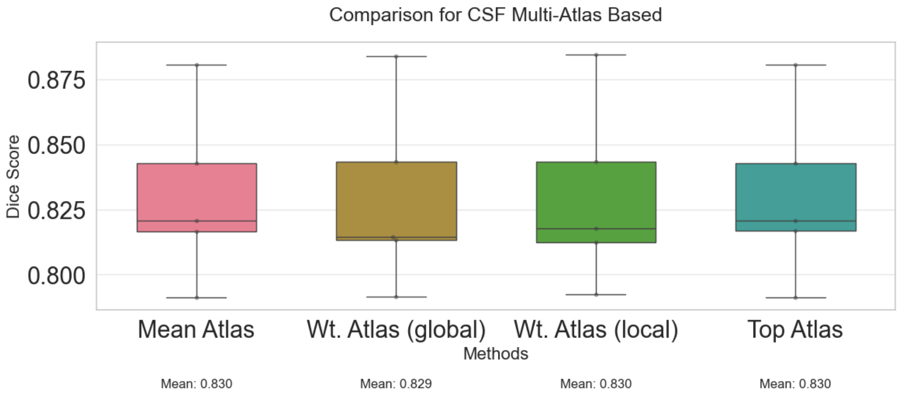
Interestingly, our 2.5D UNET model consistently outperformed other methods for the segmentation of GM and WM, and it was also the best-performing model in the CSF segmentation. This consistency across all three tissues indicates that the 2.5D UNET model is the most robust method we developed. Therefore, we can conclude that it is the most effective segmentation method among those we implemented, offering reliable performance across all main brain tissues.

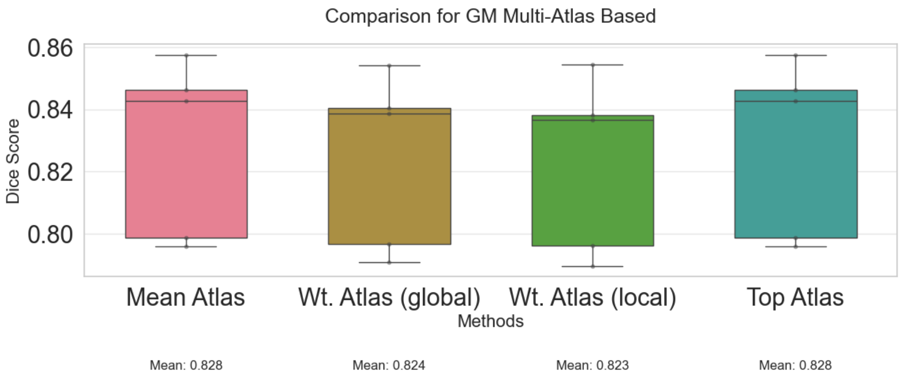
Lastly, we ensemble the results of the top three deep learning models in an effort to boost the overall performance and generalization of our models. We present the result of the ensemble in table 22.

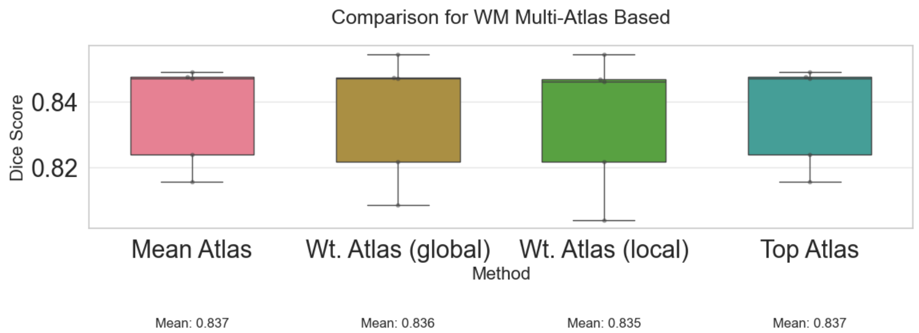


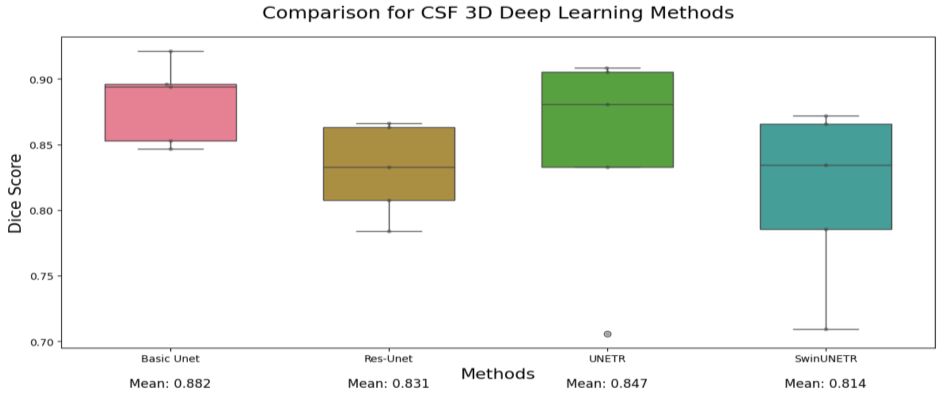

Comparison for GM BaseLine (Intensity Based)
Dice Score
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
Tissue Models
EM with TM
EM with LP
EM (A posteriori)
EM (INTO Freq=1)
EM (INTO Freq=3)
EM (INTO Freq=10)
Methods
Mean: 0.794
Mean: 0.829
Mean: 0.878
Mean: 0.924
Mean: 0.932
Mean: 0.925
Mean: 0.861

