Medical image segmentation and applications (MISA) final report

Brain tissue segmentation using Multi-atlas and Deep learning approaches

Members: Chicano, Àlex (MIC); Montoya, Ricardo (MAIA); Yalçın, Cansu (MAIA)

Abstract: Medical image segmentation is a widely used technique for both clinical and research based applications. In this project, we performed segmentation of the three main brain tissues cerebrospinal fluid (CSF), gray matter (GM) and white matter (WM), from the brain MRI dataset of IBSR 18. We followed two main pipelines; classical approaches containing multi-atlas, bayesian model and intensity based models as well as deep learning models containing U-net, DenseUnet and ResUnet. Overall, we performed experiments to find the optimized setting for our segmentation problem. Among the mean of the validation dataset, we were able to obtain **0.80, 0.88, 0.88** dice scores, using the Bayesian model, and **0.84, 0.93 and 0.91** dice scores using the DenseUnet model for the CSF, GM and WM sequentially.

1. Introduction and problem definition

Due to the significant improvement in diagnostic efficiency and accuracy, medical image segmentation frequently plays a crucial part in computer assisted diagnosis and smart medicine. Medical image segmentation seeks to make anatomical or pathological structural alterations in images more obvious. Popular medical image segmentation tasks include liver and liver-tumor segmentation, brain and brain-tumor segmentation, optic disc segmentation, etc. In this project we focused on the brain tissue segmentation.

Brain tissue segmentation is one of the most sought after research areas in medical image processing. It provides detailed quantitative brain analysis for accurate disease diagnosis, detection, and classification of abnormalities [1]. Among all brain image modalities, magnetic resonance imaging (MRI) is the most common for tissue segmentation given its importance in clinical applications and also in research.

Automatic tissue segmentation is a challenging process because of the presence of intensity inhomogeneity, noise and the complex anatomical structure of the tissues of interest. The goal of our project is to automatically segment the three main brain tissues: cerebrospinal fluid (CSF), gray matter (GM) and white matter (WM).

1.1 Dataset description

The MRI dataset used for this project, *IBSR 18*, consists of 18 skull-stripped T1-w MRI images, derived from healthy subjects [2]. The dataset is provided as follows:

Dataset split	**Number of images**	**Ground truth labels provided?**
Training	10	Yes
Validation	5	Yes
Test	3	No

These images also have different spatial resolutions and intensity distributions. Although all images have the same number of slices (256, 128, 256), the pixel size per dimension changed among images. In general, three different pixel sizes were present in the dataset and in the splits. The table below summarizes the pixel size information.

Training

ID	Pixel size	Type
01	(0.9375, 1.5, 0.9375)	A
03	(0.9375, 1.5, 0.9375)	A
04	(0.9375, 1.5, 0.9375)	A
04	(0.9375, 1.5, 0.9375)	A
06	(0.9375, 1.5, 0.9375)	A
07	(1.0, 1.5, 1.0)	B
08	(1.0, 1.5, 1.0)	B
09	(1.0, 1.5, 1.0)	B
16	(0.8371, 1.5, 0.8371)	C
18	(0.8371, 1.5, 0.8371)	C

Validation

ID	Pixel size	Type
11	(1.0, 1.5, 1.0)	B
12	(1.0, 1.5, 1.0)	B
13	(0.9375, 1.5, 0.9375)	A
14	(0.9375, 1.5, 0.9375)	A
17	(0.8371, 1.5, 0.8371)	C

Test

ID	Pixel size	Type
02	(0.9375, 1.5, 0.9375)	A
10	(1.0, 1.5, 1.0)	B
15	(0.8371, 1.5, 0.8371)	C

The grayscale intensities are also diverse among images. As a visual example we present the intensity distribution in the validation set in figure 1 below. These two main characteristics of the images have to be taken in consideration in a preprocessing step, before any model is designed or trained, as they could affect the final segmentation and the performance of the segmentation algorithms.

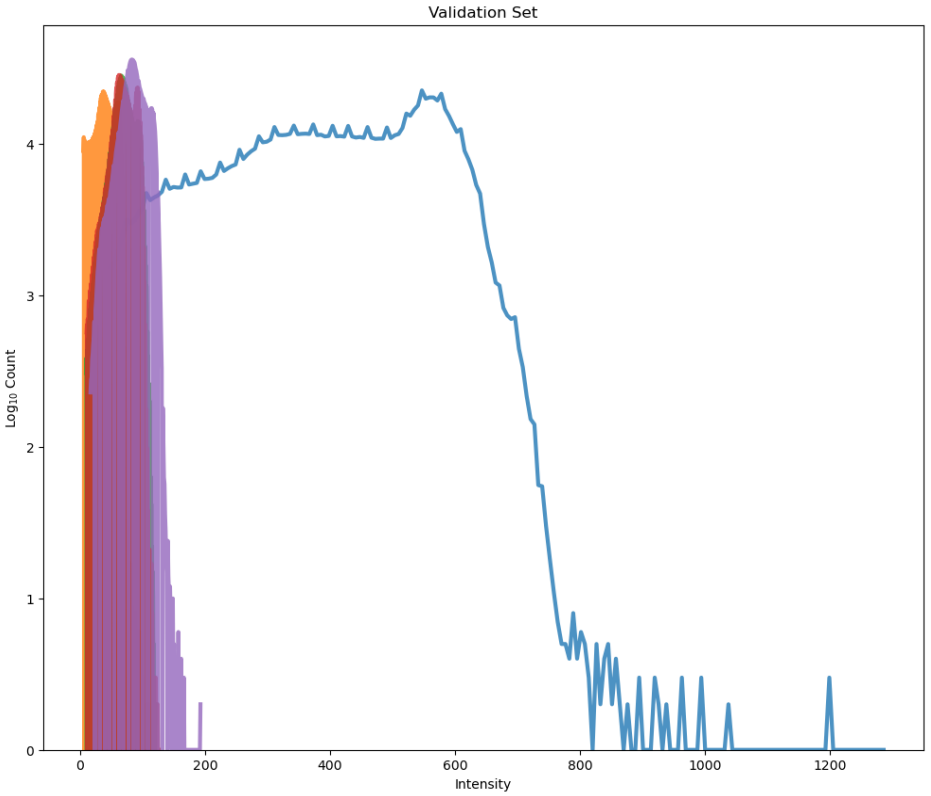

Figure 1: *Validation set images intensity log10 count.*

2. Proposal Analysis

We decided to separate the main workflow in two main sections: preprocessing and segmentation approaches.

2.1 Preprocessing

As described in the first section, the dataset has two main challenges: different pixel sizes and heterogenous intensity ranges. For the first problem, the pixel size, we decided to rely on the segmentation algorithm to solve the issue. This means that we will consider this diversity of pixel size in the designing of the models (e.g. in the registration pipeline). On the other hand, we decided to apply intensity normalization to mitigate the intensity heterogeneity.

The planned preprocessing pipeline was as follow:

1. **Bias-field correction**: for correcting nonuniformity associated with MR images.
2. **Intensity normalization:** for correcting inconsistent intensity scales across (and within) sites and scanners.
    a. Individual normalization methods (each image individually)
    b. Sample-based normalization methods (in batches or groups of images)

2.2 Segmentation algorithms

Classic approaches

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

We decided to recall all previous models used on the lab sessions and apply the best parameters and optimization learnt by us. This includes both intensity-based and atlas-based segmentation techniques. Based on our experience, **atlas-based** segmentation usually yields promising results for which we decided to go deeper with these techniques through **multi-atlas** algorithms. These methods have proven to be better when intensity information is also included. The proposed methods to developed were the following:

- **Multi-atlas**
  - Most similar atlas
  - Majority voting atlas (a.k.a. mean atlas)
  - Weighted voting atlas (using similarity metric as voting weight)
  - Top atlases (using only the most similar atlases)
- **Probabilistic atlas**
  - Bayesian method: Combining probabilistic atlas coming from top atlases and tissue models.
- **Intensity-based**
  - Tissue model
  - Expectation-maximization (EM) algorithm
    - Traditional (with several initializations)
    - With atlas into and after EM algorithm

Image registration

A key element of the atlas approaches is image registration. Throughout several courses this semester, we had the opportunity to use **Elastix**, a powerful registration software with an easy-to-use Python API called *SimpleElastix***.** *SimpleElastix* has proven efficient and diversified, allowing us to even win second place at the *Non-rigid 3D lung CT registration challenge* of the Medical image registration module. Even with all these advantages, *Elastix*, as any other classic registration technique, has shown us to have two main drawbacks: long execution time and parameter tuning. A good registration pipeline needs, then, long waiting times to run and a lot of fine-tuning to match the dataset requirements. To overcome these two issues we decided to use **VoxelMorph**, a Deep learning general purpose library for learning-based tools for image registration.

