MISA FINAL PROJECT

Tissue segmentation project (IBSR18 dataset)

January 19, 2018

Vu Hoang Minh
Yeman Brhane Hagos

1 INTRODUCTION

The central nervous system (CNS) is the part of the nervous system consisting of the brain and spinal cord. It integrates information it receives from and coordinates and influences the activity of all parts of the body. The brain is contained in, and protected by, the skull bones of the head. The cerebrum is the largest part of the human brain. It is divided into two cerebral hemispheres. The cerebral cortex is an outer layer of grey matter, covering the core of white matter [1].

White matter is composed of bundles, which connect various gray matter areas (the locations of nerve cell bodies) of the brain to each other and carry nerve impulses between neurons. Myelin acts as an insulator, which allows electrical signals to jump, rather than coursing through the axon, increasing the speed of transmission of all nerve signals. The other main component of the brain is grey matter (actually pinkish tan due to blood capillaries), which is composed of neurons. [2]. Cerebrospinal fluid is a colorless transcellular fluid that circulates the brain in the subarachnoid space, in the ventricular system, and in the central canal of the spinal cord[3].

Segmentation of brain tissues in MRI image has a number of applications in diagnosis, surgical planning, and treatment of brain abnormalities. However, it is a time-consuming task to be performed by medical experts. In addition to that, it is challenging due to intensity overlap between the different tissues caused by the intensity homogeneity and artifacts inherent to MRI. To minimize this effect, it was proposed to apply histogram based preprocessing. The goal of this project is to develop a robust and automatic segmentation of White Matter (WM) and Gray Matter (GM)) and Cerebrospinal Fluid (CSF) of the human brain.

To tackle the problem, we have proposed Convolutional Neural Network (CNN) based approach and probabilistic Atlas. U-net [4] is one of the most commonly used and best-performing architecture in medical image segmentation, and we have used both 2D and 3D versions. The performance was evaluated using Dice Coefficient (DSC), Hausdorff Distance (HD) and Average Volumetric Difference (AVD).

This report is organized as follows. In section 2, we briefly explain about provided dataset and metrics used to evaluate the performance of our implementation. Our approach and detail of implementation are described in section 3. Section 4 presents the discussion of segmentation results based on the metrics given. Finally, project management and conclusion are presented in section 5 and 6, respectively.

2 DATASET AND EVALUATION METRICS

2.1 Dataset

For this project, the proposed solutions will be evaluated on the well-known IBSR18 dataset (see figure 1) which is one of the standard datasets for tissue quantification and segmentation evaluation. The dataset consists of 18 MRI volumes including: ten volumes for training (red), five for validation (blue) and three for testing (yellow). For the training and validation images, the corresponding ground truth (GT) is provided, while for the testing set it will not be available.

By the end of this project, the results in this testing were submitted to perform a competition with all the other groups. The ranking will take into account the performance of the algorithm based on Dice (DSC), Hausdorff distance (HD) and average volumetric difference (AVD), as commonly defined in

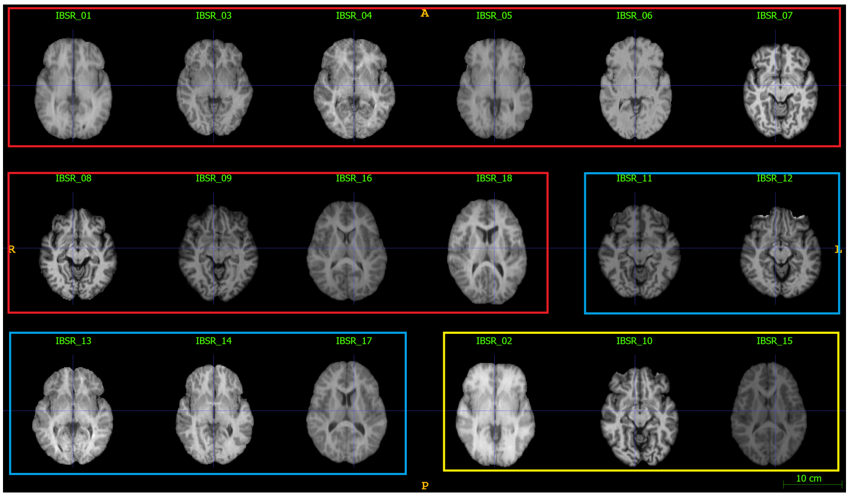


Figure 1

other challenges (e.g. MICCAI2012, MRBrainS13, and iSeg2017).