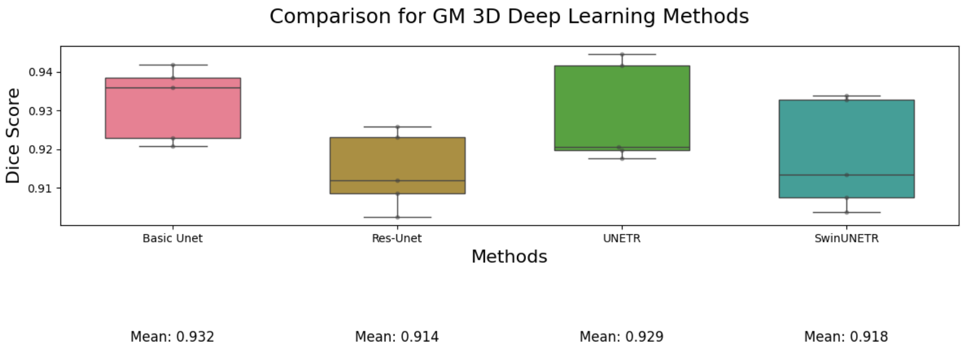

Comparison for WM BaseLine (Intensity Based)
Dice Score
0.8
0.6
0.4
0.2
0.0
Tissue Models
EM with TM
EM with LP
EM (A posteriori)
EM (INTO Freq=1)
EM (INTO Freq=3)
EM (INTO Freq=10)
Method
Mean: 0.476
Mean: 0.755
Mean: 0.757
Mean: 0.834
Mean: 0.873
Mean: 0.886
Mean: 0.843

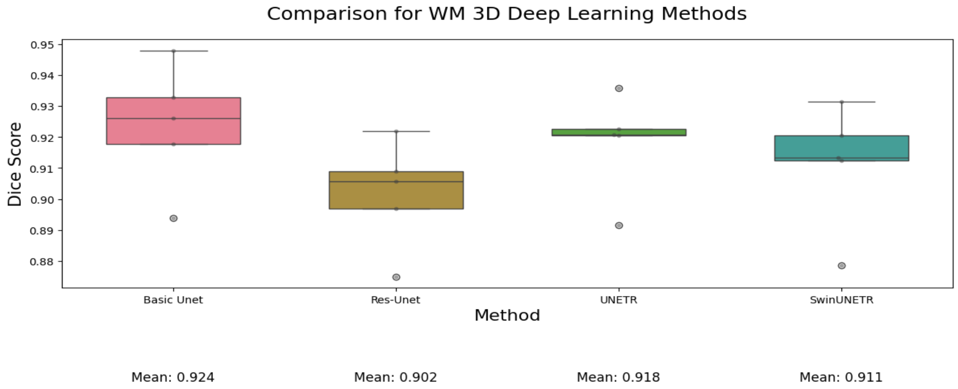

Comparison for CSF Multi-Atlas Based
Dice Score
0.875
0.850
0.825
0.800
Mean Atlas
Wt. Atlas (global)
Wt. Atlas (local)
Top Atlas
Methods
Mean: 0.830
Mean: 0.829
Mean: 0.830
Mean: 0.830


Comparison for GM Multi-Atlas Based
Dice Score
0.86
0.84
0.82
0.80
Mean Atlas
Wt. Atlas (global)
Wt. Atlas (local)
Top Atlas
Methods
Mean: 0.828
Mean: 0.824
Mean: 0.823
Mean: 0.828


Comparison for WM Multi-Atlas Based
Dice Score
0.84
0.82
Mean Atlas
Wt. Atlas (global)
Wt. Atlas (local)
Top Atlas
Method
Mean: 0.837
Mean: 0.836
Mean: 0.835
Mean: 0.837


Comparison for CSF 3D Deep Learning Methods
Dice Score
0.90
0.85
0.80
0.75
0.70
Basic Unet
Res-Unet
UNETR
SwinUNETR
Methods
Mean: 0.882
Mean: 0.831
Mean: 0.847
Mean: 0.814


Comparison for GM 3D Deep Learning Methods
Dice Score
0.94
0.93
0.92
0.91
Basic Unet
Res-Unet
UNETR
SwinUNETR
Methods
Mean: 0.932
Mean: 0.914
Mean: 0.929
Mean: 0.918


Comparison for WM 3D Deep Learning Methods
Dice Score
0.95
0.94
0.93
0.92
0.91
0.90
0.89
0.88
Basic Unet
Res-Unet
UNETR
SwinUNETR
Method
Mean: 0.924
Mean: 0.902
Mean: 0.918
Mean: 0.911