Deep learning approaches

Finally, given that we have not yet explored any **Deep Learning** segmentation method, we take this opportunity to search for similar projects, try to implement their approaches and compare the obtained results with those obtained with our well-known classic algorithms. From here, the proposed deep learning implementations include:

- **U-net segmentation**
- **Deep Residual U-net (ResUnet) segmentation,**
- **DenseUnet segmentation**

3. Design and implementation of the proposed solutions

3.1 Preprocessing

Bias field removal

The first part of our preprocessing pipeline was the bias field correction as we know from the labs that skull stripping and bias field removal are the first common preprocessing steps. In our case the former is already provided so we only needed to solve the latter. We achieved this by using the *SimpleITK N4 Bias field correction filter.* This filter is applied directly on the SITK image and returns a corrected version of the image.

Normalization

Normalization was achieved using a Python library called *intensity-normalization* specifically designed for brain MRI imaging normalization. This library contains several normalization methods from which we can mention two main groups:

- Individual image-based methods: normalize images based on one time-point of **one subject**
- Sample-based method: normalize images based on a **set of images** of multiple subjects

Considering that there is no preferred method and that their performance depends on the dataset, we decided to use both and visualize their effect on the images. Among all the methods available, we selected the most common one per group:

- *Fuzzy C-means* (FCM)-based tissue-based mean normalization (individual)
- *Least squares (LSQ)* tissue mean normalization (set of images)

Figure 2 shows the intensity counts for the validation set (same as in figure 1) after applying both methods.

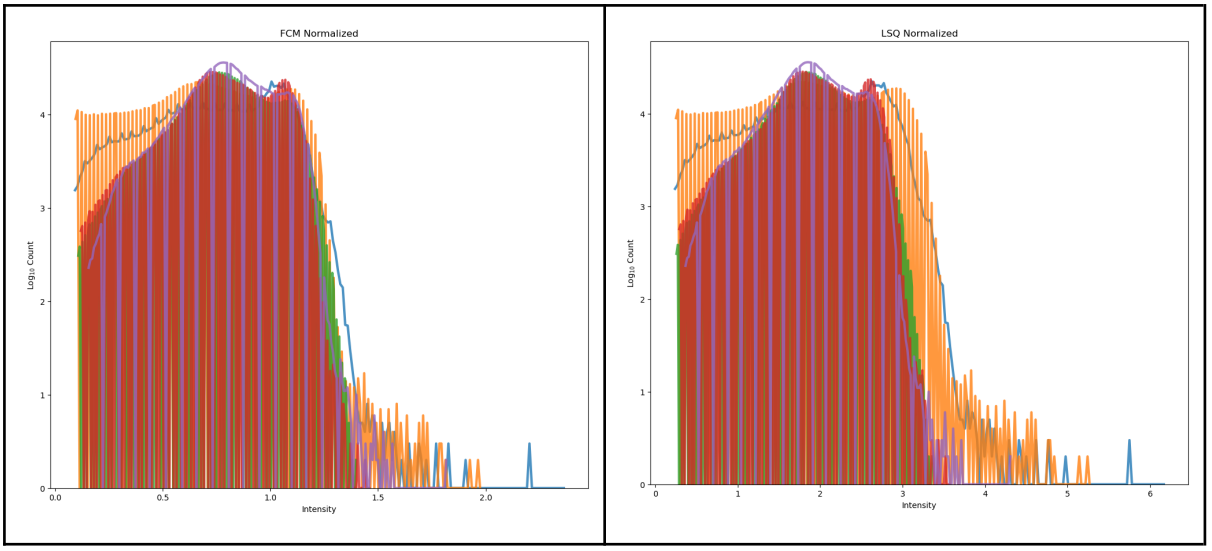

Figure 2: *Validation set images intensity log10 count after normalizations. On the left FCM normalization and on the right LSQ normalization.*

From the normalization results we can see that both methods are succeeding in normalizing the images, with slight differences. As we perceived more matching between intensity counts using FCM, and because we would prefer a more generalizable less group-based normalization, we decided to use FCM normalization on all our dataset splits, visually checking that all intensity counts are normalized.

3.2 Classic approaches

Vocabulary clarification
Individual atlas: Each of the images and its labels in the training set.

3.2.1 Multi-atlas

Multi-atlas segmentation methods consist of using prior spatial information coming from other similar images and their known segmentation labels (individual atlases) and combining their labels using a *label fusion decision*. This method relies on two main characteristics: the quality of the segmentation labels of the individual atlases and the registration quality.
Before anything, to apply any multi-atlas method we needed to register all 10 training images to the target (validation) images individually.

Image registration
We used the VoxelMorph library [3] and 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. The implementation of this registration pipeline can be found in the *most_similar.ipynb* Jupyter notebook. In a nutshell, because of the easy-to-use API given by VoxelMorph, for this registration pipeline to work we only need to give the fixed and moving images, and a moved version of the moving image will be given, as well as a warp to

transform other images as well. In figure 3 we show the main part of the code used to register and, as it can be seen, it is simple and straightforward.

```
with tf.device(device):
    # load configuration
    config = dict(inshape=inshape, input_model=None)
    #register and get warp
    warp = vxm.networks.VxmDense.load(model_path, **config).register(moving, fixed)
    #use warp on moving images and labels
    moved = vxm.networks.Transform(inshape, nb_feats=nb_feats).predict([moving, warp])
    moved_label = vxm.networks.Transform(inshape, nb_feats=nb_feats).predict([label, warp])
```

Figure 3: *Main part of the code used to register images.*

To compare the performance of *VoxelMorph* against the traditional registration approaches, we used *SimpleElastix* library as well, using as reference for registration the parameter files found in the *Elastix parameter file Zoo.*

Most similar atlas

The easiest segmentation that can be built by having multiple atlases registered to the same target image (validation image) is to select the atlas with the best similarity metric. This means that we will trust the label propagation of only the most similar atlas to our target image. There exist several similarity metrics but experience with the MIRA course showed us that the most reliable metric for brain MRI imaging is the *Mattes Mutual Information (MMI)* metric, used by many registration tools like *Elastix*. The atlas with the best MMI was selected per validation image and the propagated labels were used as segmentation masks. Table 2 shows an example of how all individual atlases were registered into validation image 11, in descending metric order, having patient 9 as the most similar in this case. Other metrics like MSE and correlation were also used but the most similar atlas was always the same.

Id train	CSF	GM	WM	Metric
9	0.7802	0.8087	0.8387	-0.4190
8	0.7038	0.7919	0.8236	-0.4172
7	0.7592	0.7744	0.8199	-0.4167
5	0.7149	0.8171	0.7772	-0.3992
3	0.6834	0.8013	0.7716	-0.3973
4	0.7524	0.8116	0.7848	-0.3884
1	0.7461	0.8106	0.7546	-0.3832
6	0.6706	0.7871	0.7369	-0.3727
16	0.6205	0.7321	0.7369	-0.3259
18	0.6844	0.7148	0.7405	-0.3091

Table 2: *Example of a most similar atlas table, with Dice score per tissue and the MMI metric for validation image 11. The id train is the id of the individual atlas.*

Majority voting (mean atlas)

Majority voting is the simplest fusion condition for multiple atlases, and consists in simply summing the probabilities of all registered individual atlases and dividing that sum by the number of individual atlases. Another way to see this is having a weighted voting, where all weights are defined the same, w = 1/n, where n is the number of individual atlases. The implementation of this atlas can be found in the *atlas_models.ipynb* notebook.
For the simplicity of this model, we decided to also apply this method to the Elastix registered atlases and compare the results.

Weighted voting atlas

Different to the previous method, in this case the weights for each individual atlas will be different. We chose to use the similarity metric as weight because its value gives us a numeric value of how similar the individual atlas and the image were. We expect to give more importance to individual atlases that are more similar to our image. The actual value of the weight is normalized by the sum of all metrics, given by the equation

$$w_i = \frac{S_i}{\sum_i S_i}, \quad (1)$$

where S_i is the similarity metric of the i-th individual atlas.

Given that using different metrics beside MMI did not give different results in the most similar atlas selection, we decided to keep only MMI as the weighting metric for this atlas method. The implementation can also be found in the *atlas_models.ipynb* notebook.