2.2 Evaluation metrics

In this project, three metrics are used to evaluate the proposed method: DSC, HD and AVD.

First, the DSC, also called the overlap index or F1 score, is the most used metric in the evaluation of medical volume segmentations. In addition to the direct comparison between automatic and ground truth segmentations, it is common to use the DSC to measure reproducibility (repeatability) [5]. DSC is computed by:

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

where TP, FP and FN are the numbers of true positive, false positive and false negative predictions for the considered class.

Second, the distance between crisp volumes (HD) between two finite point sets A and B is defined by:

$$\mathrm{HD} = \max(h(A,B), h(B,A)) \tag{2}$$

where $h(A,B)$ is called the directed HD and given by:

$$h(A,B) = \max_{a \in A} \min_{b \in B} \|a - b\| \tag{3}$$

Finally, the AVD is defined by:

$$\mathrm{AVD}(A,B) = \frac{100 \times \|A - B\|}{\sum A} \tag{4}$$

where A is groundtruth and B is predicted volume of one class.

3 Implementations

3.1 Proposed methods

Figure 2 shows the proposed approaches for our segmentation task. We have proposed CNN and Atlas-based approaches with preprocessing. Preprocessing consists of histogram matching and equalization, normalization and patch extraction. Histogram matching and equalization were used in both approaches, while normalization is applied in the atlas, and patch extraction for CNN approaches.

For CNN based approaches we have experimented with 2D U-net and 3D U-net. The CNN are trained and evaluated on overlapping patches extracted from the input volumes. For implementing the 3D U-net, we have adopted the code in GitHub repository https://github.com/ellisdg/3dUnetCNN. To reconstruct the final segmentation result from overlapping patch predictions, we have employed majority vote combining.

The overall architecture of 2D and 3D U-net that we have used in our project is similar to [4], and it is given in figure 3. The U-net in figure 3a and 3b consist of a contracting path and an expansive path path. The contracting path consists of the repeated application of two 3x3 (3 x 3 x 3 for 3D) convolutions (padding *same*), each followed by a rectified linear unit (ReLU) and a 2x2 (2 x 2 x 2 for 3D) max pooling operation with stride 2 for downsampling. At each downsampling, the number of feature maps is doubling. Every step of expansion path contains an upsampling of feature map followed by up-convolution that halves the number of feature maps and concatenation with the corresponding layer in contraction path. For 2D U-net (figure 3a) the input size is 32 x 32 and the output is 32 x 32 x 4, where the four channel corresponds to the background, CSF, GM, and WM. Similarly, 3D U-net based segmentation has an input shape of 32 x 32 x 32, while the output is 32 x 32 x 32 x 4.