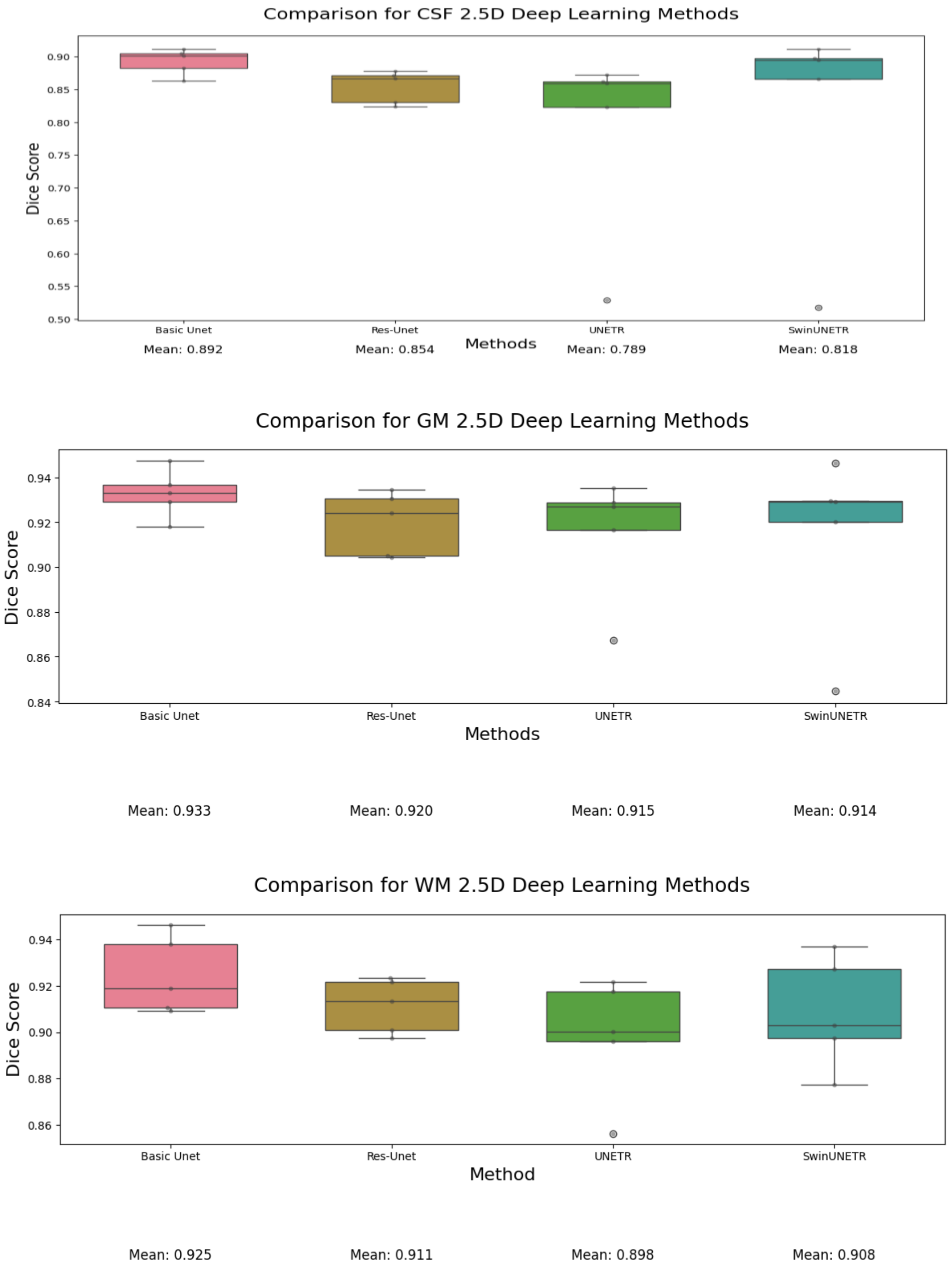


Fig 12. Comparison of plots for different methods

We provide one table each summarizing the results obtained with each method.

4.2.3. Result for Baseline Models

- Tissue Models

Image	**CSF**	**GM**	**WM**	**Avg**
IBSR_11	0.5460	0.7434	0.0014	0.4303
IBSR_12	0.4454	0.8400	0.4955	0.5936
IBSR_13	0.0019	0.7043	0.6253	0.4438
IBSR_14	0.00014	0.8294	0.7883	0.5392
IBSR_17	0.0016	0.8533	0.4694	0.4415

Table 3. Result for Tissue Models

- EM algorithm with tissue model initialization

Image	CSF	GM	WM	Avg
IBSR_11	0.0986	0.7134	0.8637	0.5586
IBSR_12	0.1429	0.7505	0.7641	0.5525
IBSR_13	0.00054	0.8909	0.6763	0.5226
IBSR_14	0.2850	0.8986	0.7786	0.6541
IBSR_17	0.4980	0.8897	0.6947	0.6941

Table 4. Result for EM algorithm with tissue model initialization

- EM algorithm with label propagation initialization

Image	CSF	GM	WM	Avg
IBSR_11	0.0025	0.9125	0.8748	0.5966
IBSR_12	0.1752	0.7995	0.7602	0.5783
IBSR_13	0.0005	0.8908	0.6759	0.5224
IBSR_14	0.2850	0.8983	0.7775	0.6536
IBSR_17	0.4980	0.8897	0.6947	0.6941

