University of Girona

Medical Imaging and Applications (MAIA)

Medical Image Segmentation and Applications (MiSa)

Final Project Report on:

Brain Tissues Segmentation using Deep Learning

BY: Tewele Weletnsea Tareke

ID No: u1970634@campus.udg.edu

Aroj Hada

ID No:u197064@campus.udg.edu

Submitted To: Dr. Sandra Gonzalez-Villa

Dr. Arnau Oliver

1. Introduction

Brain imaging is important for the diagnosis of brain-related diseases such as neurological disease (Parkinson's disease), neurodegenerative disease (Alzheimer's syndrome), and brain tumors. According to the American Cancer Society and the National Cancer Institute Report, brain and nervous system cancer is the tenth most common cause of death for both genders. Recently, a drastic increase in the number of brain disorders has been noted. This has indirectly led to an increased demand for automated brain segmentation solutions to assist medical experts in early diagnosis and treatment interventions. The main challenge of this project is to develop deep learning approaches for the segmentation of brain tissues. Accurate and consistent segmentation of target brain regions or tumors from the surrounding tissues using the MR images is crucial for clinical evaluation of disease progression, surgical planning, post-surgical matching, and radiation therapy outcomes. These segmentations are useful for measuring and visualizing anatomical structures, but also to analyze brain changes in case of diseases like Alzheimer [progressive neurologic disorder that causes the brain to shrink (atrophy) and brain cells to die.

2. Problem Statement

Brain image segmentation is one of the most time-consuming and challenging procedures in a clinical environment. Today different automatic segmentations are available like Freesurfer and ANTs Segmentation. .But these approaches are often need less accurate) and require additional manual segmentations which are both time consuming and challenging. Since the manual segmentation of the brain tumors and related small brain structures is laborious and time-consuming, hence it is very important to have automated solutions with high accuracy. In most segmentation and detection process white matter (WM) and Gray Matter (GM) give promising result. But Cerebrospinal Fluid (CSF) not detected well and needs more improvement.

3. Dataset

The proposed solutions have trained and evaluated on the well-known Internet Brain Segmentation Repository (IBSR18) dataset. This is one of the standard datasets for tissue quantification and segmentation evaluation. We used 18 MRI volumes of T1_w images. Every image consists of a finite set of image elements called voxels in 3D space. Each image element is uniquely specified by its intensity value and its coordinates (**i, j, k**), where **i** is the image row number, **j** is the image column number, and **k** is the slice number in a volumetric stack; We used to split the original dataset into: training (ten volume, validation (five volume), and testing (three volume).And also we have used data augmentation using ImageGenerator Method (discussed below in detail).

.

Training Subset		
Volume Name	Volume	Spacing(mm)
IBSR 01,03,04,05, 06	256 x128 x256	0.94 x 1.5 x 0.94
IBSR 07,03,08, 09	256 x128 x256	1 x 1.5 x 1
IBSR 16, 18	256 x128 x256	0.84 x 1.5 x 0.84
Validation Subset		
Volume Name	Volume	Spacing(mm)
IBSR 11, 12	256 x128 x256	1 x 1.5 x 1
IBSR 13, 14	256 x128 x256	0.94 x 1.5 x 0.94
IBSR 17	256 x128 x256	0.84 x 1.5 x 0.84
Test Subset		
Volume Name	Volume	Spacing(mm)
IBSR 02	256 x128 x256	0.94 x 1.5 x 0.94
IBSR 10	256 x128 x256	1 x 1.5 x 1
IBSR 15	256 x128 x256	0.84 x 1.5 x 0.84

Table 1: summary on the original IBSR18 dataset used in this paper

4. MRI Pre-Processing

The task of segmenting brain tissues become more challenging as different scanners are used with different parameters during acquisition. That usually leads to intensity heterogeneity, contrast variations, and different types of noise [1].Therefore, data homogenization is often necessary. Hence we did bias field correction, normalization and extracting important patches as preprocessing steps on the MR images for segmentation.

4.1) Standardization: The input volumes were standardized to zero mean and unit standard deviation using the volume statistics as mentioned in the reference paper of Rajchl et al. [2]. It can be formulated as equation 1.formulated as equation_1

Equation_1 $$V_{new} = \frac{Vold - \mu}{\sigma}$$

4.2) Bias field correction (Intensity Distribution in Brain MRI): The intensity of brain tissue is one of the most important features for brain MRI segmentation. The bias field, also called the intensity inhomogeneity, is a low-frequency spatially varying MRI artifact causing a smooth signal intensity variation within tissue of the same physical properties; see Figure 1. The bias field arises from spatial inhomogeneity of the magnetic field, variations in the sensitivity of the reception coil, and the interaction between the magnetic field and the human body.