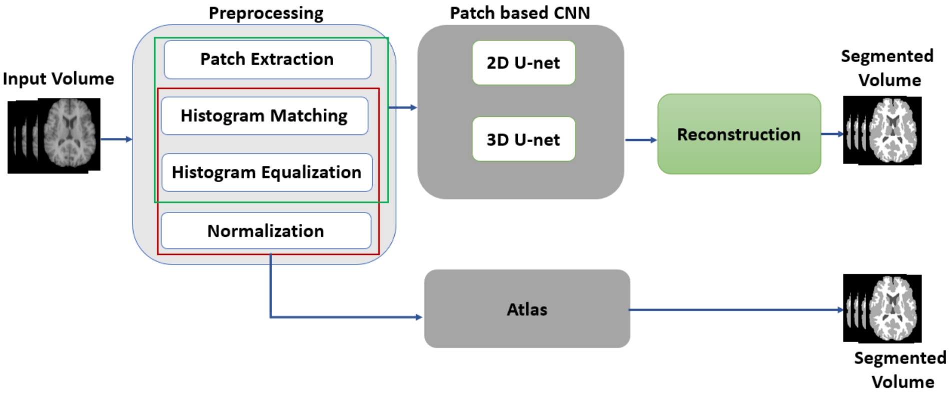
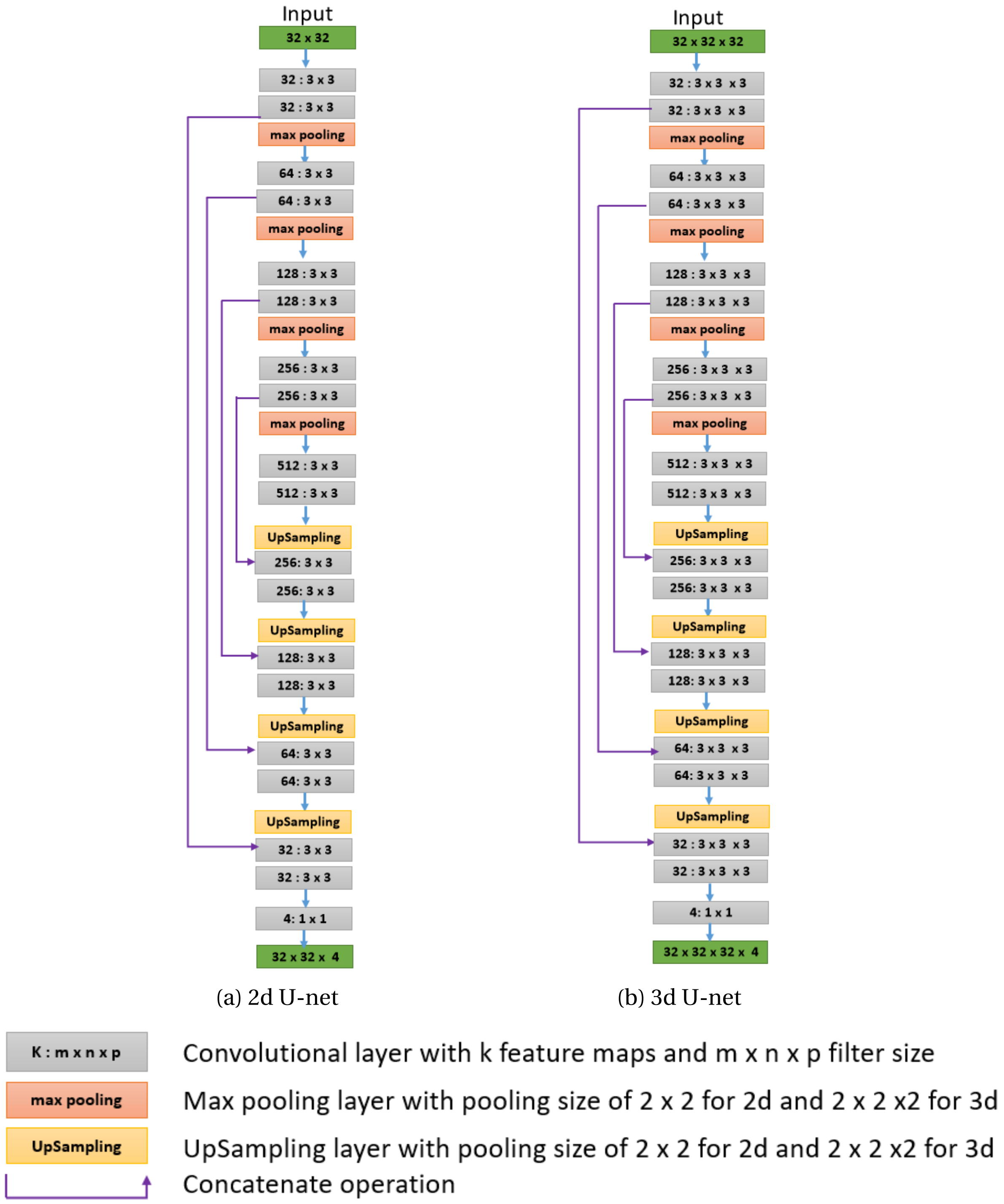


Figure 2: Proposed approaches: preprocessing, core implementation (CNN and atlas) and post-processing.

We have also proposed to try with probabilistic atlas. Atlas is a model for a group of images with parameters that are learned from a training dataset, and it refers to the pair of an anatomical image which is a template image and a probability tissue distribution volume(image space). The anatomical image is a reference image of the atlas, while probability tissue distribution is a volume which shows the spatial distribution of probabilities that a given voxel belongs to any of the classes or tissues under consideration. Moreover, we have built tissue model, which shows the intensity distribution of the three tissues.