Top atlases

The most similar atlas tables, as the one shown in table 2, indicated to us that some individual atlases are significantly less similar to our validation images individually, so including them in the label fusion, even if weighted, may reduce the segmentation performance. Because of this, we decided to select only the top 3 individual atlases and fuse them. We also decided to put a condition in which the individual atlas would be fused only if the metric difference was less than 0.05. This guarantees us that all atlases keep optimal similarity values with the target. Because the top atlases already have high similarity values with the target, we decided to fuse the labels simply by averaging them.

3.2.2 Probabilistic atlas

Because the Bayesian approach gave us top results in the lab sessions, we decided to include it in this project. The Bayesian framework consists in combining both the tissue model probabilities (intensity information) and the probabilistic atlas (spatial information), by directly multiplying both probabilities per tissue. The tissue model obtaintion will be explained in the next section. As for the probabilistic atlas, we decided to use the probabilities of the top atlases instead of the whole training set, as we consider them more faithful to the target spatial information.

3.2.3 Intensity-based methods

Tissue model

Calculating the tissue model would be a straightforward process if the intensities were all in the same range and with the same width of histogram bins (usually integer grayscale values). Nevertheless, in our case, due to the normalization process, the original bin widths of the because of this, in order to obtain the tissue model we needed to map this float ranges back to a integer range, where all bins have the same length (1 unit). After applying this mapping we were able to construct a tissue model, which is shown in figure 4. The implementation can be found in the *tissue_models.ipynb* notebook.

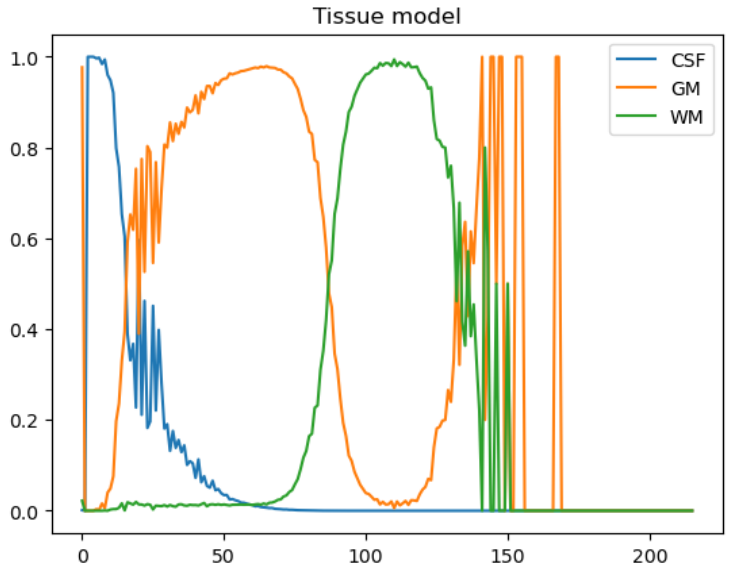

Figure 4: *Tissue model constructed using the training set.*

Expectation-maximization algorithm

Finally, we decided also to include the EM algorithm, even though it showed poor performance in the lab sessions. The reason is that, even though they normal implementation showed low Dice scores, including the atlas information after the maximization step showed promising results.

3.3 Deep Learning approach

Early approaches to medical image segmentation often depend on edge detection, template matching techniques, statistical shape models, active contours, and machine learning, etc. Due to the rapid development of deep learning techniques, medical image segmentation will no longer require hand-crafted features and convolutional neural networks (CNN) successfully achieve hierarchical feature representation of images, and thus become the hottest research topic in image processing and computer vision. [4]. The reason for the success of the CNN's are because of their nature of being insensitive to image noise, blur, contrast, etc., providing excellent segmentation results for medical images.

According to the number of labeled data, it is possible to categorize the methods into three groups as supervised, unsupervised and weakly supervised models. Also thinking about the type of the segmentation task, image semantic segmentation is a pixel level classification that assigns a corresponding category to each pixel in an image. In our project, since the dataset contains labels for the training and validation set and the goal was to assign corresponding categories to each pixel, it is considered as an image semantic segmentation with a supervised approach.

In order to achieve high results in semantic segmentation, researchers proposed the encoder-decoder structure that is one of the most popular end-to-end architectures. In these structures, an encoder is frequently used to extract image features while a decoder is frequently used to output the final segmentation findings and restore extracted features to the original picture size.

One of the well known architectures in encoder-decoder fashion is the **U-net model** established in 2015 [5], which has been widely used for the medical image segmentation. In this project, we have applied 2D U-net and Residual U-net models slice by slice to the whole dataset. Because of the limitations in the number of images in our dataset, we have trained our model by applying patch based applications together with data augmentations.

3.3.1 Data preparation

In the data preparation phase for the deep learning model, our input data already had bias field removal applied and intensity normalization. In order to increase the number of samples in the dataset, first data sampling using patches and then data augmentations were applied.

Data sampling using patches can be an effective way to train models. By extracting a smaller set of patches, it is possible to train models more efficiently, while still capturing important features and variations in the data. In our dataset preparation, we have extracted patches of sizes of 32x32. Even though we experimented with different sizes of the patches, 32x32 was the most convenient.

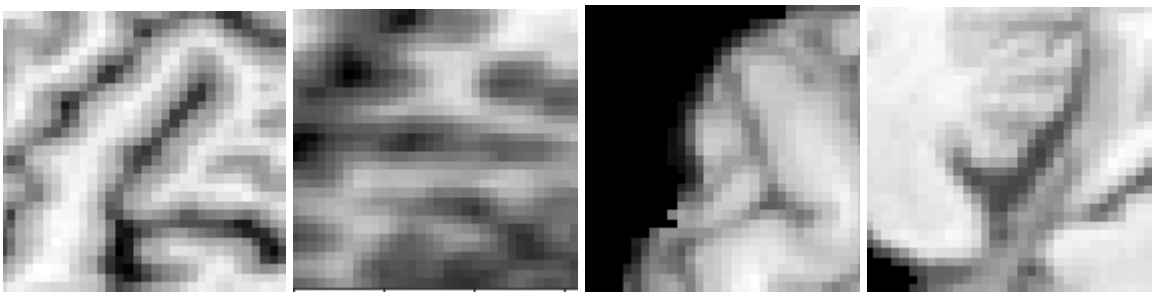

Figure 5: *2D Patches extracted from the training set*

Data augmentation is a technique used to artificially increase the size of a dataset by generating modified versions of the existing data. It is applied to increase the performance of the model. By increasing the number of samples, it is possible to train larger models with

higher numbers of parameters leading to higher accuracies. Data augmentations were applied using ImageDataGenerator from Tensorflow Keras. The applied data augmentation parameters are given in the table below.

Parameter Name	Setting
featurewise_center	True
featurewise_std_normalization	True
rotation_range	50
width_shift_range	0.2
height_shift_range	0.2
shear_range	0.2
zoom_range	0.3
horizontal_flip	True
fill_mode	'nearest'

Table 3: *Data augmentations applied to patches*

In the deep learning applications, we have used the baseline architecture provided by Jose Bernal's session on Deep Learning. Models were coded using Tensorflow Keras framework and they were trained on Google Colab platform which has GPU of Tesla K80-12GB and RAM of 12GB.

3.3.2 2D Unet architecture

The U-Net uses a perfect symmetric structure with skip connections to address issues with standard CNN networks used for medical image segmentation. Different from the common image segmentation, medical images usually contain noise and show blurred boundaries. This results in hardness in the detection of the objects in the medical images, if the detection is only based on low-level features. In order to solve the problem, U-net effectively fuses the low-level and high-level features by combining low and high resolution feature maps through skip connections. The U-net architecture is shown in the Figure 6 below.

3.3.3 2D ResUnet architecture

Residual U-Net (ResU-Net) is a variant of the U-Net architecture that was developed for image segmentation. It extends the U-Net architecture with residual blocks, which are composed of multiple convolutional layers with skip connections that add the input of the block to the output of the block. This allows the model to use the residuals learned by the convolutional layers in the decoder part to improve the segmentation task in the encoder part.