Table 5. Result for EM algorithm with label propagation initialization

- EM algorithm with label propagation and tissue models (a-posteriori)

Image	CSF	GM	WM	Avg
IBSR_11	0.7378	0.9246	0.8782	0.8468
IBSR_12	0.6532	0.9161	0.8265	0.7986
IBSR_13	0.4044	0.9321	0.8250	0.7205
IBSR_14	0.7310	0.9239	0.8351	0.8300
IBSR_17	0.8398	0.9242	0.8067	0.8569

Table 6. Result for EM algorithm with label propagation and tissue models

- EM algorithm with label propagation and tissue models (into freq =1)

Image	CSF	GM	WM	Avg
IBSR_11	0.7466	0.9275	0.8952	0.8564
IBSR_12	0.7809	0.9364	0.8814	0.8662
IBSR_13	0.4725	0.9256	0.8449	0.7477
IBSR_14	0.7543	0.9380	0.8860	0.8594
IBSR_17	0.8328	0.9302	0.8560	0.8730

Table 7. Result for EM algorithm with label propagation and tissue models

- EM algorithm with label propagation and tissue models (into freq =3)

Image	CSF	GM	WM	Avg
IBSR_11	0.0	0.9121	0.8964	0.6028
IBSR_12	0.0	0.9297	0.8943	0.6080
IBSR_13	0.0714	0.9164	0.8545	0.6141
IBSR_14	0.3225	0.9338	0.9000	0.7187
IBSR_17	0.4258	0.9342	0.8871	0.7490

Table 8. Result for EM algorithm with label propagation and tissue models

- EM algorithm with label propagation and tissue models (into freq =10)

Image	CSF	GM	WM	Avg
IBSR_11	0.0509	0.9267	0.91025	0.6293
IBSR_12	0.3589	0.9237	0.8864	0.7230
IBSR_13	0.1219	0.5869	0.6426	0.4505
IBSR_14	0.4519	0.9333	0.8896	0.7583
IBSR_17	0.5013	0.9368	0.8881	0.7754

Table 9. Result for EM algorithm with label propagation and tissue models

4.3. Results for Multi-Atlas

- Mean Atlas

Image	CSF	GM	WM	Avg
IBSR_11	0.8166	0.7986	0.8474	0.8209
IBSR_12	0.8207	0.7959	0.8469	0.8212
IBSR_13	0.7912	0.8462	0.8157	0.8177
IBSR_14	0.8428	0.8427	0.8488	0.8447
IBSR_17	0.8807	0.8575	0.8239	0.8541

Table 10. Result for Mean Atlas

- Weighted Atlas (global)

Image	CSF	GM	WM	Avg
IBSR_11	0.8145	0.7967	0.8543	0.8219
IBSR_12	0.8133	0.7909	0.8469	0.8170
IBSR_13	0.7914	0.8405	0.8088	0.8136
IBSR_14	0.8434	0.8386	0.8472	0.8431
IBSR_17	0.8839	0.8542	0.8219	0.8534

Table 11. Result for Weighted Atlas

- Weighted Atlas (local)

Image	CSF	GM	WM	Avg
IBSR_11	0.8177	0.7961	0.8541	0.8226
IBSR_12	0.8124	0.7896	0.8461	0.8161
IBSR_13	0.7923	0.8366	0.8042	0.8111
IBSR_14	0.8435	0.8381	0.8468	0.8428
IBSR_17	0.8847	0.8543	0.8219	0.8537

Table 12. Result for Weighted Atlas

- Top Atlas

Image	CSF	GM	WM	Avg
IBSR_11	0.8167	0.7987	0.8474	0.8209
IBSR_12	0.8207	0.7959	0.8469	0.8212
IBSR_13	0.7912	0.8463	0.8157	0.8177
IBSR_14	0.8428	0.8427	0.8488	0.8448
IBSR_17	0.8808	0.8575	0.8239	0.8540

Table 13. Result for Top Atlas

4.4 Results for Deep Learning

- Basic-Unet

Image	CSF	GM	WM	Avg
IBSR_11	0.8821	0.9291	0.9464	0.9192
IBSR_12	0.9007	0.9180	0.9191	0.9126
IBSR_13	0.8630	0.9329	0.9094	0.9018
IBSR_14	0.9049	0.9473	0.9381	0.9301
IBSR_17	0.9112	0.9366	0.9107	0.9195

Table 14. 2.5D Model Result

Image	CSF	GM	WM	Avg
IBSR_11	0.8467	0.9360	0.9479	0.9102
IBSR_12	0.8960	0.9228	0.9261	0.9261
IBSR_13	0.8528	0.9207	0.8940	0.8892
IBSR_14	0.8942	0.9419	0.9329	0.9230
IBSR_17	0.9214	0.9384	0.9178	0.9259

Table 15. 3D Model Result

- UNETR

Image	CSF	GM	WM	Avg
IBSR_11	0.5287	0.8672	0.8564	0.7508
IBSR_12	0.8727	0.9167	0.9216	0.9037
IBSR_13	0.8229	0.9269	0.9002	0.8833
IBSR_14	0.8588	0.9351	0.9177	0.9038
IBSR_17	0.8616	0.9289	0.8962	0.8956