In addition to atlas and U-net based architectures, we have tried with iSen2017 (codes found in the following GitHub repository https://github.com/ellisdg/3dUnetCNN) It gives good Dice coefficient on the patches during training, however, when we reconstruct the segmentation result, we have realized it is clipping bottom part of the volume. The problem was in all volumes.



(a) 2d U-net

(b) 3d U-net

Figure 3: Multiclass U-net architectures

3.2 Preprocessing

As mentioned in section 3, we have adopted U-net (2D, 3D and a customized 3D model) [4] for this project.

We observe that the intensities of the provided dataset are not standardized (see figure 1): some volumes lack of contrast, for example, volumes IBSR_05 and IBSR_15, some volumes are very dark (IBSR_11 and IBSR_12). This problem makes the dataset very challenging to segment into three classes CSF, GM, and WM. Hence, preprocessing techniques, that normalize to bring each volume into a range that similar or standard to the distribution of the whole dataset, are needed. Later in section 4, we will compare the performance of our best approach using preprocessed dataset and original one to demonstrate the importance of data preprocessing.

To standardize the dataset, we use two methods:

- First method: normalization,
- Second method: normalization to 0-1, Contrast-limited Adaptive Histogram Equalization (CLAHE) and histogram matching.

Note that the normalization techniques in the first and second methods are different.

In the first method, we apply a standard normalization technique: minus each voxel values by the mean and divide by the standard deviation.

In the second standardization method, we begin by selecting one volume in the training set. Then, we normalize this volume in the range of 0 and 1. Next, we apply CLAHE on this normalized volume and set the result to the reference volume. Finally, we apply histogram matching for the whole dataset to this reference volume. Figure 4 shows some slice 128 of some volumes after applying the second method

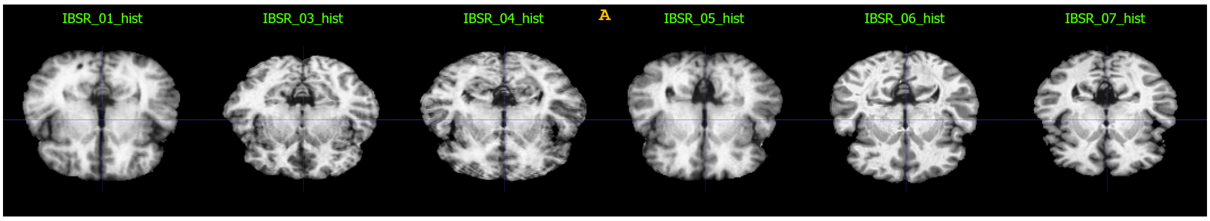


Figure 4: Slice 128 of six standardized volumes using the second method

3.2.1 Normalization

There are two types of normalization that we have used in this project.

The first type, which is used in the first method mentioned above, is to bring the mean intensity of a volume to 0 and divide each of the voxels in that volume by the standard deviation. It can be formulated as the following:

$$V_{new} = \frac{V_{old} - \text{Mean}}{\text{Standard Deviation}} \tag{5}$$

The second type of normalization is to normalize the intensity of each of the volumes to 0 (minimum) and 1 (maximum). This technique is very straightforward and can be seen in the following equation:

$$V_{new} = \frac{V_{old} - \text{Min}}{\text{Max-Min}} \tag{6}$$

3.2.2 Contrast-limited adaptive histogram equalization

CLAHE is an image-enhancement technique which aims to enhance the contrast using Adaptive Histogram Equalization (AHE) in its contrast limiting. CLAHE is designed to prevent the over-amplification of noise that AHE can give rise to. CLAHE limits the amplification by clipping the histogram at a predefined value before computing the CDF [6].

In this project, we adopt CLAHE to enhance the contrast of a selected volume, and set the enhanced volume as the reference volume to apply histogram matching in the whole dataset.

3.2.3 Histogram matching

In image processing, histogram matching is the transformation of an image so that its histogram matches a specified histogram. The goal of histogram matching is to normalize two images when the images were acquired at the same local illumination (such as shadows) over the same location, but by different sensors, atmospheric conditions or global illumination [7].

In this project, we adopt histogram matching as the last step in the second standardization method.