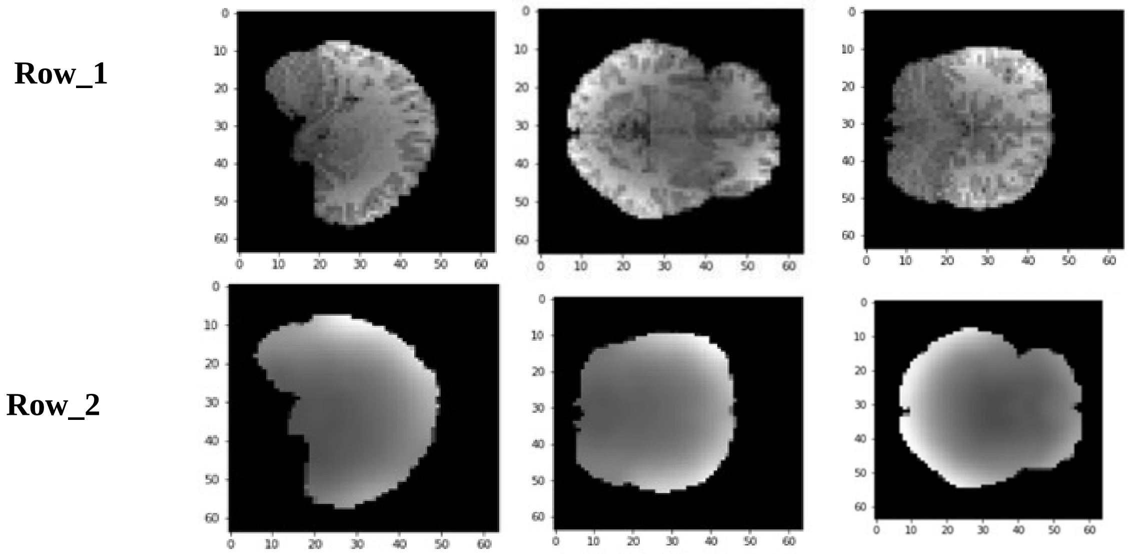
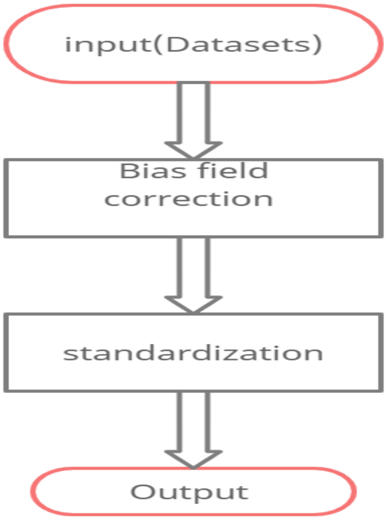


Figure_1: Bias field correction as Preprocessing steps: (Row_1) the original T1-W MR images), (Row_2) the brain tissue image after bias field correction

Figure_2: Preprocessing pipeline I



4.3) Data Augmentation: is a technique that can be used to artificially expand the size of a training set by creating modified data from the existing one. It is a good practice to use data augmentation to prevent overfitting, or if the initial dataset is too small to train on, or even if you want to squeeze better performance from your model. We have generated different number of synthetic images (depends on the batch size used) in each iteration using ImageDataGenerator **[7]** and lets us augment our images in real-time while your model is still training. This will not only make your model robust but will also save up on the overhead memory tough the model trained with synthetic images not perform well.

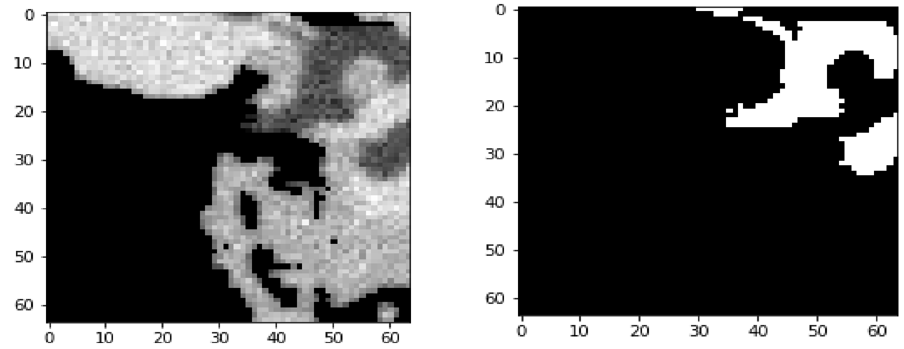


Figure_3: Synthetic training patches of 64 x 64 size with its CSF label

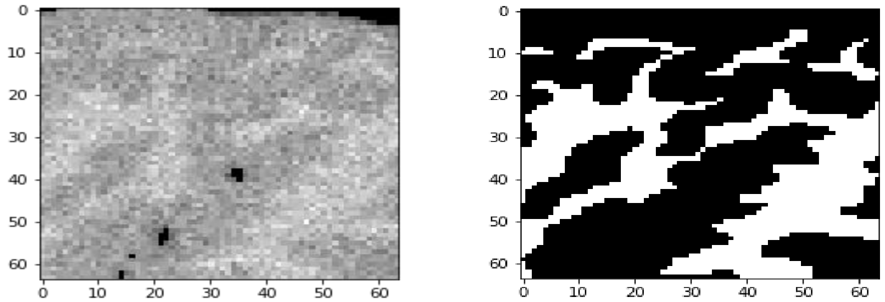


Figure_4: Synthetic training patches of 64 x 64 size with its white matter (WM) label

4.4) Resampling Dataset: The original datasets used for this segmentation were size of 256 x 128 x 256 and different spacing among them as shown in table 1. We have done resampling only on the training datasets to make our model robust by resizing the image dimension and changing the space among the voxels in to equal size.