Table 16. 2.5D Model Result

Image	CSF	GM	WM	Avg
IBSR_11	0.7059	0.9176	0.9226	0.8487
IBSR_12	0.8806	0.9198	0.9206	0.9071
IBSR_13	0.8331	0.9206	0.8916	0.8817
IBSR_14	0.9055	0.9446	0.9358	0.9286
IBSR_17	0.9083	0.9417	0.9207	0.9236

Table 17. 3D Model Result

- ResUnet

Image	CSF	GM	WM	Avg
IBSR_11	0.8242	0.9043	0.9236	0.8840
IBSR_12	0.8669	0.9050	0.9134	0.8951
IBSR_13	0.8301	0.9240	0.8975	0.8838
IBSR_14	0.8775	0.9346	0.9218	0.9114
IBSR_17	0.8717	0.9305	0.9010	0.9011

Table 18. 2.5D Model Result

Image	CSF	GM	WM	Avg
IBSR_11	0.8077	0.9120	0.9220	0.8806
IBSR_12	0.8631	0.9024	0.9056	0.8904
IBSR_13	0.7840	0.9085	0.8750	0.8558
IBSR_14	0.8327	0.9231	0.9089	0.8882
IBSR_17	0.8663	0.9259	0.8969	0.8964

Table 19. 3D Model Result

- SwinUNETR

Image	CSF	GM	WM	Avg
IBSR_11	0.5177	0.8446	0.8773	0.7465
IBSR_12	0.8950	0.9201	0.9274	0.9142
IBSR_13	0.8656	0.9292	0.9030	0.8993
IBSR_14	0.9118	0.9462	0.9369	0.9316
IBSR_17	0.8978	0.9294	0.8976	0.9083

Table 20. 2.5D Model Result

Image	CSF	GM	WM	Avg
IBSR_11	0.7095	0.9038	0.9133	0.8422
IBSR_12	0.8721	0.9135	0.9206	0.9021
IBSR_13	0.7856	0.9075	0.8787	0.8573
IBSR_14	0.8343	0.9339	0.9313	0.8998
IBSR_17	0.8660	0.9327	0.9124	0.9037

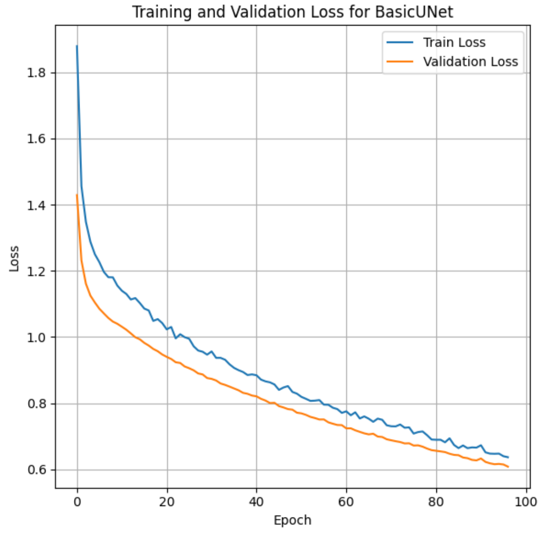
Table 21. 3D Model Result

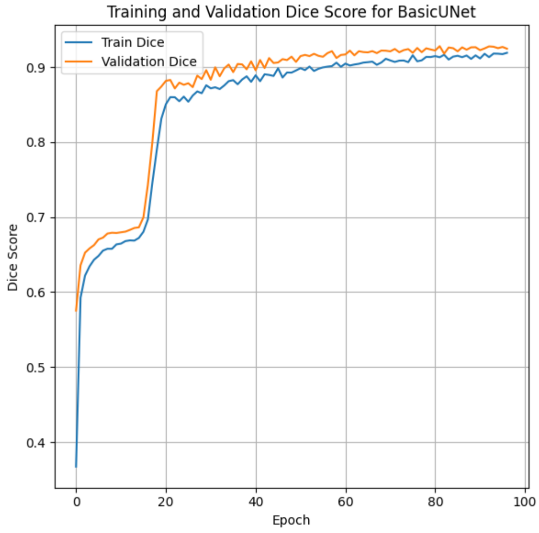
- Ensemble Model

Image	CSF	GM	WM	Avg
IBSR_11	0.8780	0.9355	0.9513	0.9217
IBSR_12	0.9034	0.9250	0.9270	0.9184
IBSR_13	0.8670	0.9327	0.9098	0.9032
IBSR_14	0.9100	0.9504	0.9428	0.9344
IBSR_17	0.9215	0.9437	0.9239	0.9297
MEAN	**0.8960**	**0.9375**	**0.9309**	**0.9215**

Table 22. Ensemble of top 3 models (2 * 2.5D and 3D Basic UNETs)

The training curves for both 2.5D and 3D models are illustrated in the figures below.