3.2.4 Mask

The purposes of the masks is to filter out unwanted information and keep only the essential one. We use masks mainly for the test set for the following two purposes:

- To eliminate patches (2D and 3D) containing only background voxels to reduce segmentation runtime,
- To keep only voxels lying inside the mask and set the rest to 0 (background) after the segmentation. The reason is to reduce the error made by U-net. For instance, U-net might predict a background voxel as CSF, GM or WM. With a mask, we could minimize such error.

Four steps in figure 5 demonstrate how the masks are found in each slices:

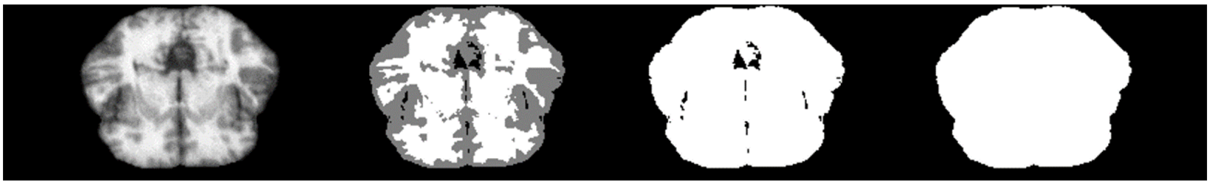


Figure 5: Four steps in mask extraction

- Step 1: extract slice by slice in a volume and process on it.
- Step 2: cluster this slice by k-means when k=3.
- Step 3: set regions contains the brightest cluster and the darkest cluster as background, and set the middle one as foreground.
- Step 4: fill holes to find the mask.

3.3 Training

In this project, we adopted U-net for the segmentation task. The CNN architectures were discussed before . In this section, we will discuss the CNN training. For 3D U-net and iSén7 model, we adopted codes from the following link: Keras 3D U-Net Convolution Neural Network (CNN) designed for medical image segmentation.

3.3.1 Technical details

The proposed method has been implemented in the Python language, using Keras with Tensorflow backend. All experiments have been run on a GNU/Linux machine box running Ubuntu 16.04, with 32 GB RAM. CNN training has been carried out on a single TITAN-X GPU (NVIDIA Corp, United States) with 12 GB RAM.

3.3.2 Patch extraction

For training our network, we extract 2D and 3D patches for the 2D and 3D U-net, respectively, from the training set. When training the CNN, it is important to take into account how the training samples are extracted from an image and what the patch-size is.

For the extraction, random selection of a certain number of samples from an image is one of the common techniques in the literature. However, in this project, we proposed to use two different techniques:

- Extract and keep all patches with a specific extraction step.
- Extract more patches consisting of CSF voxels by reducing the extraction step. The reason is due to the unbalance of three classes.

Here, we used *extract_patches* of *sklearn* package to extract patches. We observed a memory-handling problem of this function. In specific, the runtime to extract, for example, 2000 3D patches, could take 5-6 times longer than to extract 1000 3D patches. Hence, we came up with a solution for this kind of problem. In specific, we can divide one volume into sub-volumes. Next, we extract patches on each sub-volumes and stack them by the end.

About the patch-size, we have tried with different values: 16, 32, 64. Finally, patch-size at 32 gave us the best performance for both 2D and 3D U-net.

3.3.3 Number of classes selection

To the best of our knowledge, most of research works on this dataset work with three classes. It means that, they train extracted patches on three classes to return a model which predict the probabilities of three classes on each voxels. However, there is one issue of this method: how to find the optimized threshold to predict a background voxel?

Some works set the threshold at 0.5; some works set it higher or lower. To overcome such obstacle, we proposed to train on four classes: CSF, GM, WM, and Background. It works perfectly in the end.

3.3.4 Loss function

We started off by employing dice coefficient loss function. Here, DSC is computed for all of four classes. However, after some trials we observed that the DSC for CSF is very poor compared to GM, WM and

Background. The reason is due to the small number of CSF. That is CSF has less weighting than the rest.

There are two solutions for this problem. First, we can adopt categorical cross-entropy which a built-in loss function of Keras. Second, we need to derive a loss function that gives the CSF class the same weight.