Training Subset		
Volume Name	Volume	Spacing(mm)
IBSR 01,03,04, 05, 06	240 x 192 x 240	1 x 1 x 1
IBSR 07,03,08, 09	256 x 192 x 256	1 x 1 x 1

IBSR 16, 18	214 x 192 x 214	1 x 1 x 1

Table 2: Resampled IBSR18 training datasets

5. Patches Extraction

The main goal of this step is to extract the useful patches and we want to provide the model with meaning information instead of feeding random inputs. Hence, we have extracted different number of useful patches from volumes and let the network train on the important patches. Most importantly, 13584 train patches with 64 x64 size are extracted with their corresponding ground truth to train the final model (used for prediction) and 4624 validation patches with same size of the train patches are extracted to validate the model.

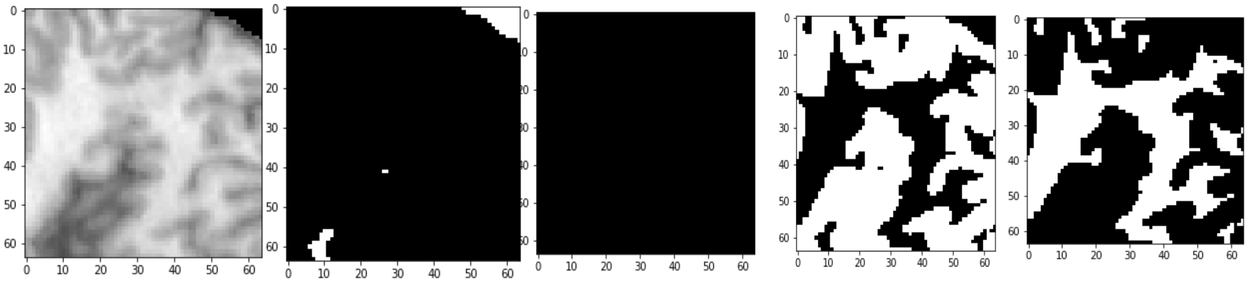


Figure_5: The above image indicates, the 152th extracted train patches with it labels (image, background =0, CSF =1, GM=2, WM =3) from left to right respectively.

As we can see from the above extracted image patches, there is label or ground truth for CSF that indicates the extracted image only contain white and gray matter tissues.

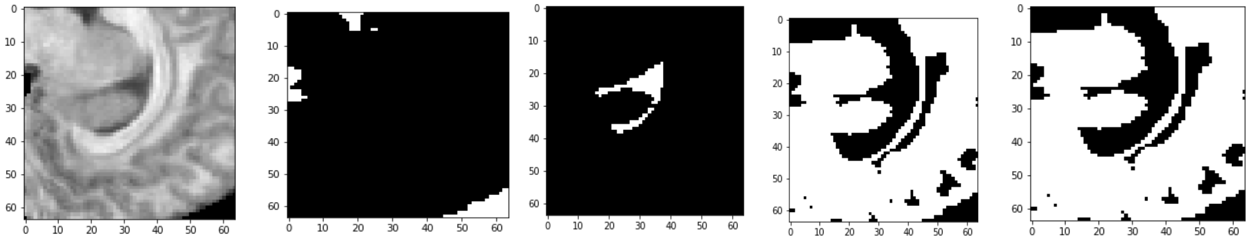


Figure_6: The above image indicates, the 1930th extracted validation patches with it labels (image, background =0, CSF =1, GM=2, WM =3) from left to right respectively.

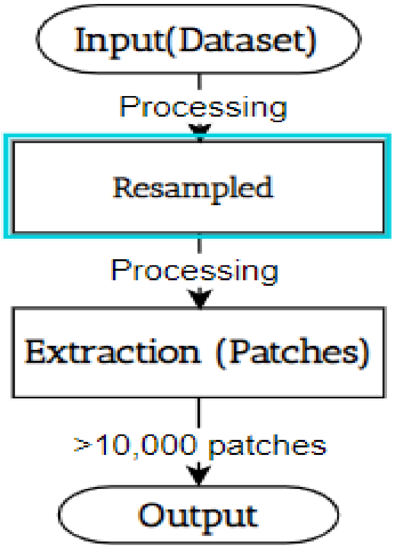


Figure_7: Preprocessing pipeline II

6. Methodology

6.1) **Network Architecture:** For implementation of the project we have used two different deep learning architecture and we get a chance to compare and contrast the result of these network methods.

6.2) First Approach: U-Net Model Implementation:

The U-net is a convolutional network architecture used for fast and precise segmentation of images. This is a very popular architecture in biomedical images. Basically U-net is SegNet network with skip connections. The architecture contains two paths: A contraction path => **Encoder**, an Expanding path: **Decoder**

- The **encoder** is used to capture the context in the image, whereas the **decoder** will enable precise localization. **Figure_6** shows the architecture of the model.
- We have used four layer of UNet and SegNet deep learning architecture with different kernel size and we got very promising result for the tissue segmentation of the brain.