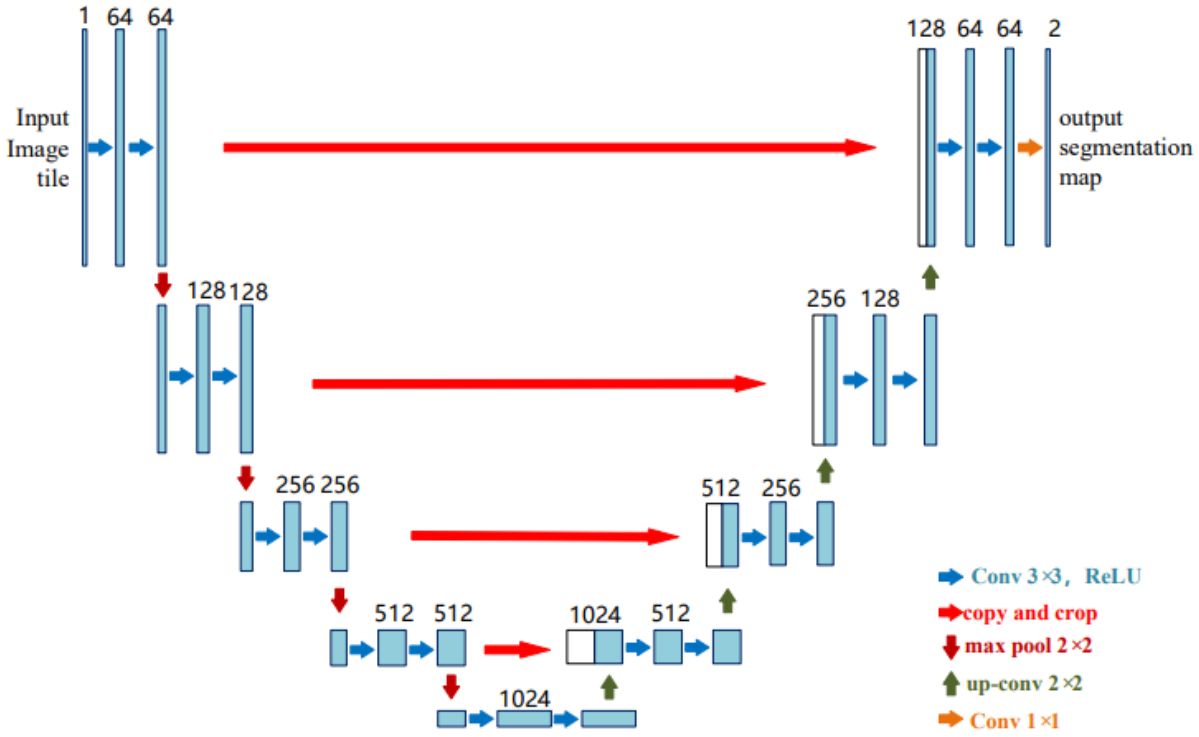

Figure 6: *The U-net architecture [5]*

The Residual U-Net architecture consists of a encoder part (left side of the U-shape) and decoder part (right side of the U-shape). The encoder part is composed of a series of convolutional and max pooling layers that reduce the spatial size of the input data, while the decoder part is composed of a series of convolutional and up-sampling layers that increase the spatial size of the input data. The Residual U-Net architecture also includes skip connections, which allow the model to use information from the encoder part to help with the segmentation task in the decoder part.

3.3.4 2D DenseUnet architecture

Dense U-Net (DenseUnet) is a variant of the U-Net architecture that was developed for image segmentation. It consists of an encoder path and a decoder path. The encoder path is composed of a series of convolutional and max pooling layers that reduce the spatial size of the input data, while the decoder path is composed of a series of convolutional and up-sampling layers that increase the spatial size of the input data. The Dense U-Net architecture also includes skip connections, which allow the model to use information from the encoder path to help with the segmentation task in the decoder path. In addition, it includes dense connections, which allow the model to use information from all previous layers in the decoder path to improve the segmentation task.

Dense U-Net has been shown to achieve state-of-the-art results on a variety of image segmentation tasks, including medical image segmentation and object detection. It is known

for its ability to handle small and imbalanced datasets, and for its robustness to noise and artifacts.

The baseline applications for 2D ResUnet and 2D DenseUnet are built on the basis of the applications of Kolarik et al. [6].

3.3.5 Model parameters

In this section, we would like to discuss the parameters used and the hyper parameter tuning for our U-net models.

Optimizer: An optimizer is a machine learning algorithm that is used to update the parameters of a model based on the training data. The goal of the optimizer is to find the set of model parameters that minimize the loss function, which measures the difference between the model's predictions and the true labels of the training data. In our experiments we have experimented with the two well known optimizers:

1. **Stochastic Gradient Descent (SGD):** This is a simple optimizer that updates the model parameters by taking the gradient of the loss function with respect to the parameters and moving in the opposite direction.
2. **Adam (Adaptive Moment Estimation):** This is a popular optimizer that uses estimates of the first and second moments of the gradients to adapt the learning rate for each parameter.

Loss function: A loss function is a function that is used to measure the difference between the predicted output of a model and the true output. In the context of deep learning, the loss function is used to train the model by updating the model parameters in order to reduce the loss. In our experiments, we have experimented with three different loss functions:

1. **Categorical Cross-Entropy (CCE):** This is a common loss function for multi-class classification tasks, where the goal is to predict a categorical label (e.g., one of several classes). It can be used for image segmentation by treating each pixel in the image as a separate class and predicting a class label for each pixel.
2. **Dice Loss:** This is a loss function that measures the overlap between the predicted and true masks. It can be used for multi-class image segmentation by computing the Dice score for each class separately and averaging the scores.
3. **Focal Loss:** This is a loss function that is designed to address class imbalance, which is often a problem in image segmentation tasks. It down-weights easy examples and focuses on hard examples, which can help the model learn to classify rare classes more accurately.

Metrics: It is used to evaluate the model's predictions on a validation or test set, and to compare the performance of different models. In our project, we have used three different metrics:

1. **Hausdorf distance (HD):** It is a measure of the distance between two sets of points in a metric space. It is defined as the maximum distance between a point in one set and the nearest point in the other set.
2. **Dice Similarity Coefficient (DSC):** It is a measure of the overlap between two sets of points, to evaluate the similarity between a predicted segmentation mask and a true mask. It is calculated as the ratio of the intersection of the two masks to their union.
3. **Average Volumetric Distance (AVD):** It is a measure of the distance between two sets of points in three-dimensional space. It is calculated as the average of the distances between each point in one set and its nearest neighbor in the other set.

After performing hyper-parameter tuning in our models, we have found that the best parameter setting for our U-net models were given like in the table below:

Parameter Name	Setting
Number of epochs	30
Batch size	20
Patch size	32X32
Patience (Early stopping)	10
Optimizer	Adam
Dropout rate	0.2

Table 4: *The optimum parameter setting for our models*

4. Experimental section and results analysis

4.1 Comparison between registration frameworks

After applying both methods for the 50 registrations needed for the multi-atlas approach, we found the following:

1. Execution time
    a. Elastix: Ranging from 2 minutes to 4 per image, depending on the complexity of the parameter file
    b. VoxelMorph: 13 seconds per image
2. Parameter file
    a. Elastix: Fine-tuning needed to find the best parameter settings
    b. Voxel Morph: Automatically found by the network.

Finally, as it will be seen in the result section, the registration with VoxelMorph gave better results in terms of the segmentation Dice score, thus we continued using only VoxelMorph.

4.2 Classic approaches results

A summary of the classic Dice score results can be seen in figure 7-10, with one boxplot for each tissue.

CSF segmentation

It is not a surprise that the segmentation of the CSF is the most challenging task among the three tissues. From figure 7 we can spot two main things. First, both the issue model and the pure EM method, with Bayesian initialization, failed to segment the CSF. This can be attributed to the fact that these two models are pure intensity-based models, and that the CSF range of intensities is very narrow and overlaps considerably with the GM, as seen in the tissue model in figure 4. Removing the two intensity cases allows us to better see the other results in figure 8.