Based on the goal of this project is to find a method which returns the highest average DSC of three tissues, we made our own loss function as following:

$$L(y,\hat{y}) = 4 - \sum_{i=0}^{3} \text{DSC}(y_i,\hat{y}_i) \tag{7}$$

where y_i and $\hat{y}_i$ are predicted and groundtruth for class i, respectively.

3.3.5 Segmentation reconstruction

After training, we obtain the final model. To segment a volume, we apply found model and it is very straightforward.

First, we extract patches from each volume where extraction-step is equivalent to the patch size. In this case, we set both of U-net architectures to 32. It means that for a volume with the dimension of $256 \times 128 \times 256$, we can extract 256 3D patches or 8192 2D patches. Next, we keep only patches which overlap with the mask (see section 3.2.4). After that, we stack the remaining patches in a list and apply the pre-trained model to obtain the probabilistic map which has the size of $256 \times 128 \times 256 \times 4$ (where $256 \times 128 \times 256$ is the size of a volume and 4 is the number of classes). Finally, for each voxel, we find the maximum probability among four classes and return the corresponding label.

3.3.6 Experimental details

In the given dataset, skull-stripping and bias field correction algorithm were already applied. Thus, we do not need to apply these techniques.

In our experiments, we trained and validated a single model using the available training and validation sets of ten and five volumes, respectively, while we tested on three volumes. From the training set, we extracted around 110k of size 32x32 and 12k of size 32x32x32 patches for 2D and 3D U-net, respectively. Similarly, for validation set, around 50k 2D and 5k 3D patches were extracted.

Next, the extracted patches were passed to the network for training in batches of size 32 and 16 for 2D and 3D U-net, respectively. The network was trained for 500 epochs, while to prevent the network from overfitting, we applied early stopping of the training process. The training process was automatically terminated when the validation accuracy did not increase after 30 epochs.

Overall, 3D and 2D U-net took around 2 and 4 hours for training, respectively, and one epoch with the batch size of 16 lasted for 1.5 minutes. Segmentation runtime is 40-50 and 15-25 seconds for 3D and 2D U-net, respectively. Thus, methods are fast enough to be used in clinical practice.

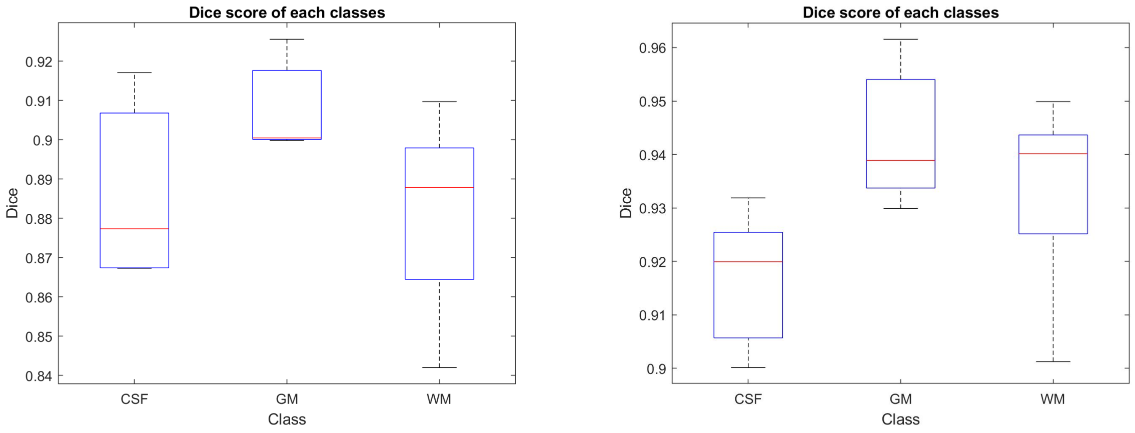
4 Results and Discussion

Figure 6 shows the the mean DSC, HD and AVD values of four methods including: Atlas-based segmentation, 2D U-net using our proposed loss function [1], 3D U-net using DSC as loss function [2] and 3D U-net using our proposed loss function [3]. Note that, all of the methods in this table adopting the same strategies for preprocessing and training except Atlas-based segmentation.