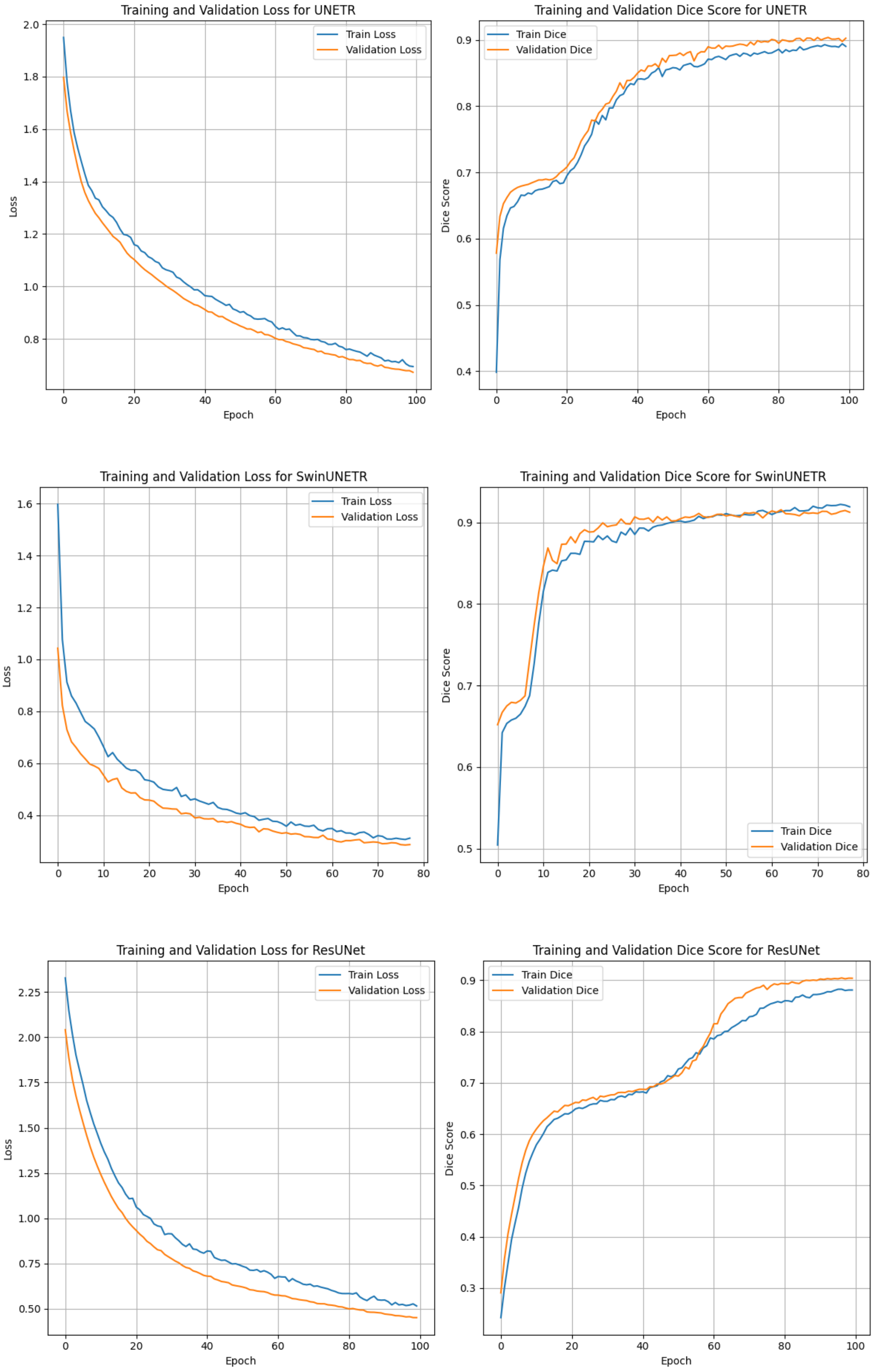
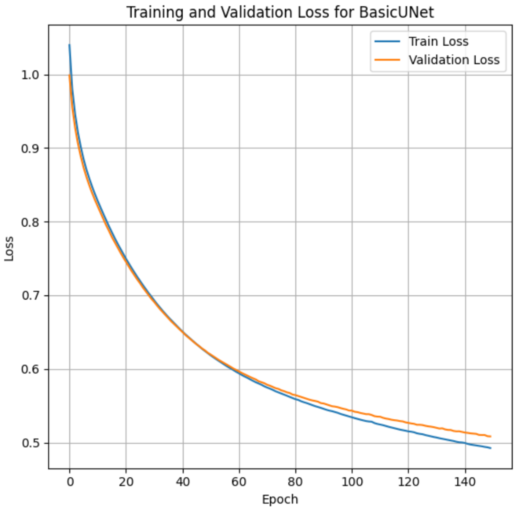
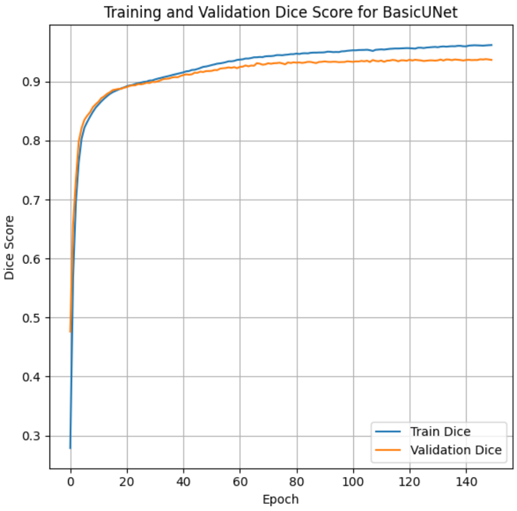
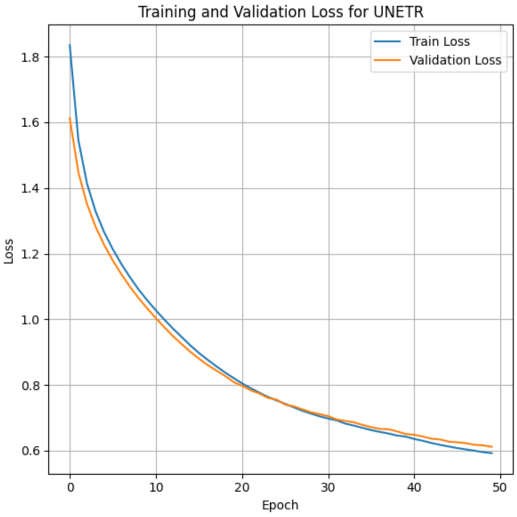
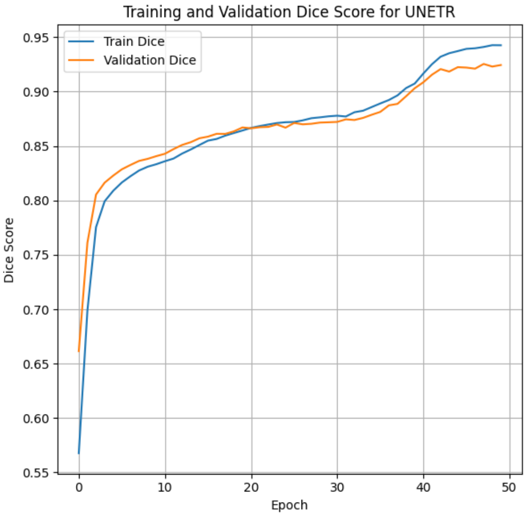