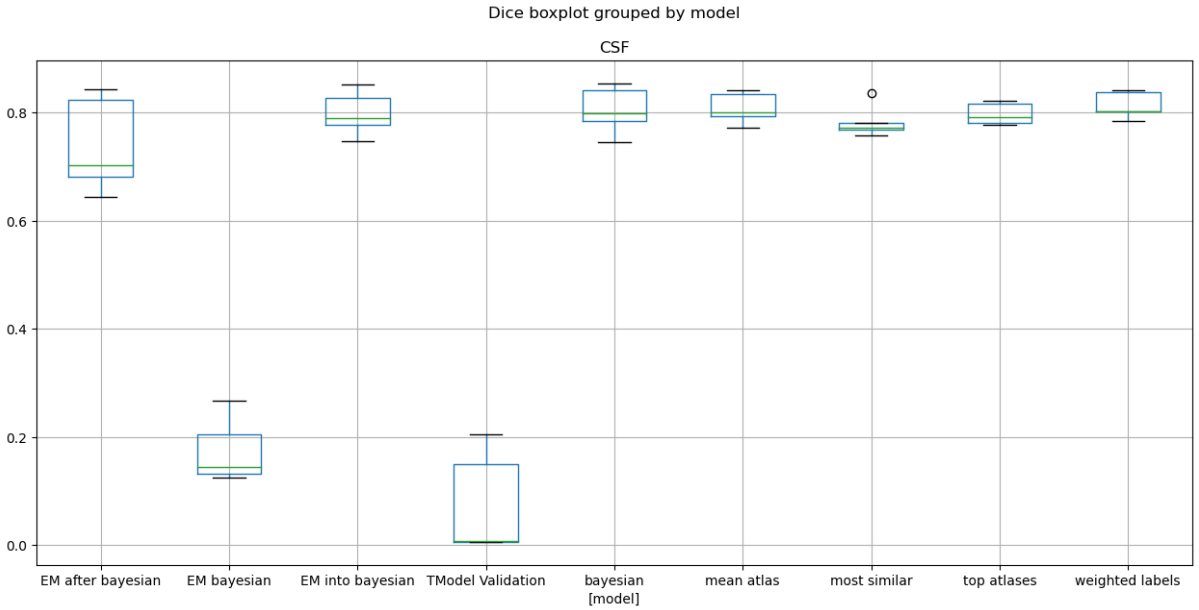

Figure 7: *Dice score boxplot of CSF for classic methods.*

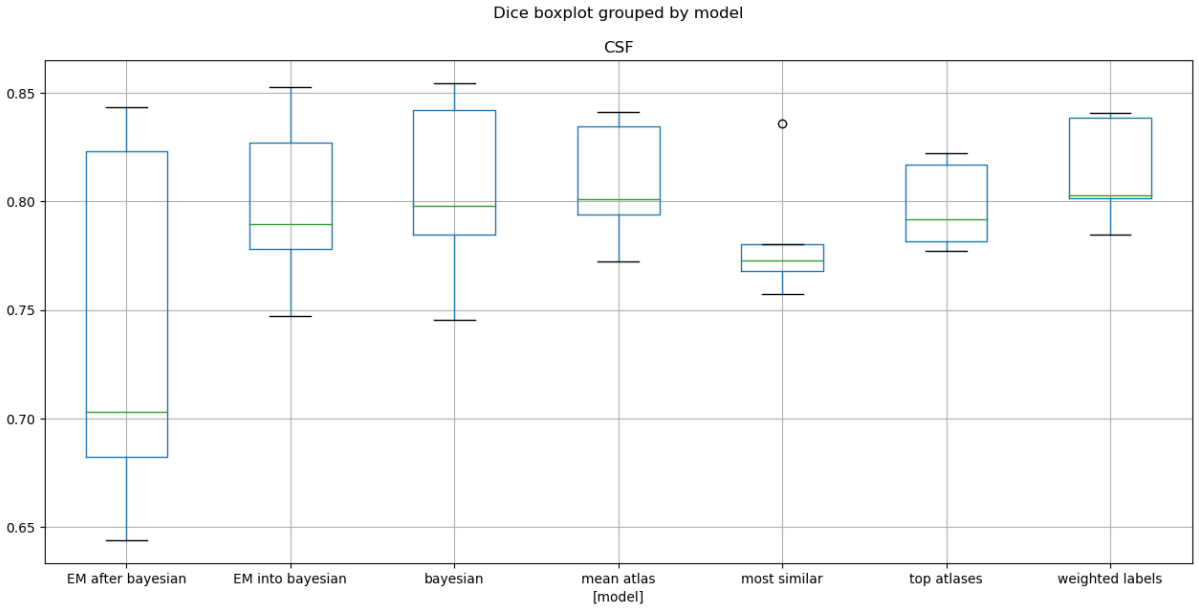

Figure 8: *Dice score boxplot of CSF for classic methods, without intensity-based methods.*

We can see from figure 8 that all models have similar performances. The most similar atlas approach has the most consistent results whereas the EM after Bayesian showed the most

spreaded Dice scores. If we consider both good mean performance and Dice distribution, the best classic model for CSF would be the weighted atlas. Looking at the volume difference (VD) box plot in figure 9, we can see that the weighted atlas approach has one of the smallest VD values (very close to zero) among the other models. The figure 9 box plot also shows that this method undersegments the CSF, as the VD is negative.

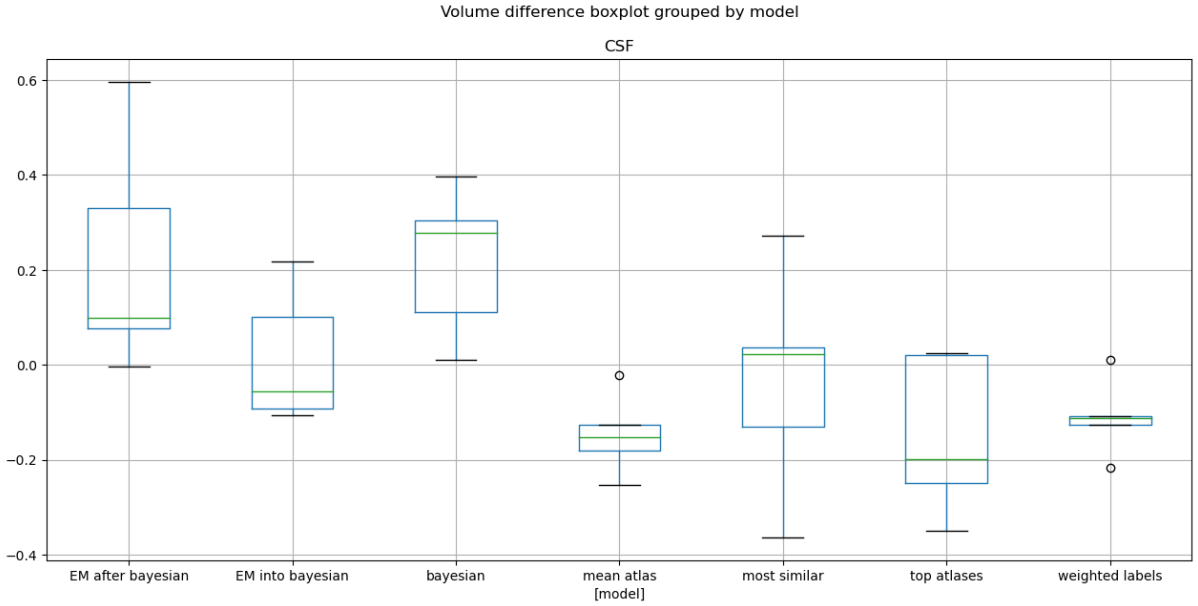

Figure 9: *Volume difference boxplot of CSF for classic methods, without intensity-based methods.*

Finally the Hausdorff distance (HD) of these same models for CSF (figure 10) shows that the weighted atlas approach has one of the lowest HD values. We can conclude that the weighted atlas method was found to be the best among our other classic segmentation methods for CSF.

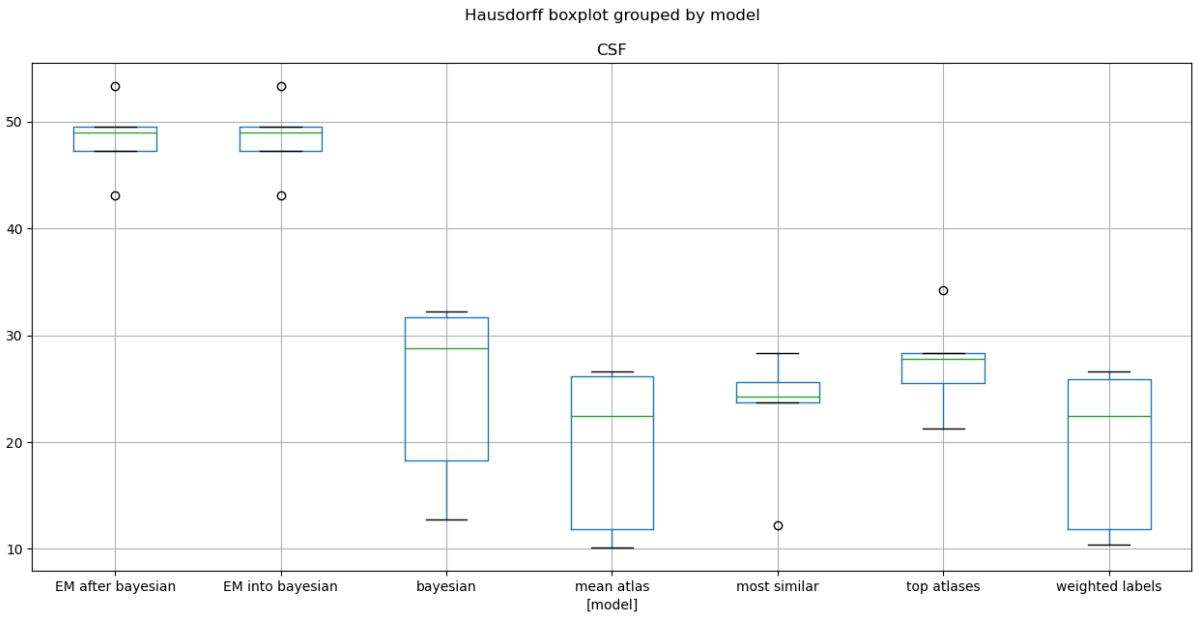

Figure 10: *Hausdorff distance boxplot of CSF for classic methods, without intensity-based methods.*