6.3) Second Approach: SegNet(Encoder-Decoder Network) Model Implementation:

SegNet is a semantic segmentation model. This core trainable segmentation architecture consists of an encoder network, a corresponding decoder network followed by a pixel-wise classification layer. The architecture of the SegNet network is topologically identical U-net network. The only difference is SegNet implemented without skip connection or concatenation. The role of the decoder network is to map the low resolution encoder feature maps to full input resolution feature maps for pixel-wise classification. We have used SegNet model (**in figure_8** without concatenation path) with different layers and the result was improved significantly comparing with U-net architecture.

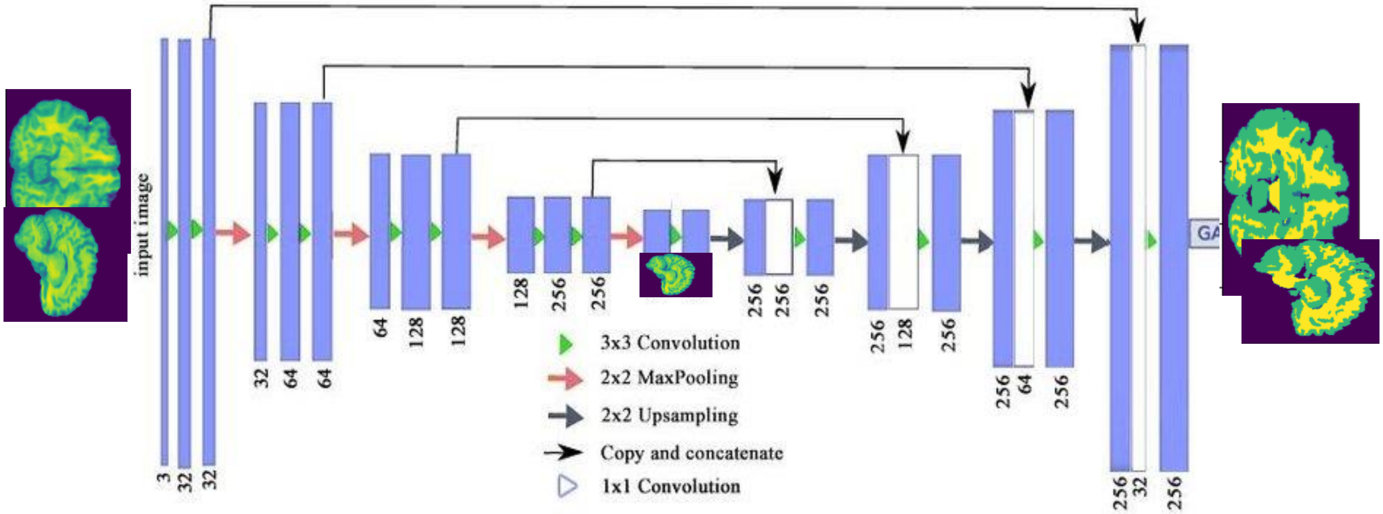


Figure_8: U-net Deep Learning Architecture

7. Environment

To train/validate/test the NeuroNet. The specification of the used GPU is mentioned in the following Table 2.

Environment/ tool used	
Item Name	Specification
Editor/Colab	Colab Pro
Keras/tensor flow version	2.7.0
GPU	NVIDIA-SMI 495.44
Memory	high-RAM
Driver Version(GPU)	460.32.03

Table 3: Summary of environment used

8. Experiments and Results

In this project we have done five experiments and got different result accordingly. The **quantitative** and **qualitative** result of three out of the five experiments are explained below in detail. Hausdorff distance (HD), average volumetric difference (AVD) and Dice Similarity Coefficient (DSC), were used for evaluating predicted labels and guiding the tuning process. Our main idea of the experiments was: firstly tune the model perfectly on the data with default parameters suggested in [3], then tune the parameters to improve the results. Here we discuss three main experiments that we have found to be meaningful. We carried out many other experiments which we didn't include them here. The different experimental result are based on fine tuning of various model, image generation, and bias field correction and standardization process.

8.1 **Evaluation metrics for 3D image segmentation**: we have a set of metrics to measure the validation of the 3D image segmentation and we have used three different kind of metrics to measure our validation results. **The Dice coefficient** [4] (*DICE*), also called the overlap index, is the most used metric in validating medical volume segmentations. In addition to the direct comparison between automatic and ground truth segmentations, it is common to use the *DICE* to measure reproducibility (repeatability). **Hausdorff Distance (*HD*)** is distance between crisp volumes or between two finite point sets A and B is defined by HD (A, B) = Max (h (A, B), h (B, A)) and The Average Distance, or the Average Hausdorff Distance (*AVD*), is the *HD* averaged over all points. **Volumetric difference (*VD*)** is a measure that considers the volumes of the segments to indicate similarity by calculating the difference volumetric difference. There is more than one definition for the volumetric distance, but we consider the definition in [6], namely the absolute volume difference divided by the sum of the compared volumes. We define the Volumetric Similarity (*VS*) as $1-VD$.

8.2 **Experiment_1:** Firstly, we have trained our model with the processed data that went through only Standardization (0 mean and Unit standard deviation) mentioned in section 4.1. The model was trained from scratch but in the case of training deep models, enough care needs to be taken to initialize the parameters as shown in [4]. The following table shows the detailed result we got for the five validation volumes.