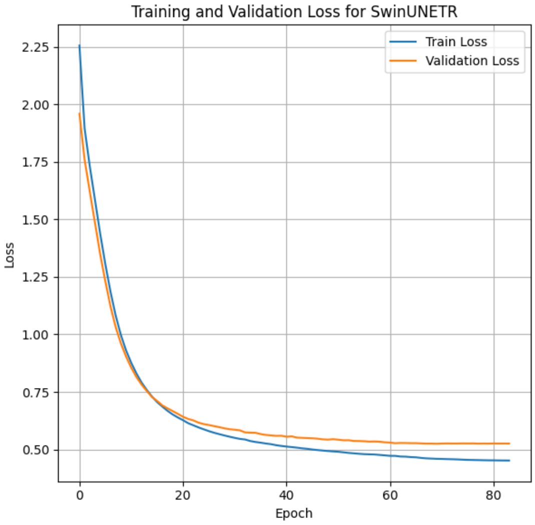
Fig 13. Training curves for 2.5D Models

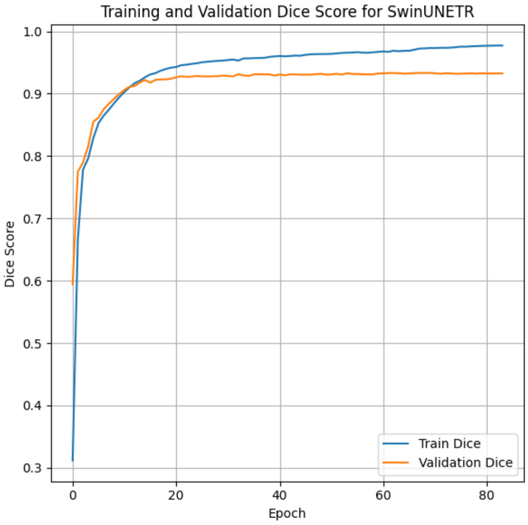

Training and Validation Loss for BasicUNet
Train Loss
Validation Loss
Loss
1.0
0.9
0.8
0.7
0.6
0.5
0
20
40
60
80
100
120
140
Epoch


Training and Validation Dice Score for BasicUNet
Train Dice
Validation Dice
Dice Score
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
20
40
60
80
100
120
140
Epoch


Training and Validation Loss for UNETR
Train Loss
Validation Loss
Loss
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0
10
20
30
40
50
Epoch


Training and Validation Dice Score for UNETR
Train Dice
Validation Dice
Dice Score
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0
10
20
30
40
50
Epoch


Training and Validation Loss for SwinUNETR
Train Loss
Validation Loss
Loss
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0
20
40
60
80
Epoch


Training and Validation Dice Score for SwinUNETR
Train Dice
Validation Dice
Dice Score
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
20
40
60
80
Epoch