Gray and white matter segmentation

Contrary to the CSF, the intensity-based segmentation methods for GM and WM generally outperformed the atlas-based methods, as shown in figure 11 and 12.

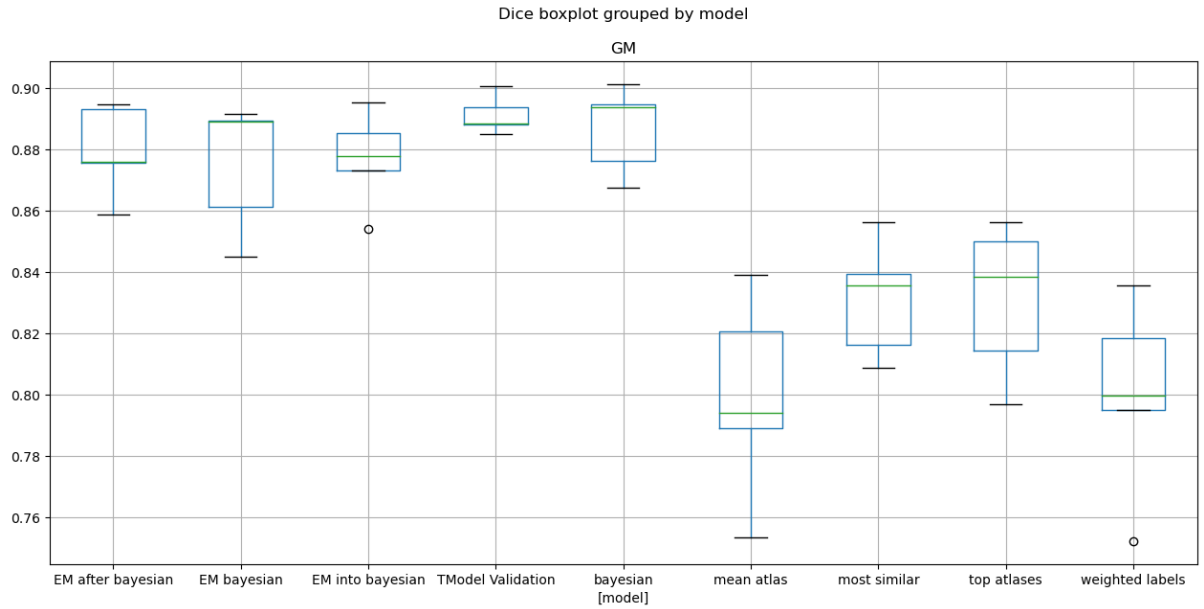

Figure 11: *Dice boxplot of GM for classic methods*

In the case of GM, it is remarkable to see that a simple method as the tissue model essentially was the best segmentation method, with a high Dice score for all validation images. It is also easy to see in figure 11 that all models that use intensity information (5 on the left) outperformed the others that did not and that relied only on prior information (4 on the right). The WM is not that clear and the GM case, as figure 12 shows. Still, the trend is clear and the best model is still the tissue model.

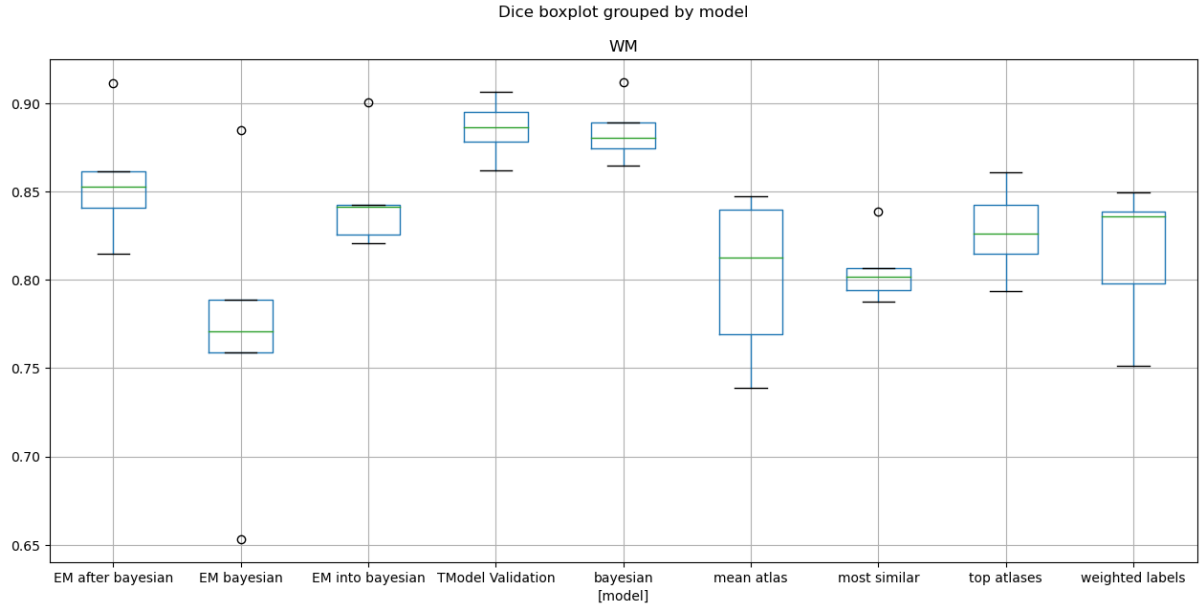

Figure 12: *Dice boxplot of WM for classic methods*

Considering how well the tissue model performs on GM and WM, it is worth noticing that the combination of this intensity information with the top atlas probabilities in the Bayesian model replicates similar high Dice scores for these two tissues and, at the same time, allows for a good CSF segmentation, something that the tissue model was incapable of reaching by it self.
With all this in consideration, we can conclude that the **Bayesian model is the most robust** across all three main brian tissues and can be considered the best classic segmentation method among the ones we built.

4.3 Deep learning approaches results

Overall, the deep learning approaches gave us one of the best and robust performances among all the methods we have tried. In the Figures below, it is possible to observe the success of the different U-net model variations with different loss function settings.

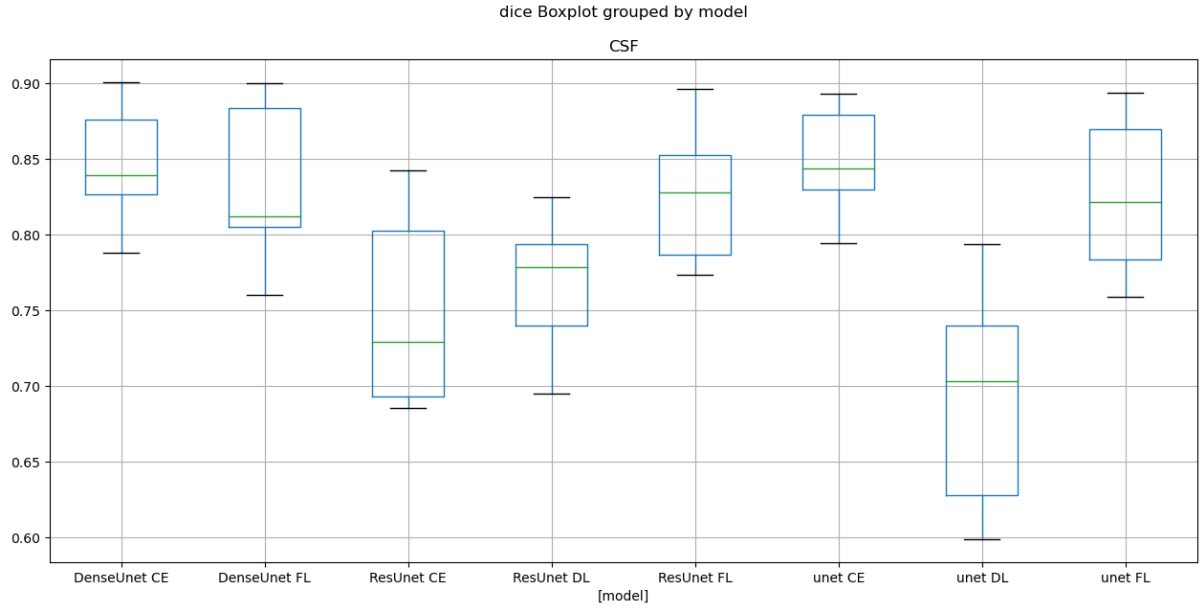

Figure 13: *CSF tissue deep learning model performances - DSC*

In the Figure 13 given above, for the segmentation of the CSF tissue, DenseUnet with cross entropy loss and with focal loss and U-net model with the cross entropy loss, were ranking the best performances.