Model	Scan ID	CSF	GM	WM	Mean(DSC)
3D UNet **Patch size 64**	IBSR_11	0.653	.772	.785	**0.745**
	IBSR_12	0.711	0.752	0.707	**0.725**
	IBSR_13	0.458	0.756	.695	**0.636**
	IBSR_14	0.273	0.812	0.754	**0.613**
	IBSR_17	0.000	0.825	0.722	**0.515**

Table 4. Performance (in Dice Coefficient) on validation subset

Model	Scan ID	CSF	GM	WM	**Average(HS)**
3D UNet **Patch size 64**	IBSR_11	35.6	25.0	44.3	**34.8**
	IBSR_12	40.4	33.7	26.0	**33.3**
	IBSR_13	50.5	28.3	31.0	**36.6**
	IBSR_14	44.6	23.0	30.0	**32.5**
	IBSR_17	95.8	22.2	29.6	**49.2**

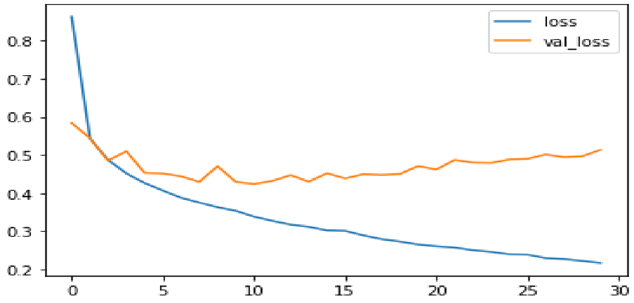
Table 5. Performance (in Hausdorff distance (HD)) on validation subset (**Distance between Segmented image and ground truth by number of voxels**)

Model	Scan ID	CSF	GM	WM	Average(VD)
3D UNet Patch size 64	IBSR_11	0.502	0.263	0.654	**0.476**
	IBSR_12	0.304	0.117	0.459	**0.234**
	IBSR_13	0.352	0.230	0.301	**0.290**
	IBSR_14	0.457	0.207	0.278	**0.321**
	IBSR_17	0.902	0.334	0.415	**0.550**

Table 6. Performance (average volumetric difference (AVD) on validation subset

As we can see from the above results, this model is not performing well and following figure shows the overall loss function of the model.

Figure_9: loss Function of Train and validation for Experiment_1



As we can observe from the above picture, the validation loss is increasing substantially and we have used early stopping method to control training process early.

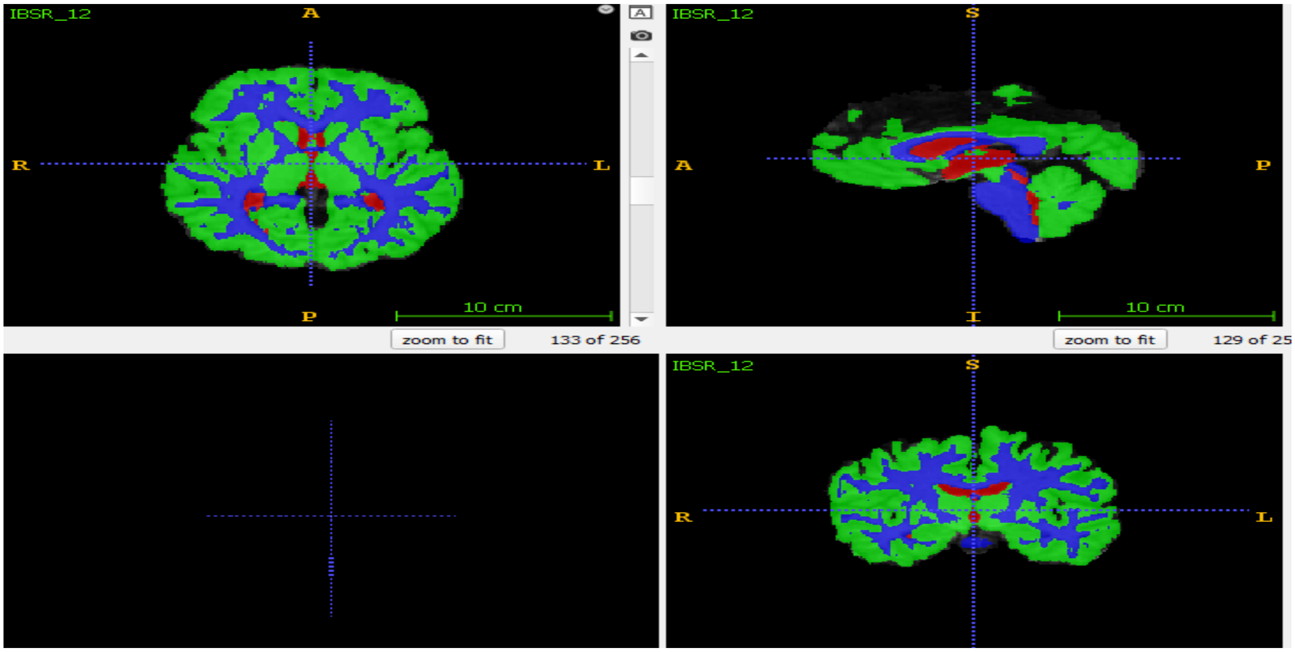


Figure_10: Qualitative result of the Experiment_1 for a single slice of IBSR_12 validation dataset

8.3) Experiment_2: To improve the result that we have seen in the above table or with the initial pre-processing pipeline (Standardization), we have used other preprocessing methods which are bias field correction (**section 4.2**) and synthetic images (**Section 4.3**). We have tried to balance the intensity inhomogeneity using biased bias field correction and increase the performance of the model using synthetic images. We trained different model with different hyper-parameters such as changing the number of training steps and number of patches. From Table 6, 7, 8 it can be seen that how prediction or the segmentation by model is improved comparing with previous result.