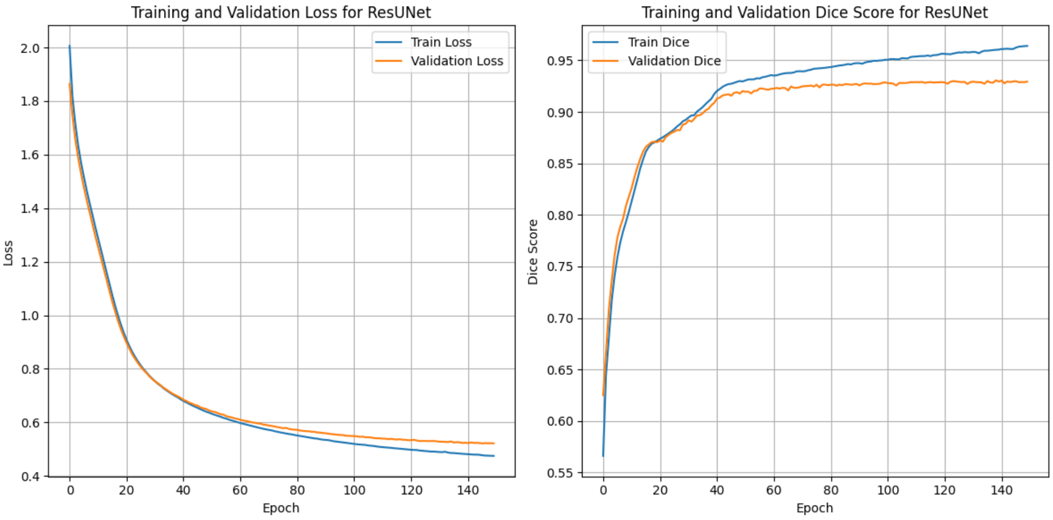


Fig 14. Training curves for 3D models

5. Conclusion

In this experiment, we successfully developed and evaluated intensity based, atlas-based, multi-atlas based and deep learning (2.5D and 3D) segmentation framework for brain MRI tissue segmentation, focusing on white matter (WM), grey matter (GM), and cerebrospinal fluid (CSF). Our findings highlight the complexity of CSF segmentation due to intensity overlap with GM, making it challenging for intensity-based models alone to differentiate the two tissues effectively. By incorporating our atlas with both label propagation and an a posteriori approach, we significantly enhanced CSF segmentation results, underscoring the value of incorporating anatomical priors. Moreover, our 2.5D deep learning approach consistently performed well across all tissue types, outperforming other methods and demonstrating robust results in segmenting WM, GM, and CSF. Overall, our 2.5D UNET model has proven to be the most effective segmentation method across the main brain tissues, making it a reliable tool for medical image segmentation tasks.

References

[1] Dora L, Agrawal S, Panda R, Abraham A. State-of-the-Art Methods for Brain Tissue Segmentation: A Review. IEEE Rev Biomed Eng. 2017;10:235-249. doi: 10.1109/RBME.2017.2715350. Epub 2017 Jun 14. PMID: 28622675.

[2] M. M. Eapena, M. S. J. A. Ancelita, and G. Geetha, “Segmentation of tumors from ultrasound images with PAORGB,” Procedia Comput. Sci.,vol. 50, pp. 663–668, 2015.

[3] L. Yang, O. Tuzel, P. Meer, and D. J. Foran, “Automatic image analysis of histopathology specimens using concave vertex graph,” in Proc.11th Int.Conf.Med. Image Comput.Comput.-Assisted Intervention, 2008,pp. 833–841.

[4] B. Foster, U. Bagci, A. Mansoor, Z. Xu, and D. J. Mollura, “A review on segmentation of positron emission tomography images,” Comput. Biol. Med., vol. 50, pp. 76–96, 2014.

[5] S. K. Adhikari, J. K. Sing, D. K. Basu, and M. Nasipuri, “Conditional spatial fuzzy C-means clustering algorithm for segmentation of MRI images,” Appl. Soft Comput., vol. 34, pp. 758–769, 2015.

[6] C. Chen, W. Xie, J. Franke, P. A. Grutzner, L. P. Nolte, and G. Zheng, "Automatic X-ray landmark detection and shape segmentation via datadriven joint estimation of image displacements," Med. Image Anal.,vol. 18, pp. 487–499, 2014.

[7] Y.Artan,A.Oto, and I. S.Yetik, "Cross-device automated prostate cancer localization with multiparametric MRI," IEEE Trans. Image Process.,vol. 12, pp. 5385–5394, Dec. 2013.

[8] Y.Artan,A.Oto, and I. S.Yetik, "Cross-device automated prostate cancer localization with multiparametric MRI," IEEE Trans. Image Process.,vol. 12, pp. 5385–5394, Dec. 2013.

[9] S. Liao and D. Shen, "A feature-based learning framework for accurate prostate localization in CT images," IEEE Trans. Image Process., vol. 21, no. 8, pp. 3546–3559, Aug. 2012.

[10] L.Wen, X.Wang, Z.Wu,M. Zhou, and J. S. Jin, "A novel statistical cerebrovascular segmentation algorithm with particle swarm optimization," Neurocomputing, vol. 148, pp. 569–577, 2015.