The mean of the evaluation metrics across validation images, for these top models, for CSF tissue are given in the table below:

Model	Dice Coefficient	Hausdorf distance	AVG
DenseUnet CE	0.845947	23.587460	0.019853
DenseUnet FL	0.831970	31.908549	0.070923

U-net CE	0.847816	84.642468	-0.000466

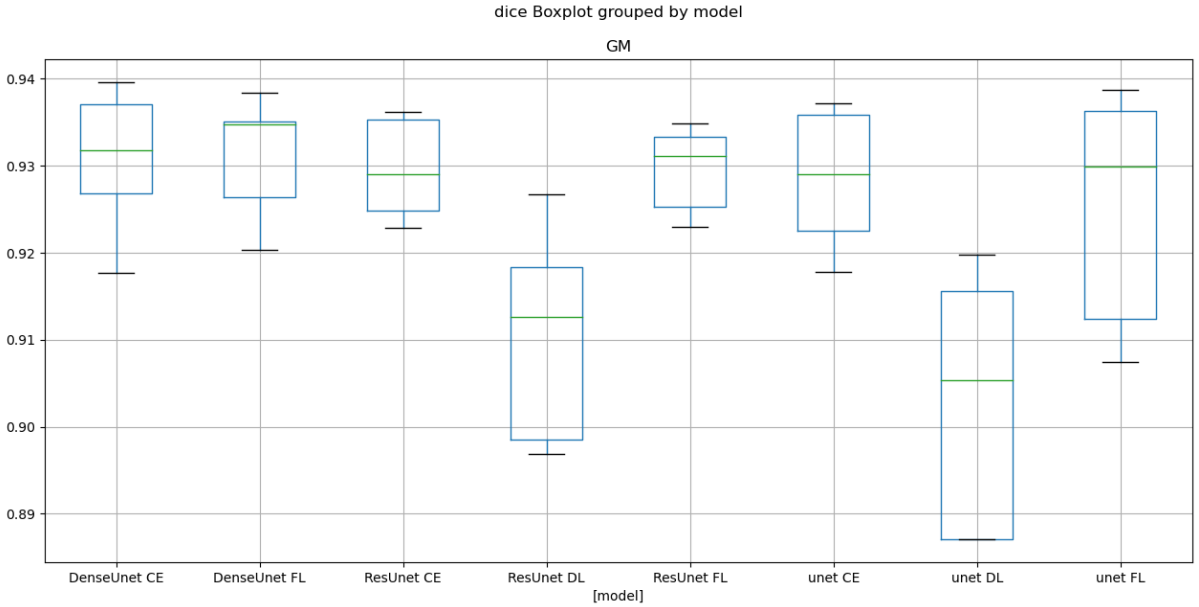

Figure 14: *GM tissue deep learning model performances - DSC*

In the Figure 14 given above, for the segmentation of the GM tissue, DenseUnet with cross entropy loss and U-net with cross entropy and focal loss were giving the best results.

The mean of the evaluation metrics across validation images, for these top models, for GM tissue are given in the table below:

Model	**Dice Coefficient**	**Hausdorf distance**	**AVG**
DenseUnet CE	0.930554	12.614502	0.021917
U-net CE	0.928448	16.194567	0.004582
U-net FL	0.924930	36.620665	0.017464

In the Figure 15 given above, for the segmentation of the WM tissue, DenseUnet with focal loss and cross entropy and ResUnet with cross entropy loss were giving the best results.

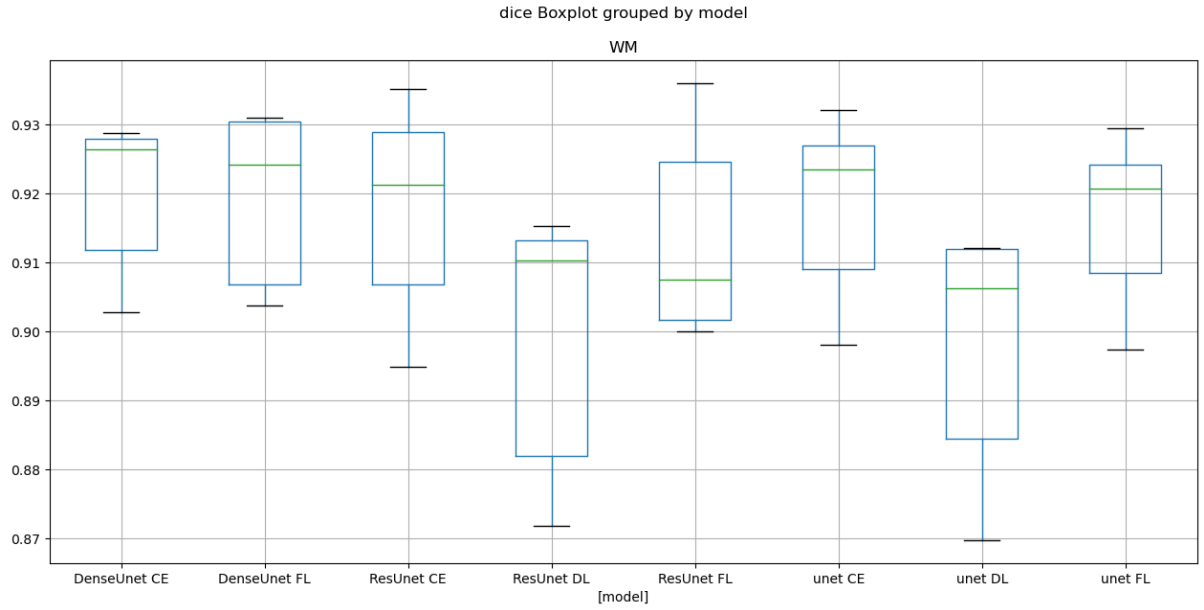

Figure 15: *WM tissue deep learning model performances - DSC*

The mean of the evaluation metrics across validation images, for these top models, for WM tissue are given in the table below:

Model	Dice Coefficient	Hausdorf distance	AVG
DenseUnet FL	0.919277	10.238725	-0.004913
DenseUnet CE	0.919571	9.299474	-0.038182
ResnetUnet CE	0.917403	9.363837	0.007880

Considering the common success of the **DenseUnet model with the categorical cross entropy loss** among three tissues, it is selected as our final model setting. The training and validation of the model took approximately 30 minutes.

Finally, in figure 16 we make a qualitative comparison of the segmentations made by both the best classic and DL models.

4.4 Comparing the best classic and DL approaches

Finally, we wanted to make a direct comparison between the best classic and DL models, and the comparison is shown in figure. We can easily spot that the DL model outperforms the classic method in all three brain tissues, with significant differences especially for the GM and WM case. This does not come as a surprise as we know the power that DL methods pose for segmentation tasks.