Model	Scan ID	CSF	GM	WM	Mean(DSC)
3D U-SegNet **Patch size 32**	IBSR_11	0.747	.815	.795	**0.785**
	IBSR_12	0.689	0.790	0.805	**0.761**
	IBSR_13	0.558	0.825	.745	**0.709**
	IBSR_14	0.612	0.832	0.801	**0.748**
	IBSR_17	0.335	0.840	0.790	**0.655**

Table 7. Performance (in Dice Coefficient) on validation subset with bias field correction

Model	Scan ID	CSF	GM	WM	**Average (HS)**
3D U-SegNet Patch size 32	IBSR_11	22.4	17.5	20.3	**20.0**
	IBSR_12	27.7	22.4	19.3	**23.1**
	IBSR_13	36.6	13.3	21.5	**23.8**
	IBSR_14	41.0	15.5	17.0	**24.5**
	IBSR_17	55.6	10.5	21.1	29.0

Table 8. Performance (in Hausdorff distance (HD)) on validation subset bias filed correction: **Distance between Segmented image and ground truth by number of voxels**

Model	Scan ID	CSF	GM	WM	Average (VD)
3D U-SegNet Patch size 32	IBSR_11	0.302	0.183	0.222	**0.235**
	IBSR_12	0.365	0.217	0.209	**0.263**
	IBSR_13	0.308	0.165	0.237	**0.236**
	IBSR_14	0.354	0.128	0.200	**0.227**
	IBSR_17	0.432	0.166	0.229	**0.299**

Table 9. Performance (in average volumetric difference (AVD)) on validation subset wit bias field correction

In this step, we didn't include the standardization preprocessing techniques. We have done bias field correction for the training subset and validation subset, but we have generated synthetic images for the training subset.

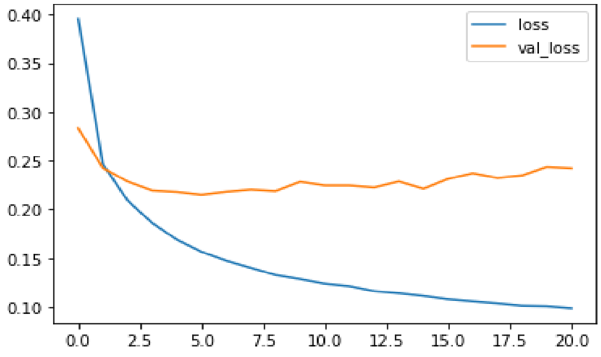


Figure_10: Loss function of train and validation Experiment_2

Though we did some preprocessing process on the dataset before the train the model, we didn't get a promising result or a model that perform/predicts well. We can mention many reasons why the model is not performing well, but the main reason is we are losing the main information or useful patches due to the preprocessing.