Model	DSC			HD			AVD		
	CSF	GM	WM	CSF	GM	WM	CSF	GM	WM
Atlas	0.686±0.065	0.872±0.035	0.771±0.067	33.658±1.343	**9.289±1.506**	12.633±2.9	49.061±6.699	25.699±7.664	44.67±13.286
2D U-net [1]	0.867±0.02	0.902±0.01	0.91±0.023	32.601±4.91	14.859±1	13.076±2.055	22.144±3.378	17.428±1.79	25.229±5.925
3D U-net [2]	0.823±0.018	0.937±0.012	0.928±0.024	41.984±4.913	14.722±0.95	11.769±1.826	24.24±3.066	15.909±2.112	22.827±5.974
3D U-net [3]	**0.917±0.011**	**0.943±0.01**	**0.933±0.017**	**17.879±7.07**	12.749±2.5	**8.718±1.184**	**16.805±2.107**	**11.044±2.197**	**13.538±3.893**

Figure 6: Comparison among our methods on IBSR18 dataset in terms of DSC, HD, and AVD. The average values show mean DSC for the presented structure DSC scores.

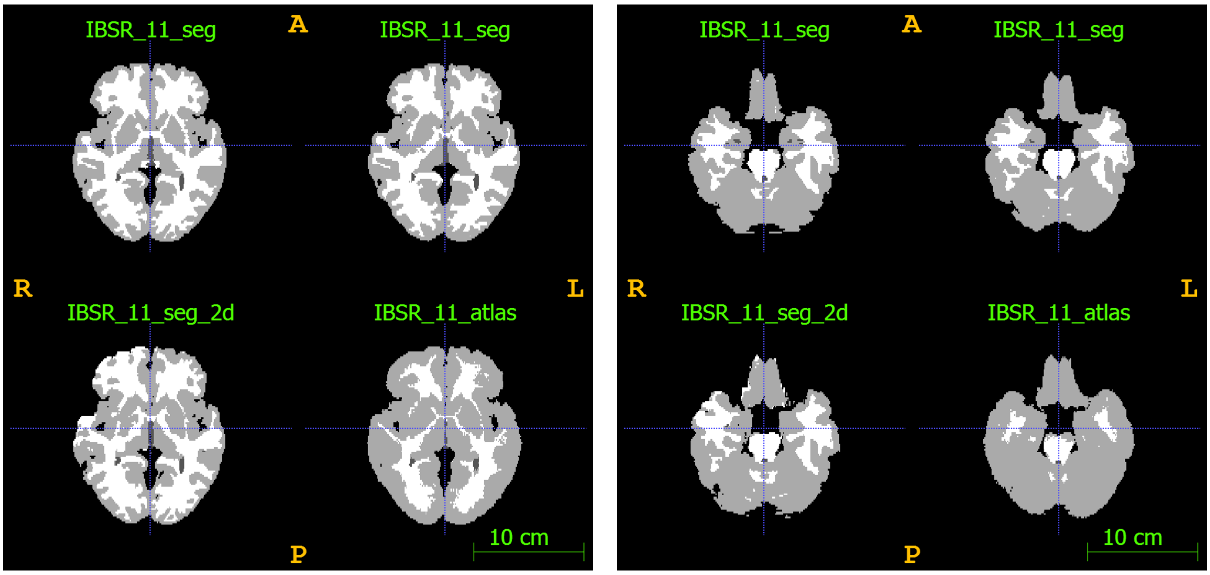
As can be seen in figure 6 the best method is 3D U-net using our proposed loss function which has mean DSC of 0.917, 0.943 and 0.933 for CSF, GM, and WM, respectively. In term of HD, our best method showed the overall mean of 12.7, whereas AVD yielded 13.3. The proposed strategy outperformed 3D U-net with dice loss function and 2D U-net and was significantly better than Atlas-based segmentation using the only normalization from 0-1. Thus, we can integrate Atlas-based as a post-processing technique with the purpose of improving HD score for CNN or combine spacial features, provided by tissue atlas probabilities, with CNN for the segmentation in MRI [8].



(a) (b)

Figure 7: Box plot of (a) 2D U-net and (b) 3D U-net

Figure 7 shows the box plots of our two best-proposed methods (2D U-net and 3D U-net). As can be seen in figure 7, all classes of 3D U-net performed better than 2D U-net. The median DSCs of 3D U-net are 0.92 0.94 and 0.94 for CSF, GM and WM, respectively, while the median DSCs of 2D U-net are 0.88, 0.9 and 0.89, respectively. The maximum DSC values of 3D U-net are also much better than of 2D U-net.