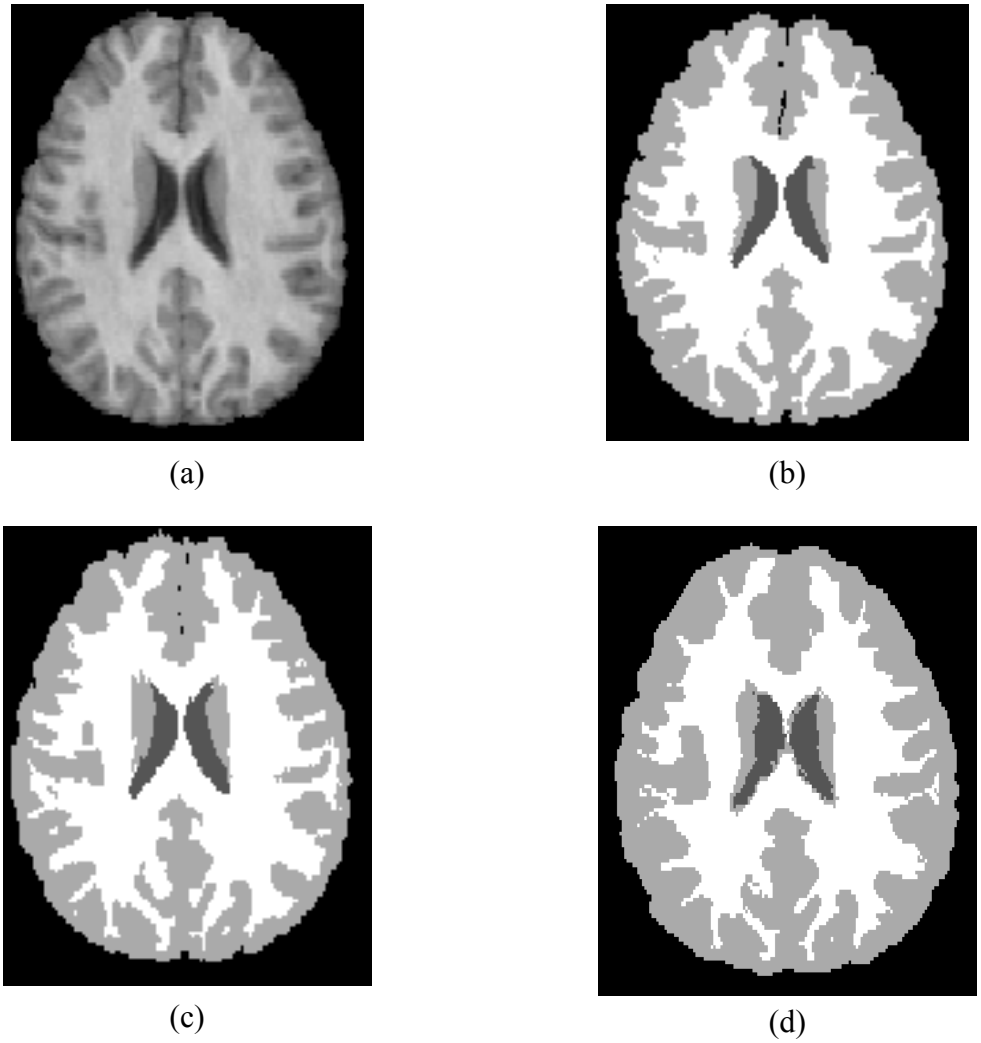

Figure 16: *Segmentation results for the validation image for IBSR_14 where (a) Original image, (b), Segmentation mask (c), Segmentation using the DenseUnet CE. d) Segmentation using the bayesian model*

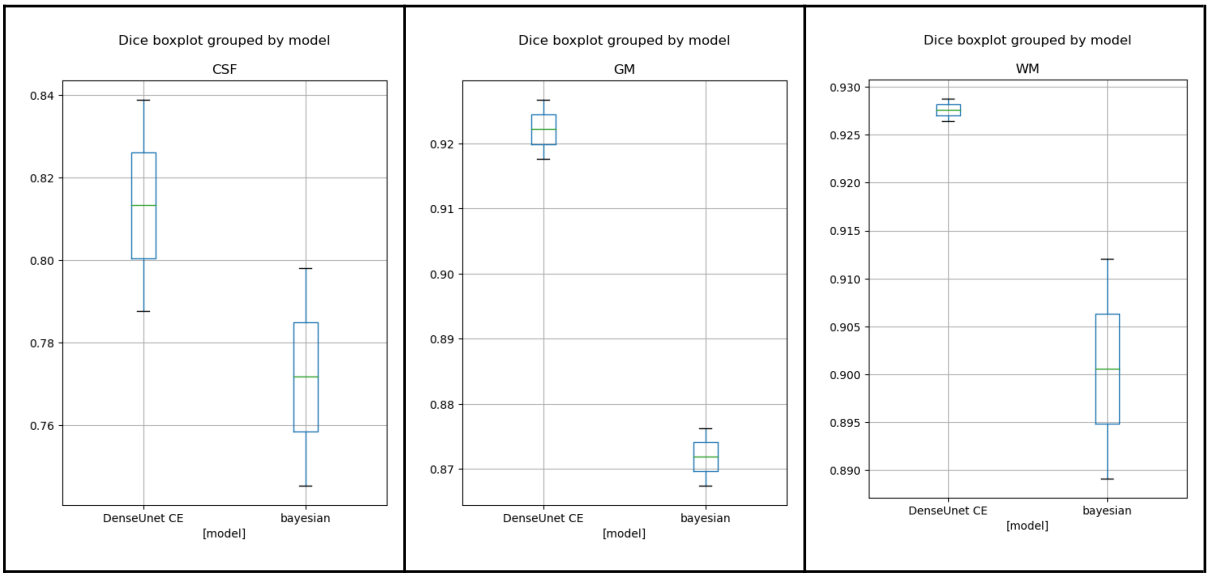

Figure 17: *Comparison of classic and DL methods in the validation set.*

5. Organization and development of the coursework

The organization of the project work was planned in the last lab sessions with the attendance of the group members in a brainstorming. The ideas from the brainstorming were kept in a shared file and in the following weeks the necessary literature review and research has been completed.

In the early weeks of December, some ideas were eliminated due to lack of resources and the project time limitations. We have decided to follow two distinct pipelines including classical and modern (deep learning) approaches in order to implement the knowledge that was taught in the lectures and also explore the modern methods. The project experiments were applied during December and the latest changes and documentation were finalized in early January.

6. Conclusions

Medical image segmentation is a challenging task considering the nature of the medical images that are containing artifacts and noises. In this project, our aim was to implement medical image segmentation in the main three tissues of the brain MRI images. Despite the challenges faced, such as different voxel sizes and intensity inhomogeneities, we were able to achieve successful results using both traditional approaches and deep learning methodologies.

In classical approaches, we applied three main experiments. Firstly, we applied a series of multi-atlas methodologies including most similar atlas, majority voting (mean atlas), weighted voting atlas and top atlasses approaches. Secondly, we applied the Bayesian framework and finally, intensity-based methodologies including, tissue model and the expectation maximization algorithm with different initializations and variations. Among all the experiments from the classical approach, we obtained 0.80, 0.88 and 0.88 mean dice scores in the validation test, for CSF, GM and WM respectively, using the Bayesian model.

In the deep learning section, we experimented with the recent advanced U-net based algorithms. In this case, we applied 3 main architectures, namely, U-net and ResUnet and DenseUnet. Due to the size of the dataset, we decided to apply a patch based approach to train our model. To match the requirements of the semantic segmentation problem, we experimented with several loss functions such as focal loss, dice loss and categorical cross entropy loss.

Overall, we were able to obtain the best results using the DenseUnet model with categorical cross entropy loss function reaching up to 0.84, 0.93 and 0.91 mean dice scores in the validation set for the tissues CSF, GM and WM sequentially. Due to the success of this model, it was selected to be used in our test dataset for the challenge.

7. References

[1] Dora, L., Agrawal, S., Panda, R., Abraham, A. (2017). State-of-the-art methods for brain tissue segmentation: A review. IEEE reviews in biomedical engineering, 10, 235-249.
[2] NITRC NeuroImaging Tools and Resources Collaboratory
https://www.nitrc.org/projects/ibsr
[3] VoxelMorph: A Learning Framework for Deformable Medical Image Registration
G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, A.V. Dalca.
IEEE TMI: Transactions on Medical Imaging. 38(8). pp 1788-1800. 2019.
[4] Wang, R., Lei, T., Cui, R., Zhang, B., Meng, H., & Nandi, A. K. (2022). Medical image segmentation using Deep Learning: A Survey. IET Image Processing, 16(5), 1243–1267. https://doi.org/10.1049/ipr2.12419
[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. cite arxiv:1505.04597Comment: conditionally accepted at MICCAI 2015.
[6] Kolařík, M., Burget, R., Uher, V., Říha, K., & Dutta, M. K. (2019). Optimized High Resolution 3D Dense-U-Net Network for Brain and Spine Segmentation. Applied Sciences, 9(3), vol. 9, no. 3.