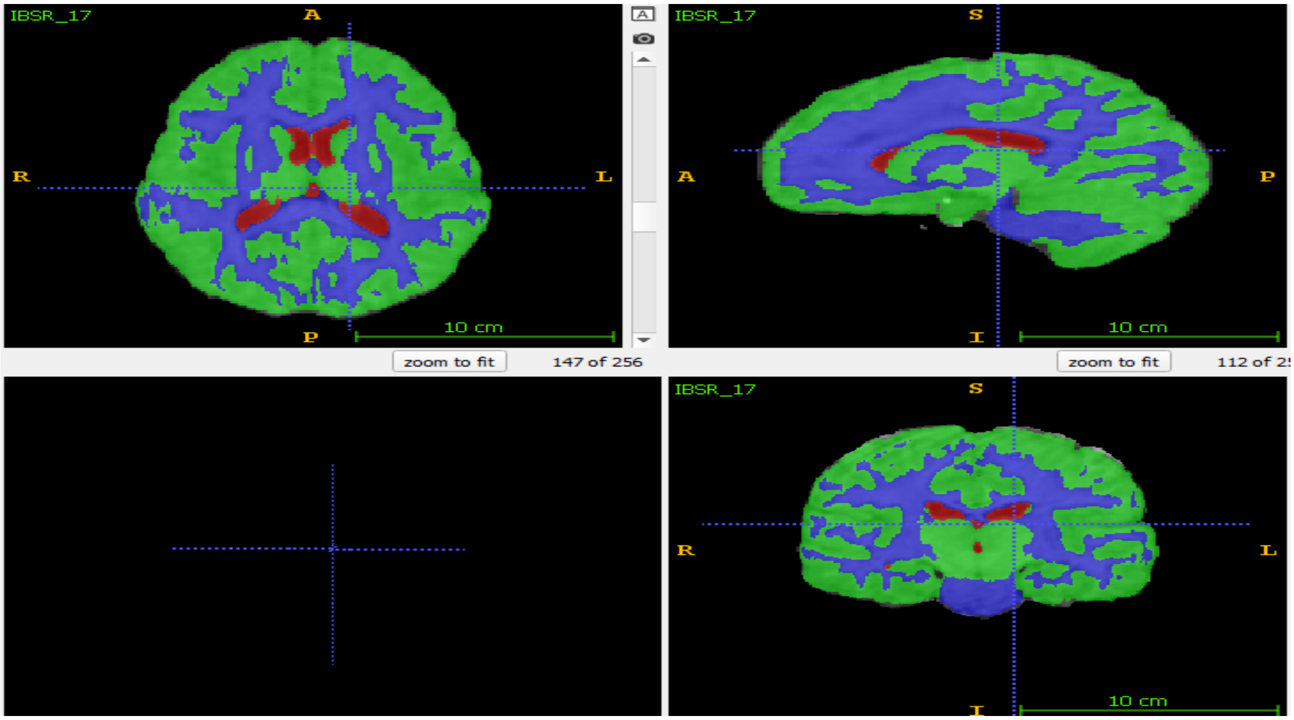


Figure_11: Qualitative result of Experiment_2 for a single slice of IBSR_17 validation dataset

8.4) Experiment_3:

We have achieved some results with some preprocessing pipeline as mentioned above. Next the analysis of the model performance well with hyper-parameter tuning, resampling and patch extraction. Random and useful patch extraction (**Section 5**) was performed and showed significant improvement on the performance of the model. Tuning various hyper parameters like content threshold, patch stride and patch size contribute a lot in performance of the prediction as shown in the following table [9, 10, 11]. In this stage we tried random (not the same number of patches from each of the classes in this case as we have four class CSF, GM, WM and Background) extraction strategies for extracting the sample patches and it works well (**Section 5.1**)

Model	Scan ID	CSF	GM	WM	Mean(DSC)
3D U-SegNet **Patch size 64**	IBSR_11	0.8907	0.9150	0.9011	**0.9022**
	IBSR_12	0.8752	0.9020	0.8886	**0.8852**
	IBSR_13	0.8102	0.9153	0.8546	**0.8600**
	IBSR_14	0.8434	0.9128	0.8551	**0.8704**
	IBSR_17	0.8870	0.9286	0.8664	**0.8940**

Table 10: Performance the model (in Dice Coefficient) on validation subset

Model	Scan ID	CSF	GM	WM	Average(HS)
3D U-SegNet **Patch size 64**	IBSR_11	11.696	7.405	4.949	**08.016**
	IBSR_12	18.980	07.071	8.246	**09.099**
	IBSR_13	14.313	09.723	11.165	**12.200**
	IBSR_14	18.814	8.366	10.180	**12.453**
	IBSR_17	14.416	12.041	15.165	**13.873**

Table 11. Performance of the model (in Hausdorff distance (HD)) on validation subset: **Distance between Segmented image and ground truth by number of voxels**

Model	Scan ID	CSF	GM	WM	Average(VD)
3D U-SegNet **Patch size 64**	IBSR_11	0.0404	0.0624	0.1409	**0.0812**
	IBSR_12	0.0905	0.0718	0.0718	**0.0780**
	IBSR_13	0.3065	0.0001	0.0078	**0.1048**
	IBSR_14	0.0102	0.0521	0.0871	**0.1494**
	IBSR_17	0.1275	0.0844	0.2204	**0.1441**

Table 12: Performance of the model (in average volumetric difference (AVD)) on validation subset