(a) (b)

Figure 8: Segmentation results by our proposed method on the IBSR 18 dataset. Two slices of brain images are shown in two figures. In each figure, from the left to right and top to bottom is the groundtruth, 3D U-net, 2D U-net and Atlas based segmentation, respectively.

A few distinct typical brain images and the corresponding tissue regions segmented by our method are illustrated in figure 8. From the ground-truth and input images, we notice that these brain tissue regions have very complicated appearances and great variations at different slices, which can make the segmentation task very challenging. Moreover, the intensity values of voxels from separated brain tissues overlap considerably. For this reason, it may create mistakes on the classification methods based on the intensity information. Leveraging advantages of CNN, our proposed method can efficiently segment the brain regions even for voxels lying at the borders of tissues.

We observe in figure 8 that the first 3D U-net gave the smoothest and best results. 2D U-net also performed well, but due to the lack of information with other voxels along slice, it can not be compared to 3D one in term of evaluation score. Atlas technique performed the worst, especially for WM.

5 Project Management

To finish the project with in the given time frame, we scheduled a proper plan to meet the deadlines. The schedule of our meetings and an internal progress report is given in the following Table 1. All the deadlines were decided by mutually discussing and selecting the best approach scenario, and making sure to have a progress in every lab session. An internal progress report was also maintained which was updated daily based on the work done or problem faced and that needs to be solved. We shared also a sharelatex to write report.

Table 1: Internal progress report

Name	Work Assigned	Problems Faced	Next step
Week 1 and 2			
Both	Preprocessing	No	Testing preprocessed image
Both	Atlas based and set up CNN	No	Experimenting with CNN
Week 3 and 4			
Minh	Experiment 2D U-net 4 classes	Accessing server	Training
Yeman	Experiment 2D U-net 3 classes	Accessing server	Training
Week 5			
Both	Experiment 3D U-net 4 classes	No	Update report
Minh	iSen2017	Not smooth result	Look for solution
Week 6 and 7			
Both	Compute other metrics	No	Finalize codes
Minh	Code commenting	No	Finalize report
Both	Report finalizing	No	Submitting report

6 Conclusion

In this project, we have developed brain tissue segmentation method of MRI images. As there is intensity overlap between the different tissue, and intensity heterogeneity, to minimize these effect we preprocessed the images using histogram-based approaches and intensity normalization. We have experimented and evaluated atlas, 2D, and 3D U-net based methods, and the performance was evaluated quantitatively using DSC, HD and AVD. Overall, we have found that 3D U-net with our proposed loss function performed better than the others in all of the evaluation metrics for all tissue types, except HD for GM.

During the project, we have tried with several of the preprocessing techniques used in deep learning for medical image segmentation; however, due to time constraint we did not experiment with post-processing methods to further smooth the segmentation results and can be considered as possible future work.

By the end of the project, our approach outperformed the other groups which lead us to the win of the MISA challenge.

References

[1] John Kiernan and Raj Rajakumar. *Barr's the human nervous system: an anatomical viewpoint*. Lippincott Williams & Wilkins, 2013.

[2] Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven A Siegelbaum, A James Hudspeth, et al. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.

[3] Henry Gray, Susan Standring, Neel Anand, Rolfe Birch, Patricia Collins, AR Crossman, Michael Gleeson, Girish Jawaheer, Ariana L Smith, Jonathan D Spratt, et al. *Gray's anatomy: the anatomical basis of clinical practice*. Elsevier, 2016.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedi-

cal image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[5] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15(1):29, 2015.

[6] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.

[7] Dinu Coltuc, Philippe Bolon, and J-M Chassery. Exact histogram specification. *IEEE Transactions on Image Processing*, 15(5):1143–1152, 2006.

[8] Mohsen Ghafoorian, Nico Karssemeijer, Tom Heskes, Inge van Uden, Clara Sanchez, Geert Litjens, Frank-Erik de Leeuw, Bram van Ginneken, Elena Marchiori, and Bram Platel. Location sensitive deep convolutional neural networks for segmentation of white matter hyperintensities. *arXiv preprint arXiv:1610.04834*, 2016.