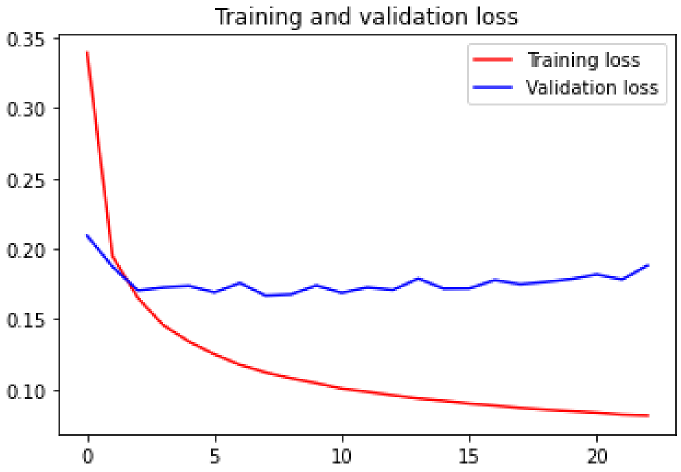


Figure_11: Loss function of train and validation Experiment_3

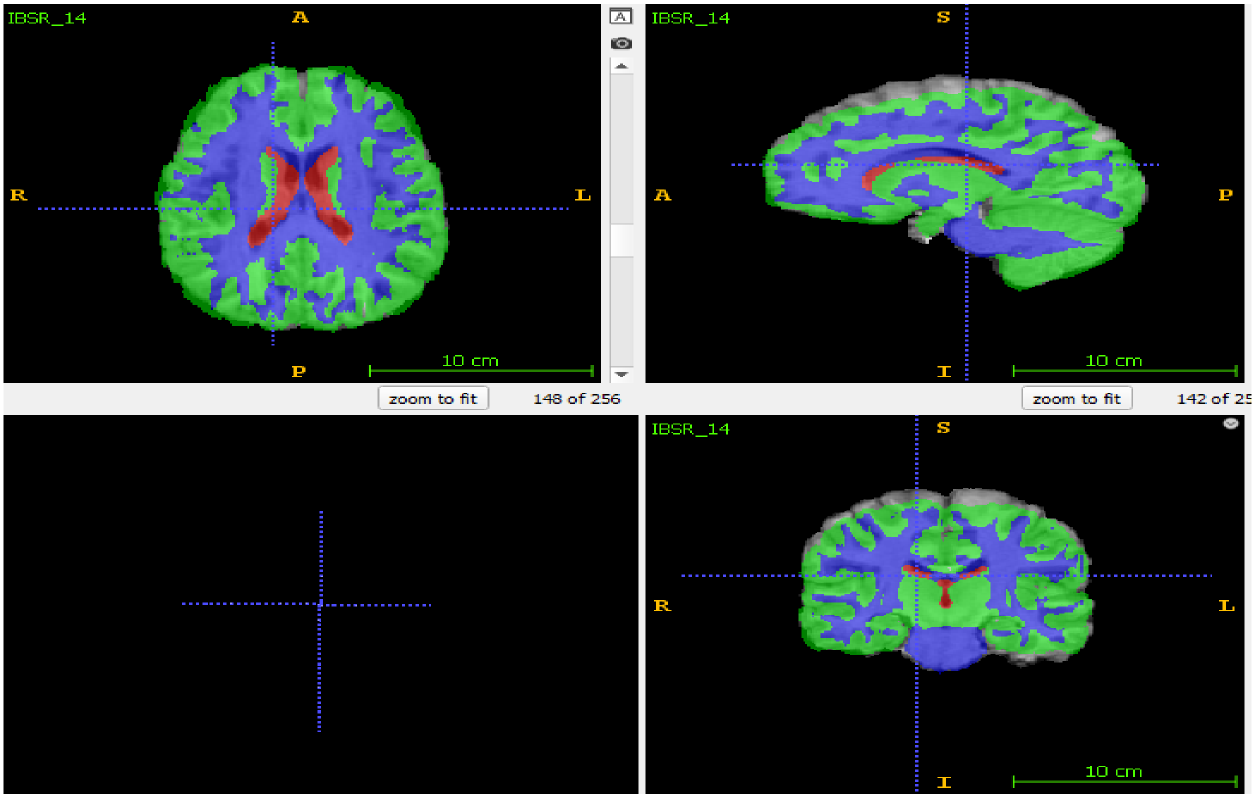


Figure_12: Qualitative result of Experiment_3 for a single slice of IBSR_14 validation dataset

8. Conclusion

In this work, we proposed using Unet and SegNet architectures for 3D brain tissue segmentation and investigated the performance of the models under different hyper-parameters and preprocessing techniques. Before this, preprocessing steps necessary to prepare images for MRI segmentation have been done sequentially (described in Section 4). The most important steps include standardization, bias field correction, Image generation, resampling, and useful patches extraction. Our results show that random patch extraction preprocessing was clearly effective in boosting the performance. Resampling assisted mainly in unifying the voxel spacing to make all the inputs have similar spatial spaces. Generating synthetic images with same size of patch helped a little for the improvement of the performance. Patch sizes had important effect on the performance as well. The best patch size that fitted the model was 64 x 64, while, smaller sizes just did not work as good. Future work includes extending the analysis with modified ResNet-Unet architectures. We also plan to ensemble different models to improve the DSC for CSF especially.

References

[1] X. Sun, L. Shi, Y. Luo, W. Yang, H. Li, P. Liang, K. Li, V. C. T. Mok, W. C. W. Chu, and D. Wang. Histogram-based normalization technique on human brain magnetic resonance images from different

[2] M. Rajchl, N. Pawlowski, D. Rueckert, P. M. Matthews, and B. Glocker. NeuroNet: Fast and Robust Reproduction of Multiple Brain Image Segmentation Pipelines. arXiv e-prints, 2018.

[3] Nabil Ibtehaz1 and M. Sohel Rahman1: MultiResUNet : Rethinking the U-Net Architecture for Multimodal Biomedical Image Segmentation

[4] Konstantinos Kamnitsas, Christian Ledig, Virginia F.J. Newcombe, Joanna P. Simpson, Andrew D. Kane, David K. Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. Medical Image Analysis, 36:61 – 78, 2017.

[5] Dice LR. Measures of the amount of ecologic association between species. *Ecology*. 1945;**26**(3):297–302. doi: 10.2307/1932409.

[6] Reddy AR, Prasad EV, Reddy LSS. Abnormality detection of brain mr image segmentation using iterative conditional mode algorithm.

[7] https://www.pyimagesearch.com/2019/07/08/keras-imagedatagenerator-and-data-augmentation/

[8] Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation