

UNIVERSITY OF KWAZULU-NATAL

College of Agriculture, Engineering and Science



A Patch-based Convolutional Neural Network for Localized MRI Brain Segmentation

by

Trevor Constantine Vambe

219095439

Supervisors:

Prof. Serestina Viriri

Dr. Mandlenkosi Gwetu

A dissertation submitted in fulfillment of the academic requirements for the degree of Master

of Science in Computer Science,

Durban, South Africa, 2020

Declaration of Authorship

I, Trevor C. Vambe, declare that this dissertation titled, ‘A Patch-based Convolutional Neural Network for Localized MRI Brain Segmentation’ and the work presented in it are my own.

I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this dissertation has previously been submitted for a degree or any qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly stated.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this dissertation is entirely my own work.
- I have acknowledged all main sources of help.
- Where the dissertation is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.



Trevor C. Vambe

Declaration-Supervisor

As the candidate's supervisor, I agree to the submission of this dissertation.



Prof. Serestina Viriri

Declaration-Publications

DETAILS OF CONTRIBUTION TO PUBLICATIONS that form part of and/or include research presented in this dissertation.

- Trevor Vambe, Serestina Viriri, and Mandlenkosi Victor Gwetu. A Patch-based Convolutional Neural Network for Localized MRI Brain Segmentation. ACIIDS-2021 Springer Lecture Notes. [Submitted - October 2020]

The authors' contributions in the paper above are as follows:

- 1st Author : Concept formulation, literature review, design and implementation of algorithms and preparing and editing the write-up.
- 2nd and 3rd Authors : Concept improvement, providing advice, highlighting important considerations with respect to methods and techniques, and proof-reading manuscripts.

Abstract

Accurate segmentation of the brain is an important prerequisite for effective diagnosis, treatment planning, and patient monitoring. The use of manual Magnetic Resonance Imaging (MRI) segmentation in treating brain medical conditions is slowly being phased out in favour of fully-automated and semi-automated segmentation algorithms, which are more efficient and objective. Manual segmentation has, however, remained the gold standard for supervised training in image segmentation. The advent of deep learning ushered in a new era in image segmentation, object detection, and image classification. The convolutional neural network has contributed the most to the success of deep learning models. Also, the availability of increased training data when using Patch Based Segmentation (PBS) has facilitated improved neural network performance. On the other hand, even though deep learning models have achieved successful results, they still suffer from over-segmentation and under-segmentation due to several reasons, including visually unclear object boundaries. Even though there have been significant improvements, there is still room for better results as all proposed algorithms still fall short of 100% accuracy rate. In the present study, experiments were carried out to improve the performance of neural network models used in previous studies. The revised algorithm was then used for segmenting the brain into three regions of interest: White Matter (WM), Grey Matter (GM), and Cerebrospinal Fluid (CSF). Particular emphasis was placed on localized component-based segmentation because both disease diagnosis and treatment planning require localized information, and there is a need to improve the local segmentation results, especially for small components. In the evaluation of the segmentation results, several metrics indicated the effectiveness of the localized approach. The localized segmentation resulted in the accuracy, recall, precision, null-error, false-positive rate, true-positive and F1-score increasing by 1.08%, 2.52%, 5.43%, 16.79%, -8.94%, 8.94%, 3.39% respectively. Also, when the algorithm was compared against state_of_the_art algorithms, the proposed algorithm had an average predictive accuracy of 94.56% while the next best algorithm had an accuracy of 90.83%.

Acknowledgements

I would like to express profound gratitude to my supervisors, that is, Prof. Serestina Viriri and Dr Mandlenkosi Gwetu, for their invaluable counsel, guidance and feedback during the work related to this dissertation. I would also like to extend heartfelt gratitude to several academics for their motivation and advice during my research. Last but not least I would like to thank my family for their continued support every step of the way.

Contents

1	General Introduction	1
1.1	Introduction	1
1.2	Motivation	4
1.3	Problem Statement	5
1.4	Research Question	6
1.4.1	Research Aims	6
1.4.2	Research Objectives	6
1.5	Contributions Of The Dissertation	7
1.6	Organisation Of The Dissertation	7
1.7	Conclusion	8
2	Background And Related Work	9
2.1	Introduction	9
2.2	Human Brain Anatomy	9
2.3	Magnetic Resonance Imaging	10
2.3.1	Introduction	10
2.3.2	The MRI Machine	10
2.3.3	Subatomic Particles	11
2.3.3.1	Spectroscopy	13
2.3.4	Magnetic Dipole Moments	13
2.3.5	NMR And MRI Working Principles	15
2.3.6	Image Acquisition In MRI	15

2.3.6.1	The Larmer Frequency	16
2.3.7	Measuring The Magnetic Resonance Signal	17
2.3.8	Longitudinal Relaxation	17
2.3.9	Transverse Relaxation	19
2.4	Datasets	21
2.4.1	Common File Formats	21
2.4.2	Kaggle Datasets	22
2.4.3	Figshare Datasets	22
2.4.4	MICCAI Datasets	22
2.4.5	Open Access Series Of Imaging Studies (OASIS) Datasets	23
2.4.6	ITK-SNAP	24
2.4.7	National Bioscience Data Center (NBDC Human Database)	24
2.4.8	Alzheimer’s Disease Neuroimaging Initiative (ADNI)	25
2.4.9	Multiple Observations Across Networks (SIMON)	25
2.4.10	Reshare	26
2.4.11	LORIS	26
2.4.12	NeuroImaging Tools & Resources Collaboratory (NITRC)	27
2.5	Brain MRI Segmentation	29
2.6	Neural Networks (NN)	32
2.6.1	Introduction	32
2.6.2	CNN Applications To Brain Segmentation	33
2.6.3	Sliding Windows	34
2.6.4	Fully Convolutional Neural Network (Fully-CNN)	35
2.6.4.1	GANs	35
2.6.4.2	Encoder-Decoder Architectures	36
2.6.4.3	U-Net	36

2.6.4.4	Ensemble	36
2.6.5	Section Conclusion	36
2.7	Conclusion	37
3	Methods And Techniques	38
3.1	Introduction	38
3.2	Experimental Objectives	38
3.3	Machine Learning Tools Used In The Research	38
3.3.1	Programming Platform	38
3.3.1.1	Keras	39
3.3.1.2	Tensorflow	39
3.3.1.3	Matplotlib	40
3.3.1.4	SKLearn	40
3.3.1.5	Numpy	40
3.3.1.6	ADDED Packages-Mayavi	40
3.3.1.7	Machine learning Hyper-parameters	41
3.3.2	Convolutional Neural Networks Architecture	42
3.4	Experimental Design	48
3.4.1	Conversion From NifTI To Numpy Arrays	48
3.4.2	Train-Validation Splitting	49
3.4.3	Segmentation Model Voxel Voting	49
3.4.4	Data Preprocessing	50
3.4.4.1	The Chosen Dataset-NITRC-CANDI	50
3.4.4.2	Image Cropping	50
3.4.4.3	Segmentation Classes	51
3.4.4.4	Patch Generation	52
3.4.4.5	Experiments For The Present Study	53

3.5	The Proposed Algorithm	55
3.6	Evaluation Metrics	57
3.6.1	Accuracy And Loss Metrics	58
3.6.2	Dice Similarity Coefficient (DSC)	59
3.6.3	Confusion Matrix	59
3.7	Conclusion	60
4	Results and Discussions	61
4.1	Introduction	61
4.2	Proposed Algorithm's Vs Cui's algorithm	61
4.3	Brain Segmentation	62
4.3.1	Training And Validation Losses	62
4.3.2	Training And Validation Accuracy	63
4.3.3	Confusion Matrix	65
4.3.3.1	Test Statistics Derived From The Confusion Matrix	66
4.3.4	Section Conclusion	69
4.4	Localized Component PBS	70
4.4.1	Training Losses	70
4.4.2	Validation Losses	71
4.4.3	Training Accuracy	72
4.4.4	Validation Accuracy	73
4.4.5	Segmented Images Of The Hippocampus And Thalamus-Proper	74
4.4.6	Confusion Matrix	74
4.4.6.1	Test Statistics Derived From The Confusion Matrix	77
4.4.7	Section Conclusion	83
4.5	Comparson With Prior Art	83
4.6	Conclusion	87

5	Conclusion And Recommendations	88
5.1	Summary	88
5.1.1	Contributions	89
5.2	Limitations And Future Work	90
5.2.1	Limitations	91
5.2.2	Future Work	91
	Appendices	92
A	HD, MSD And DSC For OASIS, Collin27 And CANDI Datasets	93
B	ANOVA Analysis For OASIS, Collin27 And CANDI Datasets	94
C	Commonly Used Evaluation Metrics Calculated From The Confusion Matrix	95
D	Brain Tissue Classes As Provided In The CANDI Website [84]	98

Abbreviations

2.5D Two and a Half Dimensional. 5

2D Two Dimensional. 5, 38, 49, 50, 60

3D Three Dimensional. 2, 5, 10, 25, 30, 38, 40, 49, 50, 54

AD Alzheimer's Disease. 30

ADHD Attention-Deficit Hyperactivity Disorder. 28

ADNI Alzheimer's Disease Neuroimaging Initiative. vii, 25

AFNI Analysis of Functional NeuroImages. 27

AKA Also Known As. 18

ANN Artificial Neural Networks. 32

ANOVA Analysis Of Variance. x, xv, xvi, 94

ANTs Advanced Normalization Tools. 27

BIDS Brain Imaging Data Structure. 28

CAD Computer Aided Diagnosis. 2

CANDI Child and Adolescent NeuroDevelopment Initiative. viii, x, xvi, 48, 50, 51, 53, 93,
98

CE Computational Environment. 27

CMA Certified Medical Assistant. 28, 51

CNN Convolutional Neural Network. vii, xv, xvii, 2–5, 7–9, 32–37, 58, 85, 86, 91

CRBS Center for Research in Biological Systems. 27

CSF Cerebrospinal Fluid. xv, 4, 7, 26, 51, 52, 54, 55, 60–62, 65–70, 87, 89, 90, 98

CT Computed Tomography. 1, 31

Df Degrees of Freedom. 94

DICOM Digital Imaging and Communications in Medicine. 24, 26–28

DNA Deoxyrebonucleic Acid. 10

DSC Dice Similarity Coefficient. ix, x, xv, xvi, 59, 60, 93, 94

DTI Diffusion Tensor Image. 26

EC Error Correction. 84, 85, 91

FID Free Induction Decay. 15

fMRI Functional MRI. 25, 26

FN False Negative. xv, 60, 65, 66, 76, 95

FP False Positive. xv, 60, 65, 66, 76, 95, 96

FSI FMRIB’s Software Library. 27

FTP File Transfer Protocol. 27

GAN Generative Adversarial Network. 2, 3, 33

GM Grey Matter. xv, 4, 7, 51, 52, 54, 55, 60–62, 65–69, 87, 89, 90

GPU Graphics Processing Unit. 39, 91

HD Hausdorff Distance. x, xv, 26, 93, 94

HTTP Hyper Text Transfer Protocol. 25

IBSR Internet Brain Segmentation Repository. 28

ID Identity. 22, 24

IDA Image & Data Archive. 25

Inf-Lat-Vent Inferior Lateral Ventricle. 52

IR Image Repository. 27, 28

LORIS Longitudinal Online Research and Imaging System. vii, 26, 27

lr Learning Rate. 41, 42

LWV Local Weight Voting. 86

MICCAI Medical Image Computing and Computer Assisted Intervention. 22

MINC Medical Imaging NetCDF. 25, 27

MR Magnetic Resonance. 10, 24

MRI Magnetic Resonance Imaging. i, vi, 1, 2, 4, 5, 7, 8, 10, 13–15, 22, 24–26, 28, 31, 32, 37, 50

MSD Mean Surface Distance. x, xv, xvi, 93, 94

MTI Magnetization Transfer Imaging. 24

MV Majority Voting. 86

NBDC National Bioscience Database Center. vii, 24, 25

NITRC NeuroImaging Tools & Resources Collaboratory. vii, viii, 27, 28, 50

NMR Nuclear Magnetic Resonance. vi, 1, 10, 13–15, 24

NMV Net Magnetic Vector. 16, 19

NN Neural Network. vii, xv, 2–4, 32, 34, 36, 38–40, 42, 44, 47, 53–55, 58, 60, 62, 63, 65–67, 69–74, 77–79, 83, 87, 89–91

OASIS Open Access Series of Imaging Studies. vii, 23, 24, 93

PBM Patch Based Method. 86

PBS Patch-based Segmentation. ix, 5, 70

PD Proton Density. 25

PET Position Emmision Tomography. 24

R Resource Registry. 27

RAM Random Access Memory. 28

ReLU Rectified Linear Units. 33, 43

RF Radio-Frequency. 1, 15, 17, 19

ROI Regions Of Interest. 1

SGD Stochastic Gradient Descent. 41, 42, 57

SPBM Sparse Patch Based Method. 86

Sq Square. 94

TIM Total Imaging Matrix. 24

TN True Negative. xv, 60, 65, 66, 76, 95, 96

TOF Time of Flight. 30

TP True Positive. xv, 60, 65, 66, 76, 95, 96

TPU Tensor Processing Unit. 39

tsv Tab Separated Variables. 25

TVTK Traits Visualisation Tool Kit. 41

UCSD University of California San Diego. 27

URL Uniform Resource Locator. 25

VTK Visualisation Tool Kit. 41

WM White Matter. xv, 4, 7, 51, 52, 54, 55, 60–62, 65–69, 87, 89, 90, 99

WMH White Matter Hyperintensities. 31

List of Tables

3.1	Table showing a summary of the layers of the proposed NN model	44
3.2	Categorization of brain tissues into WM, CSF, GM and background	52
3.3	Confusion matrix for binary classification	60
4.1	Confusion matrix for the first experiment	65
4.2	Table showing TP, TN, FP and FN for the first experiment	66
4.3	Test statistic values obtained from the confusion matrix (Table 4.2)	66
4.4	Confusion matrix for the second experiment	75
4.5	Confusion matrix for the third experiment	75
4.6	Table showing TP, TN, FP and FN for the second experiment	76
4.7	Table showing TP, TN, FP and FN for the third experiment	76
4.8	Test statistic values based on Table 4.6	77
4.9	Test statistic values based on Table 4.7	78
4.10	Accuracy rates for the proposed CNN and state_of_the_art CNNs [32]	84
4.11	Dice ratios for the proposed CNN and state_of_the_art CNNs [126]	84
4.12	Average dice scores for the proposed CNN and state_of_the_art CNNs in segmenting the hippocampus and thalamus-proper [46]	85
4.13	Average dice scores for the proposed CNN and state_of_the_art CNNs in segmenting the hippocampus and thalamus-proper [127]	86
4.14	Dice ratios for the proposed CNN and state_of_the_art CNNs [32]	86
A.1	HD, MSD and DSC for OASIS, Collin27 and CANDI datasets	93
B.1	ANOVA table for OASIS, Collin27 and CANDI datasets for HD	94

B.2	ANOVA table for OASIS, Collin27 and CANDI datasets for MSD	94
B.3	ANOVA table for OASIS, Collin27 and CANDI datasets for DSC	94
C.1	Commonly used evaluation metrics calculated from the confusion matrix . . .	95
D.1	Brain tissue classes as provided in the CANDI website	98

List of Figures

1.1	The image shows the region of interest in diagnosing Parkinson’s disease [93] .	3
2.1	An image showing a labelled diagram of the MRI scanner[110]	11
2.2	The structure of an atom (Hydrogen) [49]	12
2.3	Images showing the earth and an atom rotating about its axis	12
2.4	Recovery of the z-magnetization after 90-degree pulse [49]	19
2.5	Recovery of the z-magnetization after 90-degree pulse [49]	20
2.6	A diagram showing the structure of a typical CNN architecture [4]	32
3.1	The neural network architecture for the proposed algorithm	43
3.2	Images showing a raw patch [a] and its ground truth patch [b]	53
4.1	First experiment’s training Loss graph	62
4.2	First experiment’s validation Loss graph	63
4.3	First experiment’s training accuracy graph	64
4.4	First experiment’s validation accuracy graph	64
4.5	Training losses for the second and third experiment	71
4.6	Validation losses for the second and third experiment	71
4.7	Training accuracy for the segmentation of two (a) and three components (b) .	72
4.8	Validation accuracy for the second and third experiment	73
4.9	Images showing localized segmentation of the hippocampus and thalamus-proper and their corresponding ground truth images	74

1. General Introduction

1.1 Introduction

The human brain is a crucial organ in the body, owing to its numerous functions. Its main functions include the control and coordination of other organs of the body. Neurological abnormalities that affect the brain have tremendous effects on the body as a whole. Several medical technologies have been invented to detect these abnormalities. Some of the technologies include but are not limited to X-ray, Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) [56]. Of these technologies, MRI is preferred for soft tissue imaging because it has no known side effects, it produces exceptionally good three-dimensional (3D) images of human internal organs, it produces no ionizing radiation and it is non-invasive [125]. MRI scanned data has to be segmented into different regions of interest (ROIs) before other important clinical steps can be taken. The segmented regions can be used for volumetric analysis during disease diagnosis or treatment planning and monitoring [20, 78]. The regions may also be used in tumour detection and tumour progression monitoring. The component volumetric analysis can be used to diagnose diseases like epilepsy, Alzheimer's, schizophrenia, bipolar disorder, multiple sclerosis, and attention-deficit disorder[73, 114].

MRI is a technique that has its roots in the NMR (Nuclear Magnetic Resonance) technique. The NMR technique was proposed by Bloch and Purcell during World War II and perfected after the same war [38, 44]. The MRI technology works by magnetizing the hydrogen protons in the human body using a powerful magnet, typically 4 Tesla. This magnetization results in the protons becoming magnetic dipoles that precess about an axis parallel to the strong magnetic field. In addition to the strong magnet, there are gradient coils (0.5 Gauss) that generate voxels and slices [38, 49]. These gradient coils are applied for short periods. Halfway through the magnetic gradients, an RF-signal is sent into the body. The RF-signal will excite protons in a cross-section of the body in the form of a slice. After the pulse, only the excited protons will be relaxed, and the relaxing protons produce sinusoidal signals. The signals are

emitted by each voxel, captured and stored as the information for each slice. The slices are integrated by a computer to produce 3D image data [38, 44]. Computer-Aided Diagnosis (CAD) systems have been extensively used in the medical field to help with visualization, quantitative analysis, feature detection and extraction and classification of regions of interest.

CAD has become pervasive in the medical field. The accuracy of these CAD systems depends on the efficiency and effectiveness of the model used. In general, there are two ways in which CADs can be implemented, firstly, using expert systems within a given domain, and secondly, using machine learning. Each approach has its advantages and disadvantages. Expert system CADs heavily depend on the developer and hence are prone to human error. Machine learning models, on the other hand, allow models to learn from given datasets; hence the models might ultimately learn how to classify the datasets but under-perform when subjected to new datasets [61]. While expert systems and machine learning CADs are both time-intensive, machine learning CADs are more flexible in terms of specifying classification logic. This research focuses on machine learning because it is not developer dependent and is generally less resource-intensive than the expert systems approach.

Machine learning has done exceptionally well in brain MRI medical image segmentation. Its algorithms require large volumes of training data [103, 128]. This data is expensive and difficult to get and requires input from several stakeholders [40]. The information is also strictly regulated by patient-doctor confidentiality laws. In addition, the data used for supervised training requires correctly annotated ground truth data to facilitate supervised learning. While several MRI segmentation algorithms have been proposed in the past few decades, the majority of these algorithms under-perform when presented with complex medical image data [32]. Both supervised and unsupervised neural network (NN) machine learning models can be used for MRI medical image segmentation [73].

Convolutional Neural Networks (CNNs) have been able to segment medical images with very high accuracy. Some of the achieved accuracy rates were as high 90% or more, an example of this being an algorithm developed by Cui *et al.* [32]. Several CNN algorithms have been proposed for brain image segmentation based on the few layers that make up a CNN. The study carried out by Toan *et al.* [20] summarized some of the state_of_the_art deep CNN. Most NNs are based on well-known architectures such as GANs, encoder-decoders, deep medic,

DenseNet, U-Net, V-Net, and VoxResNet, just to mention a few. The present study makes use of a patch-based NN.

One of the several ways of categorizing CNNs is as either fully connected CNN or window sliding [112]. The majority of researchers focused more on fully-CNN and little research has been done in window sliding. The research studies that were based on fully connected CNN include , GANs [57, 60, 99] and several encoder-decoder based networks [5]. The encoder-decoder networks included amongst others the V-Net [94, 129], Y-Net [24], U-Net [21, 30, 37, 45, 48, 63, 67, 74, 101, 111, 122] and ensemble [2, 11].

In diagnosing diseases there is no need to segment the whole brain since most neurological disorders affect specific regions of the brain. Therefore, medical practitioners normally zero in on specific components depending on the patient [41]. Figure 1.1 shows the region of the brain that is crucial in the diagnosis of Parkinson’s disease [93].

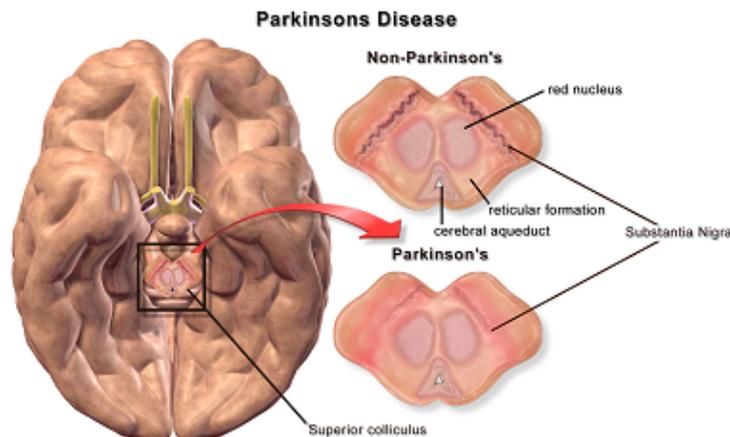


Figure 1.1: The image shows the region of interest in diagnosing Parkinson’s disease [93]

In Figure 1.1 the wearing out of the substantia nigra is also evident. Furthermore, the study carried out by Garz *et al.* [41] expanded on the point and describes how the substantia nigra’s wearing out is responsible for the development of Parkinson’s disease. With this background, the segmentation of the rest of the brain would, therefore, be a waste of computer resources. To be efficient it would be better to consider the volume that surrounds the region of interest.

The present study was an extension of the study carried out by Cui *et al.* [32]. In the present study, the first major experiment focussed on developing an algorithm for accurate human

brain segmentation into WM, GM and CSF. The second section of the present study focussed on the development of a localized component-based and patch-based segmentation (PBS) algorithm. In this section, two major experiments were carried out. The first experiment focussed on the segmentation of two components, that is, thalamus-proper and hippocampus. In the second experiment, three components were segmented, that is, cerebellum-cortex, thalamus-proper and hippocampus. Several test statistics were then used in the evaluation of the proposed algorithms.

1.2 Motivation

MRI images used in segmentation are affected by inhomogeneity of the magnetic field and partial volume effects [2, 8]. These factors affect the quality of the images, and hence human brain image segmentation algorithms are presented with data that would have been corrected for these effects. Although these images may be degraded and not in any way close to perfect, they are better than the original images. Brain segmentation also faces challenges due to the complexity of brain structures and the fact that the components show varied differences in structure and size as one moves from one patient to another [9, 17, 24]. These differences present learning difficulties for NNs. Traditionally human brain segmentation was done manually, but this had several challenges hence the introduction of semi-automated and fully automated algorithms [5, 9, 10, 13, 18]. Deep learning algorithms have proven to be very effective in human brain segmentation [2, 3, 5]. In particular, CNNs have proved to be very effective in image segmentation. The major challenges faced by CNN are the absence of sufficient data for training, model inability to adapt to different datasets, and lastly, tissue distribution overlap [13, 14, 17]. Also, some brain components are small, hence there is limited data available for training the given algorithm to segment the components [9, 23]. Also, some component boundaries are not very clear, and in some cases, the boundaries are not available, which makes segmenting them very difficult [8, 9, 11]. Although traditional algorithms and existing software (such as Functional MRI of the brain FSL and Freesurfer [5, 9]) offer ways of segmenting the human brain, they all generally suffer from either over-segmentation or under-segmentation. The former occurs when the resulting regions of interest are more than expected, while, the latter occurs when the segmentation algorithm produces insufficient regions of interest. Over-segmentation and under-segmentation result from the previously

mentioned CNN and general brain segmentation challenges. The present study aimed at testing if localized brain image segmentation could improve segmentation results.

1.3 Problem Statement

MRI is a preferred modality because it produces images with high contrast and higher resolution when compared to other modalities [120]. MRI segmentation is not a simple exercise because primarily the source images are more often than not degraded by noise, bias field, and partial volume effects. Traditionally, manual segmentation was the only method used for MRI brain segmentation. In this approach, segmentation was carried out slice by slice. On average, there would be more than 80 slices, with each being 512*512 pixels. Each brain component's contour lines would be drawn in each slice by a seasoned radiographer, and this process was time-intensive [34, 47]. However, radiological results are needed timeously, and the radiological laboratories frequently have many patients. The stated factors make manual segmentation inefficient as it is time-consuming, prone to human error, not reproducible, and highly subjective [32, 34, 40, 47]. A plethora of algorithms has been proposed to automate the process of MRI brain segmentation. Of the proposed algorithms, CNNs have been known to segment the human brain with exceptionally high accuracy rates [47]. These algorithms require a lot of training data, and this data is difficult to get [14]. Researchers noted that most of the images used as input to these algorithms had highly redundant data, hence some researchers proposed the use of patch-based segmentation (PBS). PBS generates small images (patches) from the provided images. these patches are used in training and testing the neural network. PBS can be viewed as a data augmentation strategy. Additionally, research on the implementation of different sized patches produced promising results. Some researchers also experimented with 3D and 2D patches. These researchers realized that 3D patches were time and computational resource-intensive but offered rich neighbourhood information as compared to 2D patches [113]. Some researchers have also introduced 2.5D networks which integrate sagittal, axial and coronal information in 2D image segmentation [14]. This approach is also time and computational resource intensive. On the other hand, slight changes in the volume of the human brain's components could indicate a neurological disorder. Since existing methods are yet to achieve segmentation accuracy rates of 100% (or close to 100%), there is room for further research in this domain. Any resulting improvement in segmentation accuracy would

also improve the performance of any subsequent clinical processes. Specific problems that arise in the development and use of segmentation algorithms include the low contrast between tissues to be segmented, brain component size variations, tissue class imbalance, and the brain's complexity in general. These problems result in the misclassification of pixels, resulting in low accuracy [20]. There is, therefore, a need to develop brain segmentation algorithms that are fully automated, do not over-fit given data and are effective even on small brain components [47].

1.4 Research Question

This study seeks to answer the following research question:

Does localized brain MRI segmentation improve the segmentation accuracy of the hippocampus, thalamus-proper and cerebellum-cortex?

This localized segmentation could also aid in research related to the segmentation of small brain organs, which is currently a daunting task for neural networks. Additionally, the improved accuracy could positively contribute to the accurate diagnosis, treatment and treatment monitoring of diseases such as co-resistant temporal lobe epilepsy, Alzheimer's Disease, dementia and Parkinson's Disease.

1.4.1 Research Aims

The major aims of the present study were to:

1. Propose an algorithm for brain MRI segmentation that would rival state-of-the-art algorithms in brain segmentation,
2. Implement the algorithm proposed in item (1) in localized segmentation of the hippocampus, thalamus-proper and cerebellum-cortex and evaluate its performance.

1.4.2 Research Objectives

1. Effectively preprocess MRI brain images for component-based segmentation.

2. Train a CNN to segment a ROI using MRI patches.
3. Evaluate the achieved segmentation performance.

1.5 Contributions Of The Dissertation

The main contributions of this dissertation are as follows:

- Development of an algorithm for the accurate segmentation of the human brain into three main components, that is, WM, GM and CSF. The proposed algorithm resulted in improved segmentation results when compared with the algorithm it was based on,
- The proposed algorithm was used with localized patches. The implementation was carried out in two experiments whose results were averaged for analysis purposes. The first of the two experiments segmented two components, that is, thalamus-proper and hippocampus. The second experiment segmented three components, that is, cerebellum-cortex, thalamus-proper and hippocampus. Several test statistics were then used in evaluating the performance of the proposed algorithm. The results of the proposed algorithm rivalled those of state_of_the_art algorithms in brain image segmentation.

1.6 Organisation Of The Dissertation

The remainder of the dissertation is structured as follows:

- Chapter 2 - Background And Related Work: This chapter investigates the current methods in brain MRI segmentation. It also gives a summary of brain MRI datasets and the background on the human brain anatomy and CNN segmentation algorithms.
- Chapter 3 - Methodology And Techniques: This chapter gives the experimental objectives, machine learning tools used in the study, the CNN architecture, model design, dataset preprocessing, dataset processing and evaluation metrics used in the study,
- Chapter 4 - Results and Discussions: This chapter summarises the experimental results and their discussions.

- Chapter 5 - Conclusions and Recommendations: This chapter summarises the conclusions drawn from the present study and recommends possible extensions for future studies.

1.7 Conclusion

This chapter gave a general summary of the dissertation. It additionally gave the motivation for the dissertation, its problem statement, research question, research aims and research objectives. The chapter also summarized the contributions made by this dissertation and concluded with the organization of the chapters in the rest of the dissertation. The subsequent chapter lays a foundation for the remainder of the dissertation by presenting a general background to the human brain anatomy. It also gives a summary of the basic concepts in MRI. The following chapter also gives a summary of some of the available datasets and details of the chosen dataset. It also gives a summary of the basic concepts in CNN and the related algorithms for brain MRI segmentation.

2. Background And Related Work

2.1 Introduction

This chapter covers the background and studies conducted related to the dissertation. It starts with a summary of the anatomy of the human brain. The chapter then goes on to summarize some of the basic concepts in MRI. Additionally, it gives a summary of the datasets considered for the present study. The chapter also provides a detailed description of the dataset used in the present study. It also summarises CNN basic concepts and studies related to the present study. The studies included in the summary are of those algorithms that used CNN in human brain segmentation.

2.2 Human Brain Anatomy

The brain is a key component in the body. It is one of the largest and complex organs in the body. Scientists have studied it for generations but little progress has been made in understanding the brain and how it functions. The organ is estimated to comprise of more than one hundred billion nerve cells that communicate using electrical pulses. The human brain makes it possible for diverse intricate physiological and psychological functions to take place. Some of its functions include emotions, movement and consciousness. The brain is also responsible for involuntary functions of the body such as breathing, digestion and pulse control[28, 106].

The brain is known as the organ that controls the nervous system [106]. Additionally, the nervous system is known to be responsible for controlling the body's interactions with the outside world. The brain achieves this function by controlling the individual's behaviour and coordinating the sensory functions of the body. The sensory organs that are usually used are the tongue, skin, eye, only to mention a few. The nervous system is also responsible for controlling the internal state and internal processes of the body [106]. According to Swanson and

Larry [106] there are a lot of schools of thought on how the human brain functions and generates such high-level functions as thinking and conscious thought. Some philosophers look at it from a philosophical and mathematical modelling standpoint, others took the psychological standpoint while others took the physical science of biology, chemistry and physics perspective. Additionally, Swanson and Larry stated that interdisciplinary fields like Neuroscience were introduced in the 1970s. Also, they alluded to the fact that the most central concept in all the proposed arguments was the architecture of the human brain [106].

2.3 Magnetic Resonance Imaging

2.3.1 Introduction

Magnetic resonance imaging (MRI) has its roots in Nuclear Magnetic resonance (NMR) which was pioneered by Bloch and Purcell[44]. It is a preferred modality for image acquisition for soft tissues because it is non-invasive, it has no known side effects, it can adequately produce high-resolution 3D images of medical organs and last but not least it has no ionizing radiation [125]. According to Cerutti [22], ionizing radiation is known to be dangerous, carcinogenic and causes a genetic mutation in humans. Ionizing radiation can efficiently kill those cells that are developing quickly, induces several malignant neoplasms in human and result in the de-structuring of the human deoxyribonucleic acid (DNA) [22], hence MR imaging has become pervasive in the medical imaging.

MRI images are acquired using an MRI machine which is normally found in a special radiological laboratory. In the laboratory, no metallic objects are permitted and the staff is well educated on the safety precautions to be taken when inside the laboratory.

2.3.2 The MRI Machine

The MRI machine is made up of the main magnet, gradient magnets and a transceiver. The main magnet provides a very strong magnetic field, in the order of up to 4 Tesla. The gradient magnets provide magnetic fields that are way lower than those of the main magnet, in the order of 5 Gauss. Figure 2.5 shows a typical MRI machine:

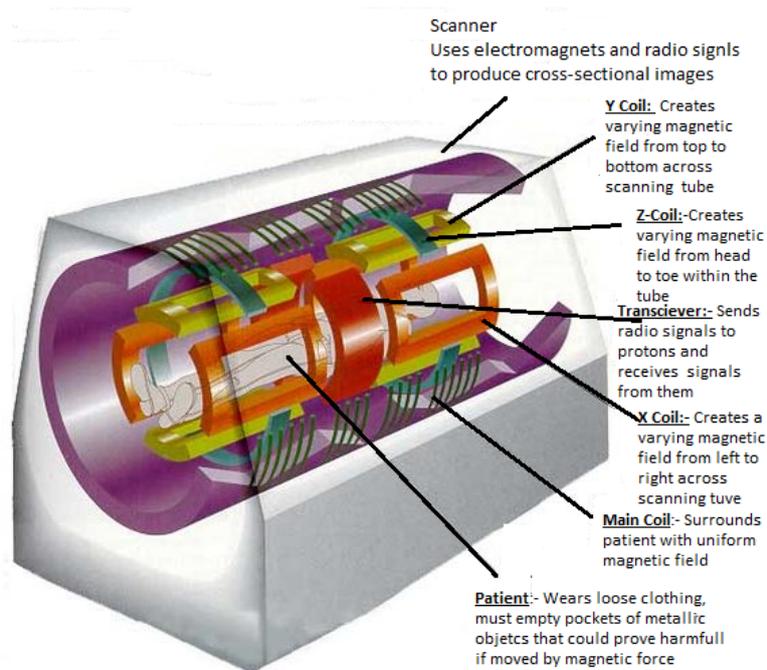


Figure 2.1: An image showing a labelled diagram of the MRI scanner[110]

The diagram in the image in Figure 2.5 shows where the patient would be positioned, the main magnetic coil, the three gradient coils (X-coil, Y-coil and Z-coil) and the transceiver. Additionally, the gradient coils, main coil (main magnet) and transceiver are described in detail in the following subsections.

2.3.3 Subatomic Particles

An atom is defined as the smallest part of an element and it comprises of neutrons, proton and electrons. All three particles have the same quantized spin. The quantized spin is given by $\frac{h}{4\pi}$, where h is the Planck's constant [49]. Planck's constant is defined as the ratio of the energy of a photon to its frequency. Additionally, electrons whiz around the nucleus of an atom in atomic shells while protons and neutrons reside at the centre of the atom. Electrons have a charge of -1.6×10^{-19} and the smallest measure of quantized spin. On the other hand, protons have the same charge as that of an electron but its charge is positive. In addition to that, the proton is 1836 times heavier than the electron. Protons exert strong nuclear force on other protons and neutrons. This is the force that binds neutrons and protons together in the nucleus. The neutrons are neutral and have almost the same mass as the protons [49]. This makes the nucleus positively charged. Figure 2.6 shows a typical atom, the hydrogen atom.

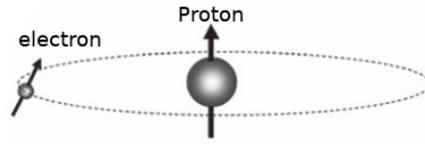


Figure 2.2: The structure of an atom (Hydrogen) [49]

By its design, the hydrogen atom has no neutrons hence the nucleus only has one proton. The atom also has only one electron whizzing around the nucleus. On the other hand, other atoms have nuclei that are a combination of neutrons and protons. Additionally, in all atoms, the number of electrons is always equal to the number of protons in the nucleus. Also, the mass of an atom is mainly derived from the protons and neutrons.

Some atoms have nuclei that have a property called spin, the spin property is analogous to the earth spinning on its axis, as shown in Figure 2.7 a. Hydrogen is one of those atoms that possess a nuclear spin. According to Grover *et al.* [44], the nucleus itself does not spin but its nature forms a magnetic moment. The magnetic moment generates a local magnetic field with a designated north and south pole. The two poles that are present in the atom are the same as those found on a bar magnet (as shown in Figure 2.7 (b)). If the atom is placed in a strong external field β_0 , the nucleus will align itself parallel or perpendicular to the field [44].



(a) The Earth rotating about its axis [109]

(b) An image of the analogy between a precessing atom and a bar of magnet [49]

Figure 2.3: Images showing the earth and an atom rotating about its axis

If a fluid is placed in the magnetic field β_0 , the fluid will spin either parallel or anti-parallel to β_0 . Spinning parallel to β_0 is referred to as the low energy state and spinning anti-parallel to it is referred to as the high energy state [44]. According to Grover, there tends to be an excess of atoms spinning parallel to β_0 . Additionally, bar magnets when placed in β_0 they align themselves parallel to the magnetic field β_0 . On the other hand, nuclei precess about

the magnetic field since they have angular momentum about β_0 . As the nuclei precess about β_0 , their velocity of precession is given by the Larmor equation (equation 2.3.5) [44]. Additionally, several atoms can be used for NMR, some of these include carbon-13, phosphorus and fluorine. For an atom to be used for NMR or MRI its nuclei spin should not be equal to zero. In MRI, hydrogen atoms are used exclusively, but in special circumstances, Xenon-189, Phosphorous-31, Helium-3 or Fluorine-19 could be used [38].

2.3.3.1 Spectroscopy

There are several types of spectroscopic techniques but the commonly known ones are absorption and emission spectroscopy. In absorption, molecules gain energy and the energy absorbed by molecules raises them to a higher state (excited). Just after the excitation molecules commence getting back to their stable state (de-excited). They return to their steady-state through radiation and the radiated energy's wavelength patterns can be used for any analytical purposes [38]. On the other hand in emission electricity is used to energize the molecules. The molecules are continuously excited and de-excited. When molecules go through the excitation and de-excitation they emit radiation. The radiation levels are capture and analyzed. It was noted in experiments that molecules that absorb radiation of a given wavelength would give release to radiation of the same wavelength [38].

A combination of both absorption and emission is referred to as fluorescence. In this type, molecules are continuously irradiated with radiation of the same wavelength which is absorbed by the molecule under investigation. The electromagnetic radiation makes some of the molecules excited and de-excited, hence releasing electromagnetic radiation which can be measured and analyzed [38].

2.3.4 Magnetic Dipole Moments

If a charged ball is made to spin and placed in an external magnetic field, it will experience a force that has a twisting effect on it. The twisting effect forces the axle of rotation to be aligned to the magnetic field of the external magnetic field. Protons and electrons are quantized particles and may only have discrete one charge, one discrete spin and one discrete mass [49]. These properties result in protons and electrons having corresponding discrete magnetic

dipole moments. Quantized particles also possess another property that is very important in NMR and MRI. Quantized particles when placed in an external magnetic field align themselves either parallel or anti-parallel to the magnetic field β_0 . More energy is required for nuclei to be in the anti-parallel state as opposed to the parallel state. For this reason, aligning anti-parallel is referred to as the high energy state while the alternative is referred to as the low energy state [49].

One of the critical properties of nuclei that is taken advantage of in NMR and MRI is magnetic moments. The following equation (Dirac's model) can be used to calculate the magnetic moment of an electron or proton. The equation for calculating the magnetic moment of an electron is as shown in equation 2.3.1:

$$\mu_e = \frac{q_e h}{4\pi m_e} \quad (2.3.1)$$

where the magnetic moment of either the electron or proton is proportional to its charge (q_0) and its magnetic spin ($h/4\pi$) and will be inversely proportional to its mass (m_0) [49]. The magnetic moment of an electron is calculated to be 658 times larger than that of the proton. Even though mathematically electrons are the preferred option, protons were chosen as the particle of choice. The proton was selected because the electromagnetic waves that are required to excite hydrogen nuclei happen to be in the same range that is required to penetrate the human tissue and bone. The electromagnetic waves required to excite electrons happen to be in the microwaves, microwaves are easily absorbed by cells after a centimetre of entry into the human body [49].

Magnetic dipole moments are also important for other uses in MRI. Some atoms like gadolinium have several unpaired outer shell electrons which makes them good magnetic resonance contrast agents. When these are placed in an external magnet, the unpaired electrons align. This creates strong non-uniform magnetic fields in the neighbourhood of the atoms. Materials like these are referred to as paramagnetic because they have magnetic properties only when they are exposed to a strong external magnetic field. These materials are used for contrast enhancement in MRI imaging [49].

2.3.5 NMR And MRI Working Principles

The MRI technique has its roots in Nuclear Magnetic Resonance (NMR). The NMR technique was first proposed by Bloch and Purcell just before World War II and refined just after the same war [38, 44]. NMR makes use of a very strong magnetic field, for humans in the order of 0.1 to 4 Tesla (Earth's magnetic field is about 0.5T), where one Tesla (T) = 10 000 Gauss. The combined use of NMR, pulsed RF signals, superconducting magnets and computerized Fourier transformation was a groundbreaking discovery that ushered in several new application technologies. The very first MRI images were produced in the year 1980 in Nottingham and Aberdeen in the United Kingdom (UK) [38, 44].

In the field of NMR and MRI, we make exclusive use of those atoms that possess nuclear spins that are not equivalent to zero. There are a number of these atoms, for instance, carbon-13, phosphorus and fluorine. The body of a human being is about 66.7% water (H_2O) and each molecule of water is made up of two atoms of hydrogen and one atom of oxygen. There are a number of other substances found in the human body whose chemical formula has hydrogen, some of these substances include fat ($C_{56}H_{108}O_{11}$) and carbohydrates ($C_{12}H_{22}O_{11}$). Hydrogen, therefore, is found abundantly in the human body, hence it was chosen as the atom of choice in NMR and MRI. As stated earlier, MRI currently makes exclusive use of hydrogen atoms except for special circumstances [38].

2.3.6 Image Acquisition In MRI

When imaging a patient, the patient is placed along with the main magnet. A vertical gradient of about 0.5 Gauss is applied for a short period. Halfway through the magnetic gradient, an RF-signal is sent into the body. The RF-signal will excite protons in a cross-section of the body in the form of a slice. The slice would be about one millimetre or less in thickness. After the pulse, only the excited protons will relax and produce a sinusoidal signal with decreasing amplitude (FID) [38, 49].

2.3.6.1 The Larmer Frequency

The most common atom found in the human body is the hydrogen atom. The hydrogen atoms' magnetic fields have a cancelling effect, hence there is no resultant magnetic field. When the human body is placed in a magnetic field, the hydrogen atoms align themselves parallel and anti-parallel to the external magnetic field β_0 [38, 49]. Empirical evidence showed that those particles that align themselves parallel to the field will be slightly more than those that align anti-parallel to the field, hence there is a resulting magnetic field. This net magnetic field is referred to as the net magnetic vector (NMV). The population ratio of the two energy levels can be calculated using the Boltzmann's distribution law, where the Boltzmann's constant is $k = 1.381 * 10^{-23} J/K$ [49]. The Boltzmann's distribution law is as shown in equation 2.3.2:

$$N_{\text{low}}/N_{\text{high}} = \exp(-\Delta E/kT) \quad (2.3.2)$$

where N_{low} is the number of atoms in the low energy state and the opposite is true for N_{high} . Research has shown that in the XY-plane these magnetic moments have a cancelling effect leaving the NMV component along the Z-plane. Empirical results also indicated that NMV was proportional to the density of protons, the external magnetic field β_0 and the square of the gyromagnetic ratio γ [38].

As stated earlier on we have nuclei in a high energy state and those that will be in a low energy state. The difference between the two energy levels can be expressed as shown in equation 2.3.3:

$$\Delta E = h\gamma\beta_0 \quad (2.3.3)$$

where h is Planck's constant ($h = 6.626 * 10^{-34} Js$), γ the gyromagnetic ratio for protons and β_0 the external magnetic field. ΔE is defined as the energy that is required to change an excited proton from a lower energy level to its higher energy level [38, 44, 49]. Changing the state from low to high can also be achieved by applying electromagnetic waves of the same frequency (f). The frequency must be generated from photons that have photon energy given by the equation 2.3.4.

$$\delta = hf = \Delta E \quad (2.3.4)$$

For resonance to occur there is a need to ensure that the photon energy and the difference between the energy levels must be equal [38, 49]. The electromagnetic waves (for changing state from low to high) should have the same frequency as given in equation 2.3.5 which was derived from equations 2.3.3 and 2.3.4:

$$f = \gamma\beta_0 \tag{2.3.5}$$

where f is the angular frequency of the protons, γ the gyromagnetic ratio (a constant fixed for a specific nucleus) and β_0 the field strength. The equation 2.3.5 is known as the Larmor frequency equation

2.3.7 Measuring The Magnetic Resonance Signal

Exciting the hydrogen protons can be achieved through RF-signals of frequencies that are calculated using the Larmor frequency equation. The de-excitation can be measured analytically and conclusions can then be drawn from there. The signal from the hydrogen nuclei can be captured and measured through the use of Faraday's law of induction. The law states that a changing magnetic field (or changing magnetic field flux) induces an electric current in a loop of wire that is close to it [49]. If we are to take an example of the precessing transverse magnetization, the total effect of precessing dipoles creates the changing magnetic field. This is induced in a wire loop that is close to the precessing nuclei. The changing flux, in this case, is owing to the fact that the hydrogen dipoles will be precessing and doing so in phase [49]. Empirical results showed that if an RF receiver is placed close to the precessing hydrogen protons, it will pick up a signal that has a frequency that is similar to the frequency of precession of the hydrogen protons [49].

2.3.8 Longitudinal Relaxation

Longitudinal relaxation by definition is the recovery of longitudinal magnetization parallel to the external magnetic field β_0 commencing just after the application of the RF-signal. Tissue magnetization can be defined as an aggregate of all the hydrogen dipoles after a 90-degree RF-pulse. These dipoles (particles) can change orientation due to a number of reasons. For particles to change direction, they do so by interacting with the neighbouring molecules as

the particles will be tightly packed. As the nuclei are turning they have thermal interactions with other molecules in the tissue (also known as (AKA) lattice) [44, 49]. T1 is then described as the longitudinal relaxation that occurs between the high energy state hydrogen nuclei and the larger macromolecules in the same tissue. The interactions of the particles with their neighbours result in an increase in the number of nuclei in the low energy state and a decrease in the number of nuclei in the high energy state. T1 involves interactions between spins and the lattice, hence it is also referred to as the spin-lattice relaxation [44, 49]. The thermal interactions that occur, exclusively occur when the lattice is made up of larger molecules than the hydrogen particles. The larger molecules should have the capacity of holding the energy that is released by the hydrogen molecules. By losing energy hydrogen moves from the high energy state to the low energy state. The amount of energy released or absorbed by molecules is given by the Larmor equation. The energy transfer happens at a rate equal to one minus an exponential. The exponential will be a reflection of the longitudinal magnetization at time zero after a 90-degree pulse [44, 49].

The recovery rate is rapid at the beginning and slows down as time progresses. The rapid recovery is because at the beginning a lot of nuclei would be changing state. The slow rate at the end is owing to the fact that at the end there will be fewer dipoles changing state, that is the rate of recovery is directly proportional to the number of dipoles changing state. Equation 2.3.6 describes the rate of recovery during longitudinal relaxation after a 90-degree pulse [44, 49]:

$$M_{z,t} = M_{z,eq}(1 - e^{-tR1}) \quad (2.3.6)$$

where R1 is described as the longitudinal relaxation rate constant or spin-lattice relaxation rate constant. R1 is the inverse of T1 and T1 is defined as the time taken to de-excite 1/e of the excited dipoles. Figure 2.8 shows the graph of the amplitude height against time:

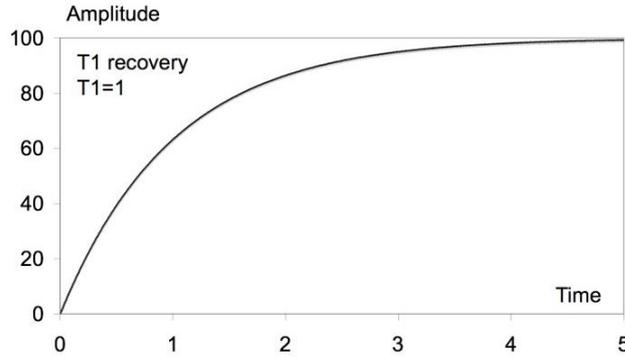


Figure 2.4: Recovery of the z-magnetization after 90-degree pulse [49]

The graph shows a steep gradient at the beginning that dies as time progresses. As stated earlier, the high rate is owing to a lot of dipoles changing state, while the slower rate after some time is because there will be fewer dipoles changing state. It takes an exceptionally long period for longitudinal relaxation to recover fully. It was then concluded to define longitudinal relaxation as the time taken for 63% of the total strength to be recovered. To shorten the time for recovery, substances like gadolinium are added to the tissue. These substances have the effect of shortening T1 as they act like an energy sponge [49].

2.3.9 Transverse Relaxation

In transverse relaxation, we consider the magnetization perpendicular to the z-axis or parallel to the XY-plane. In actual fact protons do not exactly align themselves parallel and anti-parallel to β_0 [44]. They will be precessing about the direction of β_0 so they would have components along the z-axis and components along the XY-plane as well but there is no net magnetization along the XY-plane though. If a pulse is sent along the x-axis, the vector will be restricted wholly to the y-axis. Consequently, the net effect of the pulse along the x-axis would be that the z-component of the NMV would be rotated 90-degrees into the y-axis. When the pulse is over, the union of the magnetization will decay in a transverse or spin-spin relaxation [44].

Naturally, hydrogen nuclei will have different magnetic field environments and hence precess differently in the magnetic field β_0 . When the RF pulse is introduced, they start precessing in-phase and after the pulse, they start returning to their steady-state. This results in a very fast decrease in transverse magnetization. The de-phasing that occurs is described by T2, which is described as the exponential loss of transverse magnetization after a 90-degree pulse.

Equation 2.3.7 shows the relationship that exists between the variables involved in transverse magnetization [49].

$$M_{xy,t} = M_{xy,0}e^{(-tR2)} = M_{xy,0}e^{(-t/T2)} \quad (2.3.7)$$

The graph in Figure 2.9 shows the decay for the transverse magnetization.

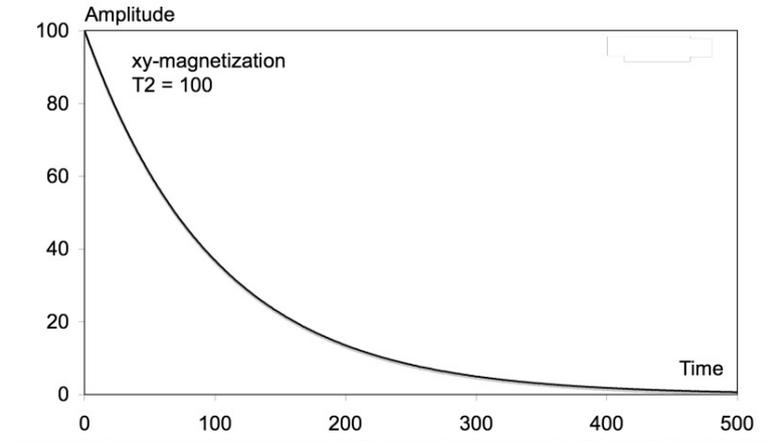


Figure 2.5: Recovery of the z-magnetization after 90-degree pulse [49]

A realistic picture of transverse magnetisation shows that each hydrogen atom will have a non-uniform magnetic field around it. These individual magnetic fields affect the magnetic fields of the dipoles in their vicinity. The combined effects of these fields on a particular atom is referred to as spin. The magnetic field interactions from other dipoles result in a decay referred to as T2-decay, spin-spin relaxation or true T2 de-phasing [49]. The true T2 de-phasing has been found to contain a lot of information that can be used for analytical purposes and therefore it is used extensively for patient diagnosis. It also takes a long time for all the strength to be recovered hence T2 can also be defined as the time it takes for 1/e or 37% of the original signal to be recovered. It can then be concluded that tissues with longer T2 values lose transverse magnetization slowly. Transverse magnetization decay can be expressed in an equation as shown in equation 2.3.8 [49]:

$$M_{xy}(t) = M_{xy}e^{-t/T2} \quad (2.3.8)$$

where t is the variable time, T2 is the transverse relaxation time constant. Empirical evidence showed that T2 is always shorter than T1 [44].

2.4 Datasets

The following section discusses the datasets considered for this dissertation. These sections describe the datasets used. Mainly focussing on where and how to get it and file formats used to store the data. The file formats commonly found in the discussion are MAT-lab, zipped, h5py, NIfTI, CSV, dicom,tsv and otl.

2.4.1 Common File Formats

The MAT-lab files are data files that are created when MAT-lab stores files to disk [71]. On the other hand, Zipped files are files that have been compressed to optimize disk space usage. These are important when downloading files or when the computer has limited space [76]. In addition, h5py files are file formats that store data in a format similar to NumPy arrays. The file formats store data in data structures that are immutable. The file type also supports a number of file format features that include compression, error detection and chunked I/O [29]. Also, the Neuroimaging Information Technology Infrastructure (NIfTI) is a file format that was designed to store fMRI and other medical image data. There are two types of NIfTI files, that is, NIfTI-1 AND NIfTI-2 where the latter is an update of the former [50]. We also have the Comma Separated Variables (CSV) and these are plain text files that comprise of a list of data items that are separated by commas. In some cases the commas are replaced by a semicolons, the files are then referred to as the character separated variables. These files are commonly used to transfer data between applications. The extension .csv is commonly used with the character separated variables [16]. In addition, there are the Digital Imaging and Communication in Medicine (DICOM) files and these are commonly used to store medical imaging data. DICOM files are also known to store patient identification data. The common extension for DICOM is .DCM [55]. There is also the Tab Separated Variables (.tsv) file format and these files are used to store data in a tabular form. The lines in the file represent records in the table and values of the record are separated by the tab character. The common file extension for tab separated variables are .tsv and .tab [39]. Finally we have the .OTL Files and these are generated when NoteTab saves files to disk. These files can be easily opened by several applications [97]. The following section discusses the datasets where these file formats are .

2.4.2 Kaggle Datasets

Kaggle provides competitions that are run in the machine learning community and participants can be from all walks of life. The competitions are free to join and are designed to promote interest in research areas that organizers think need more input from the machine learning community. The dataset had two classes, that is, a class with tumours and the second class did not have tumours. The dataset comprises of .jpg file format but the dataset provides no information about the origins of the data [59].

2.4.3 Figshare Datasets

The Figshare dataset is a compilation of 3064 T1-weighted contrast-enhanced images that were captured from 233 patients. The patients in this work had three types of tumours: meningioma, glioma and pituitary tumours. The data for the patients was captured and stored in a Mat-lab file format and saved in four different zipped files [25]. The data for each image in the zipped files were made from the struct construct that comprised of the following headings, one for meningioma, two for glioma, 3 for pituitary tumours, patient ID, image data and a vector string that contained the coordinates of discrete points on tumour border [25]. After downloading the dataset one could use SciPy to read the .mat files but certain versions like the 7.3 could not be read with SciPy. One could also use h5py to read the files or save the files using .mat earlier versions [81].

2.4.4 MICCAI Datasets

MICCAI dataset provided MRI brain datasets as part of a multi-modal brain tumor segmentation challenge aimed at promoting research in certain areas of research. The first competition was run in 2012 and since then there has been a competition every year and each competition had different tasks. In 2018, competitors were required to produce segmentation results for different types of glioma tumours and their sub-regions. After the segmentation process, participants were provided with extra information that they were expected to use to estimate the overall survival of the patient. In this competition 3T scanners were used for the acquisition of images and training, validation and test sets were released at different times during the life-

time of the competition. The first set released was the training dataset. The second set was the validation set but the ground truth was withheld for some time. Participants were given the training set at their request but the validation and testing sets were provided at the same time for all participants. The period between the release of the validation and test set was a month. In the dataset, both the training and validation datasets had four files, that is, ground truth, segmentation survival information, age and segmentation status [7]. All the subject data in the competition were in NIfTI (.nii.gz) file format and there would be a description of spin-lattice (T1), spin-lattice gadolinium-enhanced-weighted (T1Gd), post-contrast-spin-lattice-weighted (post-contrast-T1w), spin-spin-weighted (T2w) and Fluid-attenuated inversion recovery (FLAIR). Each image was manually segmented by between one and four individuals and accessed by neurologists with extensive experience. Additionally, the data had been preprocessed using BraTs protocols and these are found in [75].

2.4.5 Open Access Series Of Imaging Studies (OASIS) Datasets

OASIS seeks to make brain data freely available to those researchers that need it in an endeavour to make way for discoveries in basic and advanced neuroscience. The OASIS project is hosted by central.xnet.org [91] and it is a compilation of data from over 1000 participants that were collected over three decades. Of these participants (age range 42-95 years), 609 had no cognitive disorders while the other 489 participants did. The data collected from the participants were initially anonymized through the eradication of timestamps, all logs, ID-change, QC staff, the path to the raw data file and the raw data file. The dataset comprised of over 2000 sessions that included spin-lattice weighted (T1w), Susceptibility Weighted Imaging (SWI), fluid-Attenuated Inversion Recovery (FLAIR), ASL, time of flight, resting-state BOLD and spin-spin (T2w) images. Freesurfer was used to acquire surface measures and subcortical segmentation and details of these could be found on a downloadable CSV file [90]. The web page also made provisions for the data to be downloaded as a CSV file.

As of the writing of this document, OASIS-3 was the latest release from OASIS. The dataset made use of the longitudinal acquisition protocol, clinical, cognitive and biomarker dataset for normal ageing and Alzheimer’s disease. The data was available through the link given in [90]. One of the predecessors, OASIS-1 dataset had human brain T1-weighted cross-sectional dementia patients in the range (18,96) [69] While the other OASIS-2 dataset had human brain

MRI T1-weighted longitudinal images of dementia patients in the range (60,96) [68]. OASIS-3 comprised of human brain MRI and PET image longitudinal images of dementia patients in the range (42,95). The dataset also had 1912 Freesurfer processed outputs as well as a data dictionary [90]. The image data files were initially converted to NIfTI format by a format proposed in an article by [15]. The raw DICOM or IMA formatted images were initially converted to NIfTI through the function `dcm2nii`. The acquisition scanners were Siemens BioGraph NMR PET-MR 3T, Siemens TIM Trio 3T MTI, Siemens Sonata 1.5T and Siemens Vision 1.5T. Additionally, each image had the scanner information saved with the data [15].

2.4.6 ITK-SNAP

This dataset comprised of a zipped file that was made up of `.nii` brain MRI data files. This dataset was considered for this research but the licence disapproved its use for such purposes [105].

2.4.7 National Bioscience Data Center (NBDC Human Database)

The project NBDC was developed to ensure brain MRI data and other related files can be stored and easily retrieved for research purposes. It functioned by integrating data from multiple sites. The dataset downloaded from the project housed T1-weighted full brain images coupled with the associated clinical data for patients with schizophrenia, mood disorders (unipolar and bipolar disorder) and healthy controls acquired at three different locations in Japan. The data in the dataset was initially made anonymous and the quality of data was assessed before storing the data. As of the time of writing of the dissertation the project comprised of 176 cases, that is, normal control (129), schizophrenia (37), unipolar disorder (8), bipolar disorder (2) [82]. The data for the patients were arranged into five categories. The five categories were normal cases, schizophrenia cases, unipolar cases, bipolar cases, and types of MRI equipment used. The first four categories were tabulated and each table had four rows, that is, brain MRI, phantom MTI, subject's ID-phantom, ID-equipment, ID mapping table and clinical information. The brain MRI and phantom MRI data files were zipped files that contained NIfTI (`.nii`) files [82]. The raw subject's ID-phantom, ID-equipment, ID mapping table and clinical information rows contained information stored in tab separated variables

(.tsv) formats. Finally, the section on MRI equipment showed the details of the scanners that were used, that is, Philips, 3T and the modality were 3D-T1. Additionally, the NBDC website gave the links to download the .nii and .tsv files when needed. It also provides a data dictionary for clinical data bipolar disorder patients [82].

2.4.8 Alzheimer’s Disease Neuroimaging Initiative (ADNI)

ADNI is committed to the provision of gadgets and resources for disbanding, amalgamating, searching and making available a wide range of neuroscience data. It comprised of varied data from volunteers collected by researchers. The varied data was collected using a standard set of protocols and steps to eradicate discrepancies. The information on the dataset can easily be accessed by authorized investigators through the LONI Image and Data Archive (IDA) [54]. IDA houses brain MRI image data collected from an excess of 80 studies. The study’s main thrust was an evolution, advancement in age and the continuation of the specific disease. The software allows you to search and choose the option of the file format to download and the available options are archived file format, NIfTI and MINC file-formats [54].

2.4.9 Multiple Observations Across Networks (SIMON)

This project comprises of Brain MRI dataset of a single male in the age range (29,46). Each patient being scanned in 73 sessions at different sites with varied scanner models. Each of the sessions includes at least one anatomical T1-weighted image, in addition to other modalities such as T1, T2- and PD-weighted, diffusion, resting-state fMRI, susceptibility, arterial spin labelling. During these sessions, there were wide-ranging modalities. The datasets in this project can be easily downloaded from Amazon Web Services S3 bucket. The files from S3 can only be accessed through the use of HyperText Transfer Protocol (HTTP). In accessing the data one has to determine the correct Uniform Resource Locator (URL) and use it with an HTTP platform to access the necessary data. Wild-cards are common tools for use when searching data but they can not be used to access required data from S3 bucket, one has to use the actual name in a given search. There are many programs (eg Cyberduck) that can process S3 on the indigenous machine, these also allow individuals to browse through the data through the use of a browser. In addition to all this, the dataset provides a .csv

file that houses additional information about the 73 patients. The additional information includes dominant handedness, age, scanner model name, scanner manufacturer, the institution where the image was taken, the date of acquisition and a column for the session number. The dataset is comprised of archive files and these archive files contain .nii data files [104, 108].

2.4.10 Reshare

This project provides Brain MRI data from 22 patients that were known to have brain tumours. The main thrust in acquiring the data was to access the practicality and usability of fMRI in brain tumour surgical planning [96]. During image acquisition the radiographers made use of an 8 channel phased-grey head coil GE Sigma HDxt 1.5T scanner. Originally the scanner produced DICOM files and these were anonymized by getting rid of any information that might be used to identify individuals in the study. The DICOM files were then converted into NIfTI files using the `dcm2nii` function hence the imaging data stored in the zipped files are in the NIfTI file format. The zipped files were then stored in a .tar file format which is the file that the user downloads. Also, the dataset had an excel file (`metadata.xlsx`) attached and this file housed extra data about the dataset [96]. The initial sheet in the excel file is entitled ‘MRI_DES_data’ and comprised of patient IDs (sub-IDs), pathology, tumour location, estimates of grey/white/CSF and tumour volumes, any electrical stimulation if any were conducted and last but not least the imaging modality (T1, T2, fMRI or DTI) used. The second sheet entitled ‘clinical_data’ comprises of the demographics of the patients involved, dominant hand information and results for many clinical examinations conducted [96]. The ‘nil’ symbol used in both sheets indicates missing values or information. [96].

2.4.11 LORIS

LORIS is a software designed to aid researchers through the provision of tools designed to govern, process and distribute colossal volumes of data. Its infrastructural components include a database (NoSQL) and the database can operate effectively and efficiently even when presented with large datasets. The database also provides the users with a unique opportunity of using structured query language to access specific customized data. This data can then be pre-processed for downloading and downloaded to a locally provided storage device.

Also, LORIS enriches its experience by providing only relevant data to different users through permissions depending on who they are and the data they should have access to [65]. The infrastructure can also be customized to give more detail and give additional information such as the original data, the data file formats used and the imaging modalities used. During uploading, the infrastructure is designed to be compatible with several file formats such as NIFTI, MINC and DICOM amongst others and it can also be customized to conduct quality control before any imaging data uploading. LORIS has a sub-project entitled BigBrain and the data from BigBrain can be accessed through LORIS or from an FTP (File Transfer Protocol). The data store in BigBrain related to MRI brain was in .nii file format [65].

2.4.12 NeuroImaging Tools & Resources Collaboratory (NITRC)

Researchers in neuroimaging can freely access resources from the NITRC website upon registration to the site. The website is hosted by the Center for Research in Biological Systems (CRBS) which is a research organ in the University of California San Diego (UCSD). It was initialised in the year 2007 and continued running until the writing of this dissertation. The function of CRBS is to enable collaboration between computer scientist/information technology expert and experts in the fields of chemistry, medicine and other science-related fields [85]. The NITRC project offers comprehensive information about ways of increasing the scope of neurological related software resources and data. In general, NITRC has three sub-groups, that is, the resource registry (NITRC-R), Image Repository (NITRC-IR) and Computational Environment (NITRC-CE). NITRC-R's main thrust is the provision of software tools and other necessary resources to ensure that previously funded neuroimaging projects are made available to the neuroimaging community at large [89]. It also provides a platform for the free exchange of ideas between researchers and developers. On the other-hand NITRC-CE is a cloud-based platform that is free to download. After the download and installation have been completed, the user can make use of the platform but pay for the use of the platform when they do use it. The platform comes pre-configured with some popular neuroimaging software, for instance, ANTs, AFNI, FSL and FreeSurfer. The user can easily make use of the available software to analyze their data without the need to configure the given software hence saving them on computational resources and time. The platform has provisions for the user to configure the software they need as an addition to those already provided for [89]. Last but not least, NITRC-IR is synonymous with NITRC-R but while NITRC-R focuses on

tools NITRC-IR focuses on data. It makes it possible to download free of charge publicly available datasets. NITRC houses several files with data formats ranging from DICOM to BIDS for normal subjects and those with conditions under study. The conditions normally studied include Parkinson's, autism, ADHD and Schizophrenia [89].

Internet Brain Segmentation Repository (IBSR) was a sub-section of NITRC-IR and it offered MRI brain data along with segmentation results generated by experts in the field. The project aims to promote the development of segmentation methods and assessment. NITRC has had several versions over the years, one of the releases was CANDIShare Schiz-Bull_2008 version 1.2 (v1.2) and this version was based on 103 subjects. In this release, the organisers added the Certified Medical Assistant (CMA) general segmentation for all the subjects [83]. Version 1.2 also corrected the bug that was found in version 1.1. Also, v1.2 includes the linear and non-linear registrations for each subject. These comprised of MRI images and the segmented results for all the four categories that were involved. The categories considered were the control group (29 subjects), the group with Schizophrenia Spectrum (20 subjects), the group with Bipolar Disorder coupled with Psychosis (19 subjects) and finally the group with Bipolar Disorder without Psychosis (35 subjects). The dataset had a folder for each category and hence there are four folders and each folder was a gzipped tar folder. Each of the four files contains sub-folders for each imaged subject [83]. These sub-folders housed CMA preprocessed image data and the results of the general segmentation and all the data were in the NIfTI (.nii.gz) file format. CMA included protocols for positional normalization and bias field correction. The protocols were there to ensure that the images were in the standard orientation of the Talairach coordinate space in the accurate segmentation of the lateral ventricles and caudate nucleus in MR brain images using anatomical driven histograms. Additionally, the dataset also provided group basic demographics in a separate .csv file. The information contained in the .csv file included the age, gender and dominant handedness of the subjects [83].

For one to get access to NITRC resources one has to go through a very simple and straight forward registration process and after that, the publicly available datasets are fairly easy to download and free. It took us less than 5 minutes to download v1.1 and v1.2 which was exceptionally fast considering that a dual core was being used with four gigabytes Random Access Memory (RAM) on Ethernet. On the other hand, it has been observed over the years that at times some files are not accessible, for instance, Olmos and Tero in 2013 could not

access version 1.1 (v1.1) and v1.2 for the demographic information [87]. Researchers also pointed out that even though the segmentation results were in NIfTI file format the original acquisition data was in .otl file format and this brought about minor differences at the voxel level. The .otl file formats might be more difficult to read but they provide better accuracy, on the other hand, the preferred NIfTI file formats are easier for the general public to access and read hence they are mostly used by the research community but are less accurate [88].

The preceding sections discussed the various datasets that could be used for brain image segmentation. The information discussed included the image data-types, acquisition protocols and other crucial aspects of datasets. The following sections discuss brain image segmentation and some of the algorithms that have been proposed this far.

2.5 Brain MRI Segmentation

Image segmentation by definition is the parcellation of an image into different regions of interest. Medical image segmentation was traditionally done manually by seasoned medical practitioners. The problem with manual segmentation was that its outputs were difficult to reproduce, susceptible to human error, time-consuming and suffered from user variability. On the other hand, radiological laboratories frequently had large numbers of medical images to process [66]. This made manual segmentation inefficient and ineffective in a fast-paced environment like the hospital. Also, in image segmentation, it is critical to understand how the different organs work as units and systems [66]. Image segmentation is crucial in studying the neurological development of soft tissues like the brain and/or spinal cord, checking if any differences exist between healthy and abnormal anatomical structures and the mapping that can be done between functional activation and brain anatomy [125]. Segmentation by definition involves either organ and/or tissue characterization depending on the clinical procedure that is about to be undertaken [125].

A brain stroke is also known as cerebrovascular or cerebral circulation abnormalities (ischemia) which leads to necrosis. Stroke has non-reversible effects on the human body so there is a need for early diagnosis and treatment [26]. Also, white matter lesions are common in neural

disorders such as multiple sclerosis, small vessel disease and anaemia [124]. Identification and measurement of regions of interest is an important step in the identification of neurological disorders [124, 26]. The main sulci are used to differentiate functionally different regions in the brain. The shape and size can be used to diagnose neurological disorders. According to Borne *et al.* [17] segmenting sulci is a daunting task, because it's a highly irregular structure and also each brain contains about 120 different types of sulci [17]. Also, segmentation in medical imaging is a challenge because of the rarity of labelled images and the fact that medical images are normally very large that is, 3D [17].

The segmentation of arteries using 3D time-of-flight (TOF) magnetic resonance angiography of the brain allows for the measurement of their volume, which is important in studies focusing on understanding ageing. It is a challenging task to segment the arteries because they are a dense complex network and weak signals generated from slow-moving blood [24]. The hippocampus is responsible for the learning process, long and short term memory including memory management for daily activities. The segmentation of the hippocampus can assist in the diagnosis of coexistent temporal lobe epilepsy [21]. Hippocampus is also the region of interest in studying the progression of Alzheimer's disease. The rate of tissue decay of the hippocampus can be used to deduce the progression of AD. The structural changes of the hippocampus are also critical biomarkers for dementia [11]. The hippocampus has faint edges and overlapping intensities with its neighbouring structures, this makes the segmentation more difficult [11, 18, 37, 111]. The authors in [36], also pointed out that Alzheimer's is a type of dementia that affects a significant population of those who are above 80 years of age. AD seems to be highly correlated with symptoms such as cognitive impairments and attitudinal changes.

The segmentation of sub-thalamus nucleus and globus pallidus internal and Substantia Nigra are important for the diagnosis of diseases such as dystonia and Parkinson's diseases respectively [77]. The study of the hippocampus has attracted attention in medical image segmentation due to its connection with memory and AD [12]. The treatments that are currently available for AD are early treatments hence it is important to have an early diagnosis. Segmentation is a challenging task due to noise, differences in image acquisition protocols, inhomogeneity, differences in intensity profiles, differences in scanner models, partial volume effects, anatomical differences between subjects, magnetic field non-uniformities and low contrast between tissues [2, 8, 33]. Different components of the human body (water, bone, etc.)

exhibit different properties when exposed to different environments. Image acquisition exploits these differences to come up with the required image. The modalities in common use are x-ray, computed tomography, magnetic resonance imaging, ultrasound scan and other important modalities as stated in the work by Bandhyopadhyay and Tuhin [8]. X-ray and other modalities like computer assistance tomography (CT) have high levels of radiation and these have well-known side effects.

Segmentation of sub-cortical structures is a daunting task as they are small and normally have varied shapes and structures and they also have unclear boundaries are missing in same images[9]. Manual segmentation is time-consuming as compared to the automated and semi-automated algorithms [79]. Additionally, machine learning algorithms are known to perform exceptionally well in subjects that have approximately the same intensity values and under-perform when presented with data with differing intensity values. The differences in intensities are normally due to different scanners or imaging protocols amongst other reasons [33]. Another thorny issue in image segmentation especially for ageing people that have neurodegenerative disorders is the existence of white matter hyperintensities (WMH) which contribute to cognitive defects. These are regions of the brain that have a boosted signal appearing on T2 scans and FLAIR MRI scans and these appear as WMH in T1w images. They give a similar intensity profile to that of the grey matter and can result in the regions being erroneously classified as a grey matter regions [33]. .

Medical image segmentation is crucial as a pre-requisite or core component for many clinical processes. Some of the processes include surgical planning, abnormal organ development detection and post-surgical assessment. Almost all the image processing activities that are done are digital. In image processing the acquisition, processing, storing and analysis have all become digitalized. Several strides have been made in developing the technology used for image processing. Even though a lot of ground has been covered in the field, it remains very difficult to accurately segment medical images [8]. Many semi and fully automated algorithms have been developed. All these algorithms attempt to segment brain images within certain limits of accuracy. The algorithm's error rate is owing to the unidentified and irregular noise source, inhomogeneity and weak contrast between tissues especially on the boundaries between tissues. Medical images normally contain complex regions of interest and have to be segmented with a high degree of accuracy for meaningful diagnosis of neurological and psychiatric dis-

orders to take place, for the assessment of treatment effectiveness and as a prerequisite for surgical planning [8]. The following sections cover discussions on the literature review related to the segmentation of MRI brain images of humans, mainly focusing on those that made use of CNNs. The section commences with general discussions on concepts related to neural networks (NN).

2.6 Neural Networks (NN)

2.6.1 Introduction

A NN is a learning algorithm that forms the major component of most deep learning algorithms. Litjens *et al.* [64] gave a summary of the mathematical definitions and related equations for a neural network. These networks form the basis of artificial neural networks (ANNs), and ANNs are the predecessors of CNNs. CNN's are almost the same as ANNs. In that, they are also made up of neurons, and they learn to optimize. In effect, they are calculating machines that are comprised of several neurons working together as a system to learn inputs to produce an output [92]. A summary of the mathematical constructs involved with convolutions was shown in the study carried out by Wu [121]. CNN's have an advantage over ANNs in that they are better able to handle larger problems as compared to ANN hence they have been applied in solving numerous real-life problems [3, 92]. CNN's are made up of mainly three types of layers, that is convolutional layers, pooling layers, and fully-connected layers. When the layers are stacked in any logical order, they result in a CNN architecture. Figure 2.10 provides a typical example of a CNN architecture.

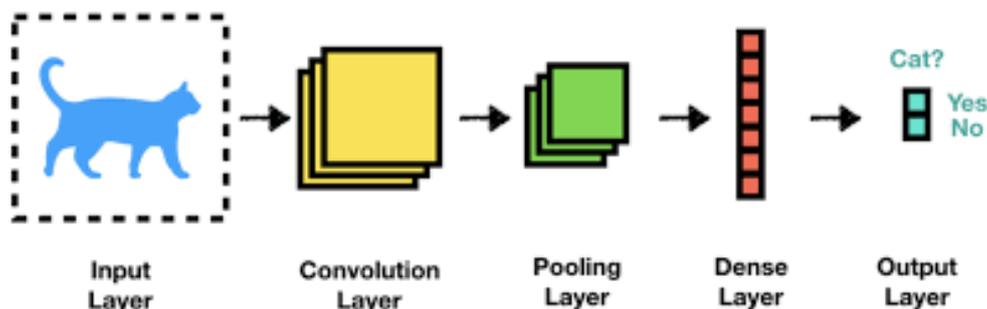


Figure 2.6: A diagram showing the structure of a typical CNN architecture [4]

Figure 2.10 shows a CNN architecture, with the input, convolutional, pooling, dense and

output layers. The neural network in the diagram could be used to classify animals in the images as either a cat or non-cat animal. The convolutional layer establishes the yields of neurons connected to the input by calculating the scalar product of weights and the region connected to the input volume. The parameters for this layer are centred around the use of learnable kernels. Kernels are integer matrices of a given size, but they are generally small in space dimensionality [92]. When the input data goes through the convolution, each of its filters goes across the input's entirety, producing activation maps. Each of the kernels will have an interrelated activation map, and these maps are stacked, resulting in a convolutional layer. CNN can minimize the intricacy of a model by optimizing its output. The optimization can be achieved through three hyperparameters, that is stride, zero paddings, and depth [92, 121]. The study carried out by O'Shea *et al.* [92] showed how these hyperparameters could be set or activated. Activation functions such as Rectified Linear Units (ReLU) or sigmoid would be used on outputs to give an output in the expected range, and this will be used as input for the next input [92].

Pooling layers would be used to reduce the number of incidents in the given input and further reduce the number of parameters within the activation function. The pooling layer will act on each of the activation maps that it receives as input. It would scale the dimensionality of the activation maps, and this would usually be achieved through the 'MAX'function as the commonly used pooling function is max-pooling. Due to the pooling layer's destructive nature, a kernel size of more than three is not advised, as this will result in decreased performance of the model [92]. In the fully connected layer, all the neurons in a layer are connected to the next layer's neurons. There are also other types of layers, for instance, dropout, softmax, and other layers used in a CNN architecture.

2.6.2 CNN Applications To Brain Segmentation

Only a few layers can be used in making a CNN architecture, but the actual architecture is very dynamic. Several algorithms have been proposed for medical image segmentation, and these include GANs, encoder-decoder, deep medic, Densenet, ensemble, patch-based CNN, transfer learning, U-Net, V-Net, Y-Net, and VoxResNet just to mention a few.

Over the past few decades, researchers have proposed many approaches to segment the brain into varied regions of interest. The approaches can be categorized in several ways, but we

categorized them as a sliding window (patch-based) and fully convolutional neural networks (Fully CNN) [42]. Sliding window algorithms make use of patch-based segmentation and still use convolutional neural networks (CNN) in their architecture while fully CNN does not use patches. Fully CNN is predominantly CNN that some researchers have used as hybrids.

The study carried out by Geceri and Evgin [42] outlined the advantages and disadvantages of each group. They went on to point out that most fully CNNs were based on the encoder and decoder architecture. They concluded that fully CNNs were more efficient than sliding windows regarding processing time and computational complexity, but fully CNN had more parameters. They went on to outline the fact that many fully CNN had been proposed, citing that some authors had used cascaded hybrids of fully CNN, and other algorithms made use of dense pooling layers [42]. They went on to cite U-Net as one of the most commonly adopted architecture. In researches, it has been proven that increasing the depth results in rich image features, which also enhances the learning of the NN, but unfortunately, this results in reduced accuracy rates. Their study also indicated that ResNets were able to take advantage of deeper networks. Hybride architectures have been developed to unify the strength of many base architectures, and the studies done by Goceri and Evgin gave some of these architectures [42].

The review by Geceri and Evgin summarised related reviews. It went on to show that some authors had zeroed in on the segmentation of the medical images, others zeroed in on only the technical aspect of segmentation, other authors focused on more aspects including registration and classification, other authors zeroed in on retinal blood vessels segmentation. In contrast, other authors zeroed in on semantic segmentation only while others have zeroed in on weakly-supervised semantic segmentation [42]. In addition, the study carried out by Litjens *et al.* [64, 102] made a comprehensive review of the methodologies and application-driven studies carried out in all fields in image processing. The review only focused on deep learning algorithms [64, 102].

2.6.3 Sliding Windows

These algorithms use patches as inputs instead of complete brain image slices or volumes. The studies in [9, 10, 17, 18, 26, 32, 43, 53, 62, 123, 124] used patch-based segmentation.

There are two types of researches that have been connected to sliding windows. First type focussed on experimenting with different patch sizes to determine the one that optimizes NN's segmentation accuracy [123]. The second focussed on implementing sliding windows in brain image segmentation. In addition, some algorithms use 2D [32, 43, 123, 124], 3D [18] while others use 2.5D [123] sliding windows. The most commonly used patch sizes are 13, 32 and 64 while patches of size 8, 16 and 60 have also been used. Patch size 13 was said to have the best segmentation output but was slow [32, 123, 124]. Also, some researchers used patches with prior label information [9, 10]. In addition, researcher pointed out that, larger patches produce better results because of the more information but smaller sized patches were known to have higher accuracy rates. Researcher have therefore proposed models that use both small and large patches [124]. Also, sliding windows are used with various architectures including DenseNet and the LeCun architectures[17, 43]. The following section discussed the fully connected convolutional neural network.

2.6.4 Fully Convolutional Neural Network (Fully-CNN)

The fully connected CNN neural networks proposed have several variations, and some are hybrids of other algorithms. The studies carried out by [5, 57, 99, 60] made use of fully connected CNN. Some of these algorithms were hybrides of well known architectures such as V-Net [94, 129], U-Net [21, 37, 48, 45, 63, 111, 122], Y-Net [24], Ensemble [2, 11], Encoder-Decoder architecture and Generative Adversarial Networks.

2.6.4.1 GANs

Research on GANs showed that they could be instrumental in medical image data augmentation and making image data anonymous. In addition a trained GAN could be used for continuous learning in decentralized datasets or in the integration of data from different institutions without fear of confidentiality violation [57, 99].

2.6.4.2 Encoder-Decoder Architectures

The encoder-decoder architecture is one of the commonly used architectures in medical image segmentation. The encoder-decoder architecture uses CNN as a backbone. Some of the common encoder-decoder architectures include V-Net, U-Net, and many more hybrids since its a common and very effective NN architecture. The following sections discussed some of the typical encoder-decoder based medical image segmentation proposed algorithms.

2.6.4.3 U-Net

The U-Net architecture has proven itself to be a very instrumental architecture in medical image segmentation hence many algorithms have been proposed based on it. The research studies in [21, 37, 48, 45, 63, 111, 122] used the U-Net architecture. Some authors made use of the U-Net in conjunction with ensembling [37], prior information [48], patches, a second U-Net and ensemble methods [30], 3D contextual information [111] with very promising results. On the other hand, some authors have shown that U-Net used in conjunction with a large step size (during patch generation) prove to be as effective as one with a small step size. Also, the U-Net produces better visual segmentation results as compared to a number of other algorithms [37, 48]. In addition, of the U-Net architectures have been known to be very fast [111].

2.6.4.4 Ensemble

Ensemble methods have gained traction in the past as they have had promising results. They combine different models and aggregate the different models' results to achieve the medical image segmentation task [2, 11]. Research has shown that by comparison three channel ensemble classifiers perform better than one channel [2].

2.6.5 Section Conclusion

The sections above discussed various algorithms that have been implemented for brain image segmentation. In the literature, some researchers proposed the use of conditional random fields (CRF) or recurrent networks to aid in image processing [94]. In addition, some authors

believed quantization introduced a regularizing effect to the NN, resulting in an increase in segmentation results. On the other hand some authors suggested the use of intensity correction in areas where the images have not been preprocessed. Authors in other texts also believed that the use of sliding windows mitigated information loss. In addition, research has shown that the use of parallel patch generation to maximize efficiency and effectiveness of a NN model in medical image segmentation. Also, to optimize model output researchers should consider using different set of parameters (learning rate, loss functions, etc) and choosing the set that optimizes the segmentation output [24].

2.7 Conclusion

The brain is responsible for coordinating the operations of the entire human body. Different parts of the brain have different responsibilities in the body. For the brain to perform its several functions effectively and efficiently, it has to be devoid of defects. Any defects in the brain usually result in severe neurological disorders that are difficult, if not impossible, to reverse. Early diagnosis and treatment of any neurological defects would go a long way in reducing the brain's effects and the body in general. For any meaningful clinical decisions to be made, there is a need to accurately segment the human brain. Researchers around the globe have proposed several human brain MRI segmentation algorithms over the past few decades. In the present study, we gravitate towards the use of CNNs in MRI brain image segmentation, which then boiled down to patch-based segmentation of the human brain MRI. We choose to gravitate towards CNN because of the promising results they have shown for human brain segmentation in several studies.

3. Methods And Techniques

3.1 Introduction

This chapter covers the objectives of the present study. It gives a summary of the machine learning tools describes the NN architecture used in the study. The chapter goes on to discuss dataset preprocessing steps for the study. Additionally, it discusses segmentation classes provided in the dataset and classes considered in the present study and how they are derived. It also discusses the methods used in patch generation and how the dataset is split into training and validation datasets. The chapter also goes on to discuss 3D and 2D pixel/voxel voting and their comparison. It also describes the experiments that were carried out in the present study. The chapter concludes with a discussion of the evaluation metrics that are considered in evaluating the performance of the proposed algorithms.

3.2 Experimental Objectives

This section of the present study focuses on the experimental objectives and these are:

1. Effectively pre-process MRI brain images for component-based segmentation,
2. Train a CNN to segment a ROI using MRI patches
3. Evaluate the achieved segmentation performance

3.3 Machine Learning Tools Used In The Research

3.3.1 Programming Platform

Several image processing tools are used during preprocessing, model development, training and evaluation. Medical image training and evaluation is known to take extended periods hence

alternatives to Anaconda were considered and some of the popular choices were Google’s Colaboration, Azure Notebooks, Kaggle, Amazon SageMaker, IBM DataPlatform Notebooks and Jupyter Notebooks. Google’s Collaboration (a.k.a Colab) was the preferred choice because it comes configured with most required modules, gives access to resources such as GPUs or TPUs and is a familiar platform for researchers. Colab is an online framework that allows users to compile and run Python through the browser. The framework uses the Jupyter Notebook interface and required no system set up to be done for one to develop or run Python code. It provided free access to computing resources including GPUs, TPUs etc. even though there were limitations due to demand from users. The GPUs and TPUs allocation algorithm prioritised users who used Colab interactively as opposed to users with models that took a long time running. The algorithm also prioritised users that recently used fewer resources. Colab had a maximum runtime of 12 hours and the available memory varied with demand. Colab comes with a lot of packages and these included NumPy, TensorFlow, Keras, pandas, Babel, OpenCV, Pillow, pip-tools, Sci-kit learn and torch only to mention a few. The platform also enables users to install other packages, in the present study we install Mayavi and other packages as per need. Additionally, during training GPUs are activated in the run-time environment. Also, the packages discussed in the following sections were installed on the platform.

3.3.1.1 Keras

The proposed model is a Keras based model, hence the layers, optimizers and regularizers are from Keras. Keras is used to import important modules like SoftMax, Activation, Input, Dense, Flatten, Dropout, Reshape, Conv2D, MaxPooling2D, UpSampling, Conv2DTranspose and image. Most of these modules are then used in model development. The model also uses the `to_categorical` function from the `utils` module from Keras, to convert ground truth classes into categorical classes.

3.3.1.2 Tensorflow

The proposed algorithms are implemented in Tensorflow. Firstly, the actual inputs to the NN are patches hence Tensorflow is used in patch extraction. It is also used in `tensor_to_array` conversion using the `eval()` function. Tensorflow is also used in the extraction of the central

pixel values for ground truth patches using `crop_to_bounding_box()` function from Tensorflow. Additionally, it is used to reshape the NumPy array to suit the requirements of the next stage in the processing cycle or the requirements of the researchers.

3.3.1.3 Matplotlib

Matplotlib is used alternatively with Mayavi for visualizing the brain or its components. Pyplot from Matplotlib is also used in visualizing the performance of the NN through plotting the graphs of evaluation metric such as `training_accuracy` and `validation_loss`. Pyplot is also used in plotting the cropped slices of both the input and ground truth images in visualising their morphology.

3.3.1.4 SKLearn

The CANDI dataset comes as a single dataset hence there was a need to split it into training and validation sets. Sklearn is used for splitting the provided dataset. Also, there is a need to encode the ground truth classes hence Sklearn is also used for the encoding.

3.3.1.5 Numpy

Numpy is extensively used in the development of the proposed algorithms. It is used in the conversion of image data from its original format. The image data format is originally `dataobjs` hence Numpy is used in converting `dataobjs` into Numpy arrays in the preprocessing stages of model development. It is also used in the integration of the missing channel and batch information to image NumPy arrays.

3.3.1.6 ADDED Packages-Mayavi

During the present study's experimental work it is important to visualise the data either for analysis or verification purposes. Mayavi [72] gives a set of tools for 3D interactive visualization of the brain or its components. It is built on top of several libraries and these include

VTK (Visualisation ToolKit), NumPy, Traits and TVTK and details of these can be found in the work done by Ramachandran *et al.* [95]. Mayavi is chosen because it works seamlessly with our data which has been converted to NumPy arrays. The following sections discuss some of the hyper-parameters used in the study.

3.3.1.7 Machine learning Hyper-parameters

Stochastic Gradient Descent (SGD)

In most machine learning problems, there would be an input vector x and an output vector y that would normally be adopted from a probability distribution $P(x,y)$. The probability of y given a certain value of $x(P(y|x))$ then would give the required relationship between the vectors x and y [119]. When \hat{y} is used to represent an estimate of y then $\hat{y} - y$ would give the error and the expression $\Gamma(\hat{y} - y)$ would be referred to as the error function. The objective function is to minimize the function given in equation 3.5.1.

$$E(g) = \frac{1}{n} \sum_{i=1}^n \Gamma(g(x_i), y_i) = \mathbb{E}_n[\Gamma(g(x), y_i)] \quad (3.3.1)$$

The linear parameterization of Γ is done by w , where w would be an element of \mathbb{R}^d and d would be the size of the vector x . SGD would make use of one sample at a time during SGD iterations. After each iteration, it would update the weight vector w by making use of time-dependent factors and this would result in the equation 3.5.2:

$$w_{t+1} = w_t - \left[\frac{\lambda}{t} \frac{\sigma}{\delta w} \Gamma(g_t(x_t), y_t) \right] \quad (3.3.2)$$

where λ would be the gain factor the function uses to update the weight at time t at the end of every iteration.

Learning Rate, Decay And Momentum

SGD would make use of the back-propagation of errors algorithm to update weights. The rate at which weights are updated is referred to as the step-size or the ‘learning’rate (lr). Learning rate (lr) is a SGD hyperparameter that has a value between zero and one. Learning rate

(lr) would be responsible for determining the corresponding change in the model given the estimated errors during weight update [19]. Learning rates should be selected with wisdom as an extremely large value of lr forces the model to converge prematurely resulting in a sub-optimal solution, on the other hand, a value too small would result in the model getting stuck in a local minima [19]. If the NN is implemented with momentum then the update of the weights would consider the current gradient as well as the previous updates to the weights. This would result in an accelerated convergence and slim chances of being stuck in local minima. Momentum also aids the parameter vector to build velocity in any given direction and it would do that at a constant gradient to avoid any oscillations. The most common momentum values are those that are close to 1, such as 0.99 [27]. In Keras implementation, the parameters momentum and decay are defaulted to zero and consequently learning is constant by default. Learning rate decay would make lr a variable whose current value is calculated using equation 3.5.3:

$$lr_{\text{new}} = lr_{\text{starting}} - \mathbf{B}lr_{\text{current}} \quad (3.3.3)$$

where $\mathbf{B} = 1/(\text{decay} + \text{iteration})$, lr_{new} is the new learning rate, lr_{starting} the initial learning rate, lr_{current} the current learning rate, decay the decay rate and iteration the number of iterations that have run thus far. Additionally, the SGD class is implemented in Keras in the optimizer class and it provides decay as one of its arguments. Also, equation 3.5.3 shows that the learning rate would be a non-linear function. It also shows that Learning rate (lr) and decay are inversely proportional to each other. A high value for decay would result in the lr quickly moving to a very small value. Consequently, a high value of decay would therefore not be advised as this would increase the chances of the model being stuck in local minima.

3.3.2 Convolutional Neural Networks Architecture

The model NN architecture proposed for the present study is an extension of the work done by Cui *et al.* [32] and its structure is as shown in Figure 3.1.

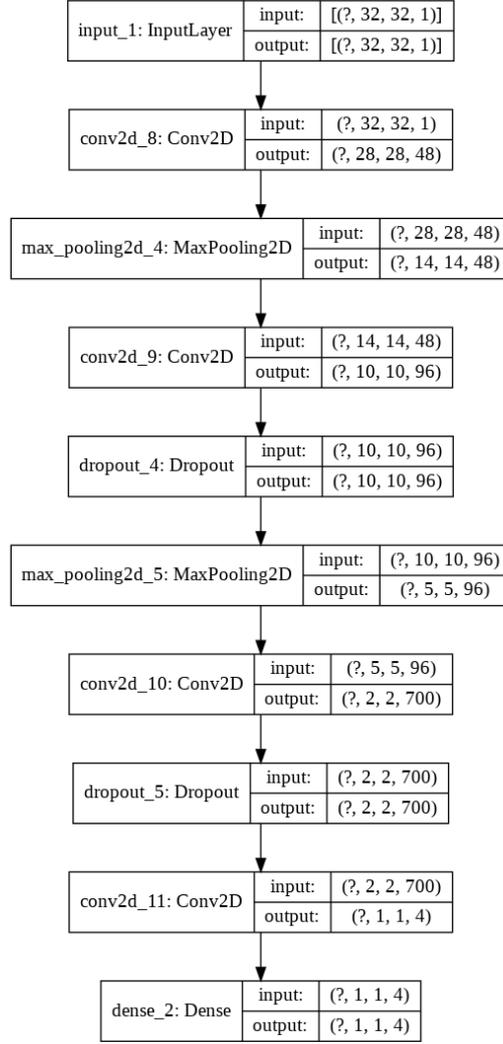


Figure 3.1: The neural network architecture for the proposed algorithm

In the model, all the convolutional layers have the `use_bias` parameter set to ‘True’ and the bias-initializers initialized to zero. In addition, ReLU activation function is used in each convolutional layer. Also, RandomNormal kernel initializers are initialized with mean 0.0, standard deviation 0.01 and no seeds set. The general equation for a convolutional layer is as shown in 3.2.1:

$$y_j = \left(\sum_i m_{ij} x_i \right) + n_j \quad (3.3.4)$$

where m_{ij} represents the i^{th} filter for the j^{th} feature maps and x_i represents the input to the convolutional layer and n_j represents the j^{th} filter bias [32]. The convolutional layer has three other important parameters, that is, `filter_size`, `stride` and `padding`. `Filter_size` gives the size of the filter used during convolution, `stride` gives the step size for the convolution and `padding` allowed the implementer to either implement padding or not. When padding is implemented its value is either set to `same` or `valid`. In the present study, `padding` is set to ‘valid’ and the

other parameters are maintained in their default state.

There is only one Dense layer and in this layer the parameters `use_bias`, `bias_initializer` and `kernel_initializer` are initialised to the same values as in the convolutional layer. It, however, uses the Sigmoid instead of the ReLU activation function. The rest of the layers and parameters are as discussed in the following paragraphs.

The input images to the NN are patches of size $32 * 32$. Greyscale is used in the present study. Table 3.1 gives a general summary of the layers used in the proposed model. It shows the 9 layers of the model, that is, 4 convolutional layers (Conv), 2 pooling layers (Pool), 2 Drop_out layers (Drop) and a Dense layer (Dense).

Table 3.1: Table showing a summary of the layers of the proposed NN model

Layer number	Layer	Layer type	Number of feature maps	input-size	Output-size
1	Conv2D	Conv	48	32	28
2	MaxPooling2D	Pool	48	28	14
3	Conv2D	conv	96	14	10
4	Drop_out	Drop	96	10	10
5	MaxPooling2D	Pool	96	10	5
6	Conv2D	Conv	700	5	2
7	Drop_out	Drop	700	2	2
8	Conv2D	Conv	4	2	1
9	Dense	Dense	4	1	1

Table 3.1 shows the types of layers, the number of feature maps used in each layer, the image input and output size for each layer. In the table the inputs to a layer are the outputs of the previous layer. The output size of each layer can be calculated by the formula shown in equation 3.2.2:

$$Output_size = \left\lceil \frac{n + 2p - f}{s} + 1 \right\rceil \quad (3.3.5)$$

where n is the input size, p the padding size, f the size of the filter and s the stride size. In addition, Algorithm 1 [1] shows a simple implementation of the Convolution algorithm.

Algorithm 1: 2D Convolution Algorithm [1]

Result: Convolved Image Matrix

```
1 hoplx = (oplx + 1)/2;
2 hoplx = (oplx + 1)/2;
3 for iy = 0; iy < ny; iy++ do
4   starty = MAX(iy-hoply+1, 0);
5   endy = MIN(iy+hoply, ny);
6   for ix = 0; ix < nx; ix++ do
7     startx = MAX(ix-hoplx+1, 0);
8     endx = MIN(ix+hoplx, nx);
9     if velmod[iy*nx+ix] != c then
10      c = velmod[iy*nx+ix];
11      readtable2D(opx, om/c, hoplx, hoply, mode);
12    end
13    dumr = dumi = 0.0;
14    k = MAX(hoply-1-iy, 0);
15    for i = starty; i < endy; i++ do
16      l = MAX(hoplx-1-ix, 0);
17      for j = startx; j < endx; j++ do
18        dumr += data[i*nx+j].r*opx[k*oplx+1].r;
19        dumr += data[i*nx+j].i*opx[k*oplx+1].i;
20        dumi += data[i*nx+j].i*opx[k*oplx+1].r;
21        dumi -= data[i*nx+j].r*opx[k*oplx+1].i;
22      l++;
23    end
24    k++;
25  end
26  convr[iy*nx+ix].r = dumr;
27  convr[iy*nx+ix].i = dumi;
28 end
29
30 end
31
```

In the implementation, there are variables `oplx`, `oply`, `nx` and `ny`. The first two of the variables represent the length of the operator while the last two represent the sizes of the data. Additionally, `opx` and `velmod` are arrays, the former contains the operators of the convolution while later contains information on whether or not to load a new operator. All this information is stored in a table and the `readtable2D` function reads the appropriate value of the operator depending on the values of `c` and `om`.

In the proposed algorithm the layer is a convolutional layer with input images of size $(32 * 32)$ and the filters are of size $5 * 5$. Using equation 3.2.2 with $n = 32, p = 0, f = 5$ and $s = 1$, the outputs of the first convolutional layer are of size $(28 * 28)$. In the convolutional layer, the filters slide over the channel. Forty-eight filters are used in this case, this resulted in a total of $28 * 28 * 48(37632)$ feature maps that are produced and 48 images.

The second layer is a pooling layer. MaxPooling and AveragePooling are the most frequently used pooling layers in images processing. MaxPooling is selected for use in the present study and an example of its algorithm is as shown in Algorithm 2 [6].

Algorithm 2: Maxpooling Algorithm [6]

Input: Input Images

Output: Maxpooled Images

1 **Initialisation:** `feature_maps = []`

2 **Def** `max_pooling(inputs, pool_size):`

```

3   foreach i in range (0, len(inputs[feature_map]) - pool_size[0], pool_size[0]) do
4       foreach j in range (0, len(inputs[feature_map]) - pool_size[0], pool_size[0]) do
5           feature_maps[-1].append(np.array(max((inputs[feature_map][j:j+pool_
6               size[0], i:i+pool_size[0])).flatten()))
7   return feature_maps

```

8

The Maxpooling function takes as input the input images and the size of the pool to be used during the Maxpooling process. The for loops help in isolating the required values and then the `max()` function is used to determine the maximum value in a pool. Maxpooling has the effect of highlighting certain features in given feature maps. This pooling layer receives 48

images of size $(28 * 28)$ as inputs. Using formula 3.2.2 with $n = 28, p = 0, f = 2$ and $s = 2$, the output images are of size 14. This, in essence, produces 48 images of size $(14 * 14)$ or a total of 9 408 feature maps. After the MaxPooling layer, the third layer is a convolutional layer. This layer receives 48 input images of size $(14 * 14)$. It makes use of 96 filters and hence produces 96 images as output. Using formula 3.2.2 with $n = 14, p = 0, f = 5$ and $s = 1$, the resulting images are of size $10 * 10$. The total number of feature maps produced as output is $9600(96 * 10 * 10)$ located on 96 images.

Large NNs normally have multiple weights and bias parameters. This makes their training difficult as this usually results in dataset over-fitting. Researchers proposed two mechanisms to reduce the possibility of over-fitting. The two approaches are regularization and Drop_outs. The Drop_out layer randomly selects and destroys (turn them to zero) a given fraction of the pixel or voxel data. In this way, the network is taught to learn redundant representations in data and hence avoid over-fitting. The fourth layer is, therefore, a Drop_out layer with a Drop_out rate of 0.5. In this layer, the input images are of size $10 * 10$. For these images, 50% of the pixel values are destroyed. In this way the number of feature maps, images and parameters does not change from those of the previous layer. The only changes that would occur will occur to half the pixel values.

The fifth layer is a MaxPooling layer and the inputs to this layer are images of size $10 * 10$. This layer maintains the number of images from the previous layer but image sizes are halved. Using equation 3.2.2 with $n = 10, p = 0, f = 2$ and $s = 2$ results in outputs of size 5. Also, feature maps are reduced to $2400(5 * 5 * 96)$ and the number of images remained as 96. After the MaxPooling layer, the sixth layer is a convolutional layer. This layer has 700 filters that are of size $2 * 2$ and input images of size $5 * 5$. Additionally, equation 3.2.2 uses the floor() function when presented with fractions. Simplifying $[(5 + 0 - 2)/2] + 1$ results in $\text{floor}(1.5) + 1$, which simplified to $1 + 1 = 2$. This implied that the output images are of size $2 * 2$. This layer produces 700 images of the stated size and $2800(2 * 2 * 700)$ feature maps.

The seventh layer is a Drop_out layer and its inputs are images of size $2 * 2$. The layer outputted 700 images that are of size $2 * 2$ just like the previous layer. Its output pixel values are randomly selected and 50% of the neural connections were rendered ineffective to avoid over-fitting. Additionally, the number of feature maps produced are the same as for the pre-

vious layer. After the Drop_out layer, the eighth layer is a convolutional layer and its input images are of size $2 * 2$. This layer has four filters of size $2 * 2$. Using formula 3.2.2 with $n = 2, p = 0, f = 2, s = 1$ results in 4 output images of size 1.

The final layer is a dense network layer. There are four input images to this layer and each is of size $1 * 1$ (four neurons). Additionally, all the input neurons are connected to the four neurons in this layer. This layer has a Softmax activation function. This activation function is a generalization of logistic regression that enables models to make predictions in multi-class classifications. Equation 3.2.3 [107] gives the general formula for Softmax regression:

$$Z_{\theta}(x^{(j)}) = \begin{bmatrix} P(y^{(j)} = n_1 | x^{(j)}; \beta) \\ P(y^{(j)} = n_2 | x^{(j)}; \beta) \\ \cdot \\ \cdot \\ P(y^{(j)} = n_m | x^{(j)}; \beta) \end{bmatrix} = \frac{1}{\sum_{j=n_1}^{n_m} e^{\beta_j^T x^j}} \mathbf{M} \quad (3.3.6)$$

where $\mathbf{M} = \left[e^{\beta_{n_1}^T x^{(i)}} e^{\beta_{n_2}^T x^{(i)}} \dots e^{\beta_{n_m}^T x^{(i)}} \right]^T$ and $x^1, x^2, x^3, \dots, x^{n_m}$ are the inputs to the Softmax layer while $y^1, y^2, y^3, \dots, y^{n_m}$ are the ground truth values. In the equation, y^j is an element of the set n_1, n_2, \dots, n_m and the quest is to determine the probability that $y^{(i)} = n_k$, where n_k is an element of n_1, n_2, \dots, n_m . Last but not least $\beta_1, \beta_2, \dots, \beta_{n_m}$ are the associated parameters. The Softmax layer produces probabilities of class occurrence given a certain input and the sum of all the probabilities added up to one. The class with the highest probability is the winning class for the classification problem. The final outputs vary, in the experiments with four classes the output is as shown in Table 3.1 but in those with three classes the output would be one less than that in Table 3.1.

3.4 Experimental Design

3.4.1 Conversion From NifTI To Numpy Arrays

The data from CANDI comprises of zipped files with NifTI data files. The present study makes use of NumPy arrays hence all the NifTI data files are converted to NumPy arrays.

NiBabel's `load()` function (`nib.load()`) is used to load the NIfTI data files. When the function loads data files from disk, it loads proxy images. These images have several properties that include a `dataobj` data structure. `Dataobj` data structures are not arrays but array proxies. The proxy objects are then passed as parameters to the function `np.array()` and this returns NumPy arrays. This is done for all the slices in all the data volumes, hence all the data beyond this point is then processed as NumPy arrays.

3.4.2 Train-Validation Splitting

The `train_test_split()` function from Sklearn was designed to split arrays or matrices into random training and validation sets. This function is used in creating the training and validation sets for the present study. The function has several parameters and the ones considered for the present study are `*arrays`, `test_size`, `train_size`, `random_state` and `shuffle`. `train_size` and `test_size` are complementary and their sum is one. In the study only '`test_size = 0.25`' is set and this implies that '`train_size = 0.75`'. It also, means that 75% of the data is for training and the rest for validation. Additionally, the `random_state` parameter can assume either the value `None` (default) or an integer value. If `random_state` assumes its default value, calling the function multiple times would result in different results for each call to the function. When set to an integer value, multiple calls to the function would produce similar results for every call to the function. In the present study, `random` is set to an integer value for consistency and additionally `shuffle` is set to `true`. The `*arrays` parameter take arrays of the same size, in this case, `x_patches` and `y_patches` are the arrays used. These are arrays of patches generated from the raw data and the ground truth respectively. The function then returns two lists containing train-validation splits for each of the sets, raw data and ground truth.

3.4.3 Segmentation Model Voxel Voting

Patch-based segmentation makes use of the neighbourhood information in estimating the value of the central voxel or cell. Several considerations have been made by researchers and these can be grouped into 2D and 3D pixel/voxel voting. Two dimensional (2D) pixel voting makes use of horizontal slices whose size is a major component of the design process. The information contained in the 2D patch is used to estimate its central cell value. These models leave out

information from the neighbourhood cells that are not in the same dimension as the slices. On the other hand, 3D models use the information contained in a volume to vote for the value of the central voxel. These models are known to provide more information during voxel voting. They are therefore expected to produce better results since morphological information is considered in multiple dimensions. 3D models, therefore, process a lot of information hence they are computational resource and time-intensive as opposed to 2D models. In the present study, the proposed model makes use of localized 2D patches in segmenting the human brain.

3.4.4 Data Preprocessing

3.4.4.1 The Chosen Dataset-NITRC-CANDI

Several algorithms have been proposed to segment CANDI brain MRI images. The studies carried out by [13, 33, 51, 52, 63, 75, 98, 100, 115, 116, 117] used the CANDI dataset. The results for the segmentation of the dataset was compared with the results of other datasets such as ADNI, MALC, IBSR, Colins27, OASIS. The results of the comparison showed that the dataset was more difficult to segment as compared to other datasets it was compared with. This can be observed from the studies done by [13, 33, 51, 52, 63, 75, 98, 100, 115, 116, 117] and in results presented in the appendix B1, B2 and B3. We chose the dataset because we identified a research gap and it was easy to access.

3.4.4.2 Image Cropping

Image processing models process large volumes of data, a decision was made to crop the images so that their size does not strain the processing system. The cropping was done to remove as much background pixels as possible while capturing most of the brain tissue pixels to be

classified. The cropping function is given in Algorithm 4 [31].

Algorithm 3: Cropping function Algorithm [31]

Input: Input Image

Output: Cropped Image

```
1 Initialisation: y , x = image.shape
2 Def crop_center(img, cropx, crophy):
3     startx = ((x//2)) - (cropx//2)
4     starty = ((y//2)) - (crophy//2)
5     m = starty + crophy
6     n = startx + cropx
7     return img[starty : m , startx : n ]
```

It takes three parameters, that is, the image, the crop_size along the x-axis (cropx) and crop_size along the y-axis (crophy). The function identified the centre of the image. Additionally, it takes the crop_size along both dimensions and divided them by two. The quotients from the division tasks are then subtracted from the centre coordinates and this gives the starting coordinates of the crop. The crop_size along each dimension is then added to the starting coordinates to give the ending coordinates. The function then returns a cropped image with dimensions as just stated above.

3.4.4.3 Segmentation Classes

The dataset provided by CANDI [86] had 39 classes. Two of these represented un-classified regions of the brain. In the present study, these two regions are classified as background. CANDI website also provided anatomical associations for the values given in the CMA general segmentation file. A copy of this data is provided in Appendix D [84]. The 39 classes are reclassified into four classes. The chosen classes are WM, GM, CSF and background. The background class comprises of all other tissue classes that do not fall into the other three classes. The study carried out in [kagglec29, 80] provided notes that are instrumental in the reclassification. Additionally, most components are made up of both white matter and grey matter, where one was the core and the other formed a layer on top. The dominant material was chosen as the class for the component and ignored the minor material. Table 3.2 shows the four classes and their corresponding tissues.

Table 3.2: Categorization of brain tissues into WM, CSF, GM and background

Category	Tissues
WM	2 , 7 , 41 , 46 , 77 , 85
GM	3 , 8 , 10 , 11 , 12 , 13 , 16 , 17 , 18, 26, 42, 47 , 49 , 50, 51 , 52 , 53, 54 , 58
CSF	4 , 5 , 14 , 15 , 24 , 28 , 43 , 44 , 60 , 72
Background	29 , 30, 36, 61

The components that are categorised as white matter include, Optic chiasm, WM-hypointensities, left and right cerebellum-white-matter, left and right cerebral-white-matter. Also, components that make up the CSF included left and right lateral-ventricle, 3rd, 4th and 5th Ventricles, left and right VentralDC (Ventral Dendritic Cells), left and right Inferior-Lateral-ventricle (Inf-Lat-Vent) and CSF. Additionally, left and right undetermined, left and right vessels and vessels were neither WM, GM nor CSF hence they are categorised as background. Finally, the rest of the components are categorised as Grey Matter, that is, left and right Cerebral-Cortex, left and right cerebellum-cortex, left and right thalamus-proper, left and right Caudate, left and right Putamen, left and right Pallidum, left and right hippocampus left and right Amygdala and Brain-stem.

3.4.4.4 Patch Generation

The function `extract_patches()` is used for patch extraction. This function is part of the Tensorflow's Image class. It takes six parameters, that is, image, sizes, strides, rate, padding and name. The `x_train` dataset is passed as the image parameter in this function. Additionally, the array `[1, 32, 32, 1]` is passed as the patch size. This implies that the patches are of size $32 * 32$ and channel and batch size is set to 1. Also, the array `[1, 1, 1, 1]` is passed as the stride size. This means that the patch extraction function uses step sizes of 1 in either the horizontal or vertical planes. Also, the channel and batch sizes are set to 1. The array passed for rate is the same as the one passed for strides but it means that all the patches generated are to be selected and used. Also, the parameter padding is set to 'SAME'. Lastly, the parameter name is set to 'NONE'. An array is then declared and named `x_patches` and it is assigned the return value of the `extract_patches()` function. The same procedure is done for `y_patches`.

The function `extract_patches()` returns a tensor which has the same data type as its input. The present study makes use of arrays hence Tensorflow's `eval()` function was used in converting tensors to arrays. The function returned NumPy arrays which have the same shape as the input tensors. The extracted patches are then rearranged into a list by appending each patch to a list and then converting the list into a NumPy array. The function `np.array()` is used in converting the list into an array. In the function, `np` stood for NumPy. This conversion is also implemented for the `y_patches`. Consequently, arrays `x_patches` and `y_patches` are $32 * 32$ and the channel size is maintained but the batch size became the number of patches generated. The following images show the first patch's raw and ground truth images. The patches are for the segmentation of the brain into hippocampus, thalamus-proper, cerebellum-cortex, and background.

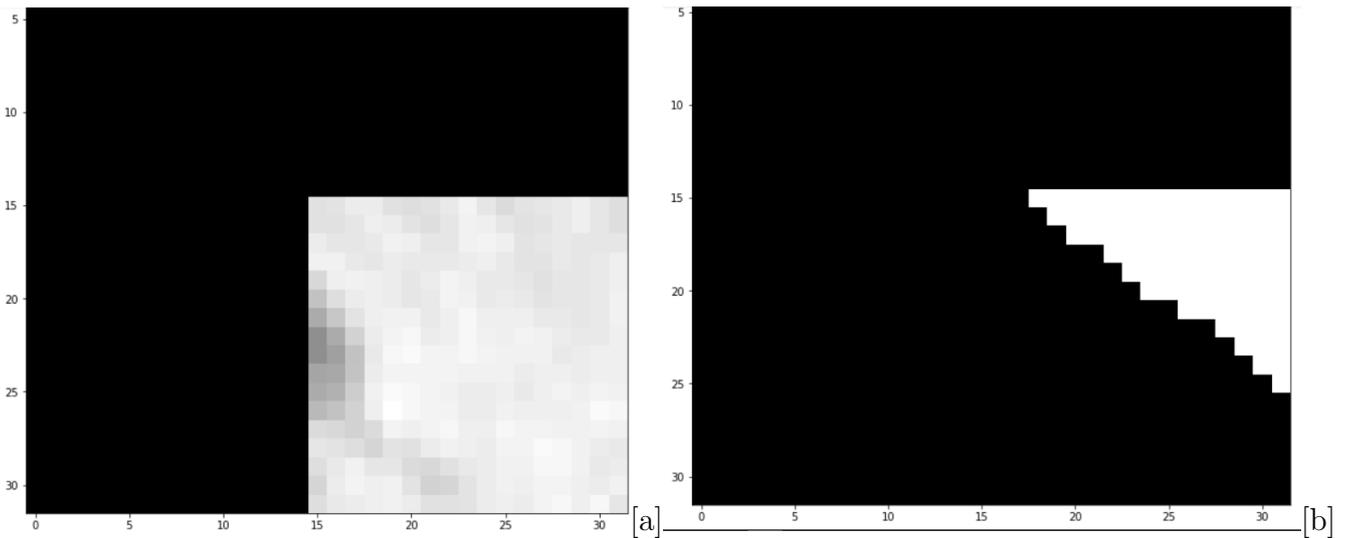


Figure 3.2: Images showing a raw patch [a] and its ground truth patch [b]

Figure 3.2. [a] and [b] shows the raw and ground truth patch images, respectively. Inspecting the images shows that segmenting the image would be difficult for both humans and computer NNs. The challenge is due to the image patch regions and their boundaries are not clear. The NN's task is to learn to segment [a] into [b].

3.4.4.5 Experiments For The Present Study

In the present study, three experiments are carried out on the CANDI provided dataset. The first and last experiment classified three tissues while the second classified two tissues. The

downloaded dataset had gone through several preliminary preprocessing stages before being uploaded for free access to researchers worldwide. In addition to the pre-upload preprocessing, in the first experiment, the 3D brain data is cropped from volumes of size $256 * 256 * a$ to volumes of size $170 * 170 * a$. In the second and third experiments, the images are cropped to volumes of sizes $80 * 80 * a$. In all the cases a is the z -dimension.

During model development and training, all the essential libraries are imported and these included Mayavi, Keras, Matplotlib, NumPy, models, optimizers, regularizers, utils, Skimage, SciPy, Nibabel, Sklearn and TensorFlow. Mayavi provides an interactive visualization of 3D volumes and hence in the initial experiments conducted Mayavi is used in the displaying of datasets, that is, both the ground truth and input data. It is also used for the component by component interactive visualization. Mayavi can also be used for the displaying of the model predicted results and provide an interactive tool for comparison between the expected output and the ground truth.

A number of functions are developed to either aid with preprocessing or as part of the NN and these include functions such as `crop_center()`, `encoder()`, `compute_class_weights()` and `extract_patches()`. The `crop_center()` function is used in the cropping of images, the `encoder()` function forms the NN of the model and the `extract_patches()` function is used to extract patches. The next task is to reclassify brain tissue pixels into WM, GM, CSF, and background.

The next task is the calling of the function that generates weights which are then later passed as a parameter in the `model.fit()` function. The next task is the extraction of patches. These are extracted from both the ground truth and the input data using the `patch_extraction()` function. The patch extraction function returns a tensor, therefore, the next task is the conversion of tensors into arrays. This is done by initially converting the tensor into a list then the list is converted into an array. Additionally, the central values for the ground truth patches are extracted using TensorFlow's `crop_to_bounding_box()` function with height and width set to 1. The results from `crop_to_bounding_box()` are appended into a list, and the list is converted into an array. The new arrays then became the array of the ground truth used during training. The ground truth array is then label encoded using Sklearn's preprocessing tools. Also, a for loop is used to eliminate patches that comprised of only background pixels. The ground truth labels that corresponded to the eliminated patches are also eliminated. The

remaining input patches and the ground truth label arrays are then fed as parameters into Sklearn's `train_test_split()`. This function then split the given data into training and validation sets, with the validation sets being 25% of the total available data. Callbacks are added to aid in the training of the NN. During training, the NN parameters are systematically altered to improve the performance of the NN. The adopted neural architecture is also systematically altered to improve the segmentation performance of the NN. Several performance metrics are used in evaluating the performance of any proposed model and its parameters.

In the first session of experiments, the algorithm proposed by Cui. *et al* [32] is slightly modified to improve its segmentation accuracy. Some of the changes that are made include the removal of the softmax layer and adoption of Softmax as the activation function of the dense layer. The first experiment segments the human brain into background, WM, GM and CSF. In the second set of experiments, the revised architecture is used in the localized patch-based segmentation. There are two major experiments carried out, the first segments two components of the brain while the other segments three components. The experiment with two components segments the brain into background, thalamus-proper and hippocampus. On the other hand, the experiment with three components segments the brain into background, cerebellum-cortex, thalamus-proper and hippocampus. The performance of the NN in each of the experiments is assessed using several evaluation metrics. The following section covers some of the evaluation metrics used.

3.5 The Proposed Algorithm

Algorithm 3 gives a summary of the proposed algorithm. This description of the algorithm has several components that are not given. All conversions from one data structure to the other are left out. The algorithm makes extensive use of arrays hence most data structures produced during the algorithm's execution are converted to arrays. Algorithm 3 also excludes the `call_back()` function and the function responsible for mitigating class imbalances' effects. It also excludes the label encoding function. The neural network (`encode()` function) was also left out, but some of the details are given in this section. The proposed algorithm made extensive use of Python's built-in functions such as `extract_patches()` and `crop_to_bounding_box` amongst others. Details of these functions are not provided in Algorithm 3.

Algorithm 3 shows the algorithm uses $80 * 80$ slices and the patches that are then extracted from these slices. The function also initializes several variables used in the neural network. Some of these variables include `base_size` and `num_classes`. Patches are then extracted using built-in Python functions. The `train_test_split()` function is then used in splitting the patches into training and validation datasets. The model is then trained using the model, and the other parameters are set as shown in Algorithm 3.

Algorithm 4: Proposed Algorithm

Result: Segmented Brain Slices

```
1 ns = number_of_slides
2 images= [ ], images1=[ ]
3 for i = 0; i < ns; i++ do
4     y_train1 = y[:, :, i]
5     x_train1 = x[:, :, i]
6     y_train = crop_center(y_train1, 80, 80)
7     x_train = crop_center(x_train1, 80, 80)
8     images.append(y_train)
9     images.append(x_train)
10 end
11 batch_size, epochs, inchannel, x, y, num_classes = 4, 185, 1, 32, 32, 4
12 for i = 0; i < ns; i++ do
13     y_patches.append(extract_patches1(y_train[i]))
14 end
15 y_patches = tf.image.crop_to_bounding_box(y_patches, 16,16,1,1)
16 y_patches = to_categorical(y_patches)
17 x_patches1 = [ ]
18 for i = 0; i < ns; i++ do
19     x_patches1.append(extract_patches(x_train[i]))
20 end
21 x_patches = x_patches1 x_train, x_valid, y_train, y_valid = train_test_split(x_patches,
    y_patches, test_size=0.25, random_state=0, shuffle = True)
22 sgd= SGD(lr = 0.001, decay = 0.005, momentum = 0.9)
23 model = Model(input_img, encoder(input_img))
24 model.compile(loss="categorical_crossentropy", optimizer=sgd, metrics=['accuracy'])
25 CNN_train = model.fit(x_train, y_train, epochs=epochs, verbose=1,
    validation_data=(x_valid, y_valid), shuffle=True, callbacks = [cp_callback])
```

3.6 Evaluation Metrics

The following section covers the evaluation metrics considered in the present study. It covers concepts such as accuracy, loss, stochastic gradient descent (SGD), learning rate (lr), decay,

momentum, dice similarity coefficient and the confusion metrics.

3.6.1 Accuracy And Loss Metrics

Machine learning and CNNs in particular still suffer from under-fitting or over-fitting. There are several reasons why models would end up over-fitting or under-fitting. To avoid models over-fitting a technique called cross-validation was proposed. In cross-validation, the datasets would be split into training and validation sets. The training set would be used for initializing and fine-tuning the parameters of the model while the validation set would be used for evaluating the predictive accuracy of the trained NN. On the training set, metrics such as training loss and training accuracy would be used to evaluate the training progress. The training loss metric through the loss function would be used by the model to update the model's parameters. On the other hand, training accuracy would be used to show how well the training was proceeding. When training a NN model, the objective would be to minimize the loss function inline with the model parameters through changing values in the weight vector. This would be achieved through the use of an optimization algorithm such as the Stochastic Gradient Descent algorithm. The training or validation loss would be a sum of the accumulated errors encountered during either training or validation. Additionally, even though the objective would be to minimize the loss function the ripple effect of this would be an increase in the training accuracy. When all things are equal, one would expect to see a decrease in the loss function values after every epoch or each iteration of the optimization. Alternatively, one would observe an exponential increase in training accuracy after every epoch.

Validation accuracy is defined as the percentage of correctly classified pixels. It would be evaluated after the model has finished training and the parameters are now fixed. Of the three metrics discussed so far, validation accuracy would be the best measure of the performance of a NN. A high value of training accuracy corresponding to a lower validation accuracy would be an indication of a dataset that would now be starting to over-fit a given dataset. When a model starts over-fitting, it would be best to stop and adjust parameters such as weight decay in regularization or the dropout rate and data augmentation could also prove to be helpful. The proposed algorithm uses the Stochastic gradient descent with the parameters learning rate (lr), decay and momentum set to 0.001, 0.005 and 0.9, respectively. Validation

loss and validation accuracy graphs are plotted to visualise the performance of the proposed algorithm. If the algorithm were to be performing well, the curves for validation loss and training loss would be exponentially decreasing and asymptotic to zero. Alternatively, the curves for training accuracy and validation accuracy would be exponentially increasing and asymptotic to 100%.

3.6.2 Dice Similarity Coefficient (DSC)

The Dice similarity coefficient (DSC) is an evaluation metric that is extensively used in quantifying segmentation overlap. Unfortunately, it is not currently implemented in Keras. There is, therefore, a need to write a DSC function or calculate it from the confusion matrix. Equation 3.5.4 provides the formula used to calculate DSC.

$$DSC = \frac{2|A \cap B|}{|A| + |B|} \quad (3.6.1)$$

In equation 3.5.4, A would be the number of pixels segmented manually while B would be those that would have been segmented by the proposed algorithm [35, 32, 58]. In the present study, the values for DSC are used in assessing the performance of the proposed algorithm against state_of_the_art algorithms in brain MRI segmentation.

3.6.3 Confusion Matrix

The confusion matrix is used as an evaluation metric in the present study. It is a table that describes how well the classification model performs predictions. The confusion matrix was originally designed for binary classification but Sklearn has an implementation for multi-label classification. However, the multi-label implementation is based on the binary confusion matrices for the presented classes. Additionally, in a binary classification, assuming that, a binary classifier would be classifying pixels belonging to one of two classes (positive/negative) then Table 3.3 would provide the confusion matrix.

Table 3.3: Confusion matrix for binary classification

	Positive	Negative
Positive	TP	FP
Negative	FN	TN

In Table 3.3, TN, FP, FN and TP represents ‘True Negative’, ‘False Positive’, ‘False Negative’ and ‘True Positive’, respectively. ‘True Positive’ would represent the number of times the algorithm correctly predict Positive. Additionally, TN would represent the number of times the algorithm correctly predict negative. Also, FP would represent the number of times the algorithm wrongly predict positive. Lastly, FN would represent the number of times the algorithm wrongly predict negative. Additionally, one advantage of the confusion matrix is that it has several evaluation metrics that can be calculated from it. Appendix C provides a list of some of the commonly used evaluation metrics. The Appendix also provides a description of each evaluation metric and a formula.

3.7 Conclusion

The chapter covered the technical breakdown of the NN considered in this research. The model commences with patch images of size $32 * 32$ and the output of the NN are 4 neurons that were governed by a sigmoid function. The evaluation metrics finally used were validation accuracy and loss, Dice Similarity Coefficient (DSC) and confusion metric and other evaluation metrics derived from it. Input images were converted into NumPy arrays for easy processing. The 39 classes in the original dataset were regrouped into four classes, that is, WM, GM, CSF and Background. The input images were cropped first before being processed further. 3D voxel voting consumes a lot of resources during processing even though it includes more morphological information. On the other hand, 2D pixel voting uses fewer resources but also limited morphological information. The present study focused on localized component segmentation since most medical diagnosis would only require certain sections of the brain to be segmented.

4. Results and Discussions

4.1 Introduction

The chapter covers the results of the experiments carried out. It commences with an extension of the study done by Cui *et al.* [32] in segmenting the brain into GM, WM, and CSF, mainly focusing on improving the segmentation accuracy of the proposed algorithm. The first section includes the training and validation loss analysis, followed by training and validation accuracy analysis. The last part of the section includes the confusion matrix and the test statistics calculated from it. Each test statistic is then evaluated and analyzed. The second section summarises the work done on localized component-based segmentation. The following sections similarly cover the evaluation metrics, as was done in the first section. Each section concludes with a section conclusion. The last section covers the comparison of the results of the proposed algorithm against state-of-the-art algorithms. The chapter then closes with a chapter conclusion.

4.2 Proposed Algorithm's Vs Cui's algorithm

There are a lot of similarities between the proposed algorithm and the one proposed by Cui. Some of these include the fact that the model layers are exactly the same, the hyper-parameters learning rate, momentum and decay are set to the same values and the initialization of weights are the same. In addition, both algorithms use patch-based segmentation in their implementation and both eliminate patches which comprise of only background pixels before training of the algorithm. On the other hand the proposed model has no softmax layer instead the activation function for the second from last layer is set to softmax. In addition, in the study by Cui patch generation is done on the whole slice whereas in the present study patch generation is only done on localised regions of interest. Also, the study done by Cui focused on binary classification, we focussed on three and four component segmentation. In their evaluation, the study done by Cui used two evaluation metric and we used several in order to

access the performance of the neural network.

4.3 Brain Segmentation

This section covers the experiments (referred to as the first experiment) done in improving the study done by Cui *et al.*[32]. The present study's main thrust is the improvement of segmentation accuracy and generally segmenting the brain into GM, WM, and CSF. The following sections discuss some of the results.

4.3.1 Training And Validation Losses

Figures 4.1 and 4.2 shows the performance of the NN during training and validation, respectively. The charts are expected to show an exponential decrease in the losses. The decrease would be an indication that the NN's loss function values were decreasing. This decrease would then imply that the NN was learning in each epoch. Also, the loss function values were expected to be asymptotic to zero. In Figure 4.1, the training losses are exponentially decreasing and asymptotic to zero as expected.

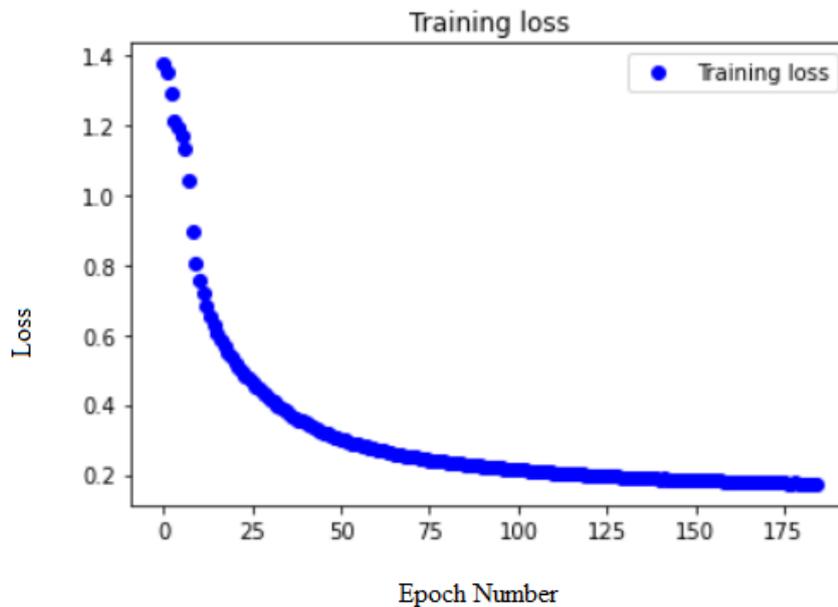


Figure 4.1: First experiment's training Loss graph

The vertical axis shows the loss values for the different epochs, while the horizontal axis shows

the epoch number. In the experiment 185 epochs are used, and the training loss drops from 1.3748 to 0.1734. The graph in Figure 4.2 shows the corresponding validation losses that are recorded as the training progressed.

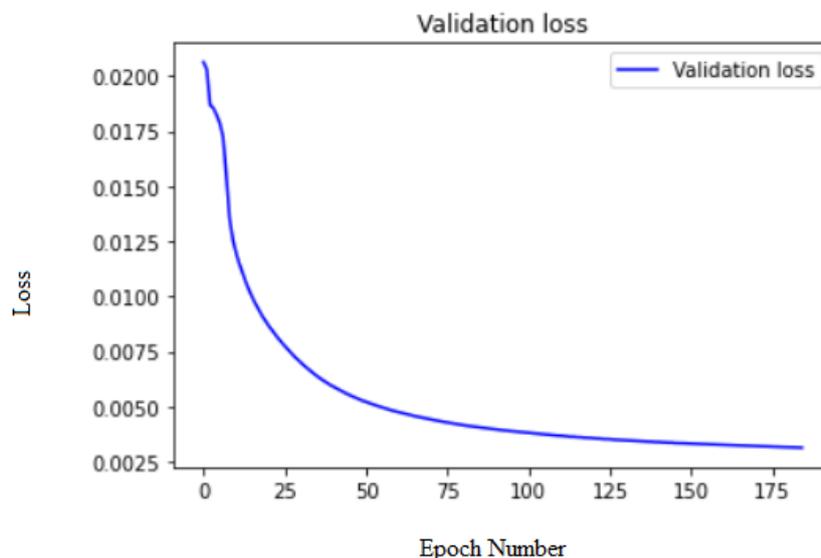


Figure 4.2: First experiment's validation Loss graph

Figure 4.2 shows the validation losses recorded for different epochs while the horizontal axis shows the epoch numbers. The validation losses drop from 0.0206 in the first epoch to 0.0031 in the last recorded epoch. In general, the two graphs indicate that the NN is learning, as demonstrated by the exponential decrease in the losses. Additionally, the two charts are also asymptotic to zero as expected.

4.3.2 Training And Validation Accuracy

Figure 4.3 shows the NN's performance in the actual segmentation of the given patches. Training and validation losses are calculated using a function whose objective function is to be minimized. In contrast, training and validation accuracy tests are based on how well the NN performed at predicting the central values of given patches. The best-expected result would be 100% accuracy, but NNs have been known to provide results as high as 90% or more.

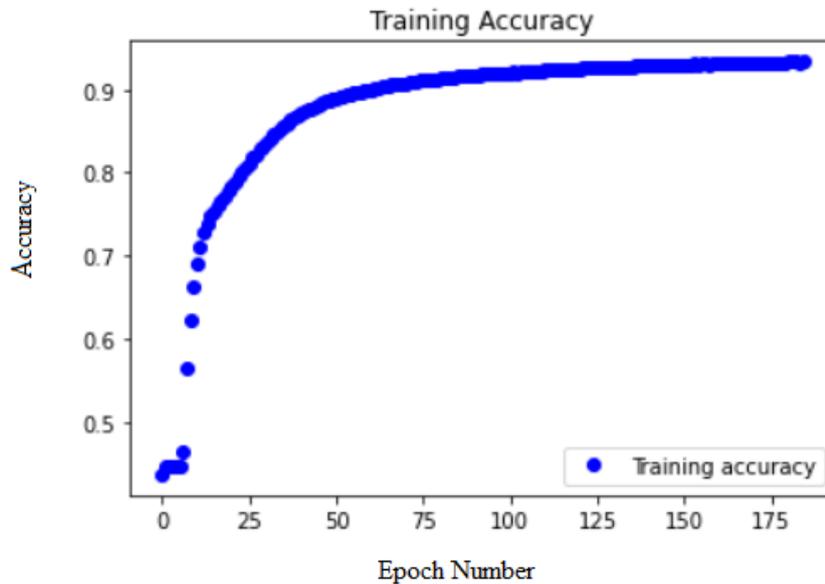


Figure 4.3: First experiment's training accuracy graph

Figure 4.3 shows the graphed results of the training accuracy rates realised during training. The vertical axis shows the training accuracy rates while the horizontal axis shows the epoch numbers. In the graph, the training accuracy rates are exponentially increasing as expected. They increase from 43.71% in the first epoch to 93.37% in the last recorded epoch. Additionally, Figure 4.4, shows the graphed results for validation accuracy rates recorded as training progressed.

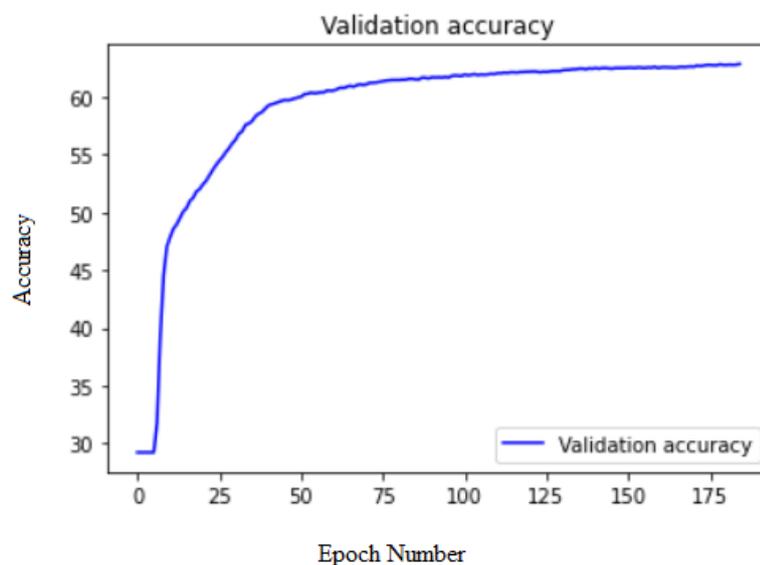


Figure 4.4: First experiment's validation accuracy graph

In Figure 4.4, the vertical axis shows the validation accuracy rates, whereas the horizontal axis shows the epoch numbers. The validation accuracy rates are also exponentially increasing.

They increased from 29.24% in the first epoch to 62.86% in the last recorded epoch. The exponential increase in the accuracy rates indicates that the NN is learning and the recorded training accuracy is more than some of the state_of_the_art NNs.

4.3.3 Confusion Matrix

After training, the NN is used to classify 2481 patches. The confusion matrix is used to test how well the NN performed in the classification task. Table 4.1 shows the confusion matrix with classification results. The top labels show the actual values, while the horizontal ones show the predicted values.

Table 4.1: Confusion matrix for the first experiment

	Actual			
Predicted	Background	GM	CSF	WM
Background	469	6	0	0
GM	7	972	2	75
CSF	0	0	4	5
WM	0	33	0	908

The confusion matrix shown in Table 4.1 shows that the background labels (476 in total) are correctly classified 469 times and never classified as either WM or CSF but are misclassified as GM 7 times. In addition, GM labels (1011 in total) are correctly classified 972 times, misclassified as WM and background 33 and 6 times respectively but never misclassified as CSF. Also, CSF labels (6 in total) are correctly classified 4 times, misclassified as GM 2 times and never classified as either background or WM. WM labels (988 in total) are correctly classified 908 times, misclassified as CSF 5 times, misclassified as GM 75 times and never misclassified as background.

The confusion matrix in Table 4.1 is for multiple labels and hence does not display true negative (TN), true positive (TP), false positive (FP), and false negative (FN) values. The formulas used to calculate the several test statistics derived from the confusion matrix require these values. Table 4.2 shows the tabulated values for the TN, TP, FN, and FP for each of the classes considered in this section of the current research.

Table 4.2: Table showing TP, TN, FP and FN for the first experiment

Brain tissue	TP	TN	FP	FN
Background	469	1999	6	7
GM	972	1419	84	39
CSF	4	2470	5	2
WM	908	1460	33	80

Table 4.2 shows that the TP values for Background, GM, CSF and WM are 469, 972, 4 and 908 respectively. In addition, the TN values for the same classes are 1999, 1419, 2470 and 1460 respectively. Also, the FP values for the same classes are 6, 84, 5 and 33 respectively. Finally the FN values for the the same classes are 7, 39, 2 and 80 respectively.

4.3.3.1 Test Statistics Derived From The Confusion Matrix

The confusion matrix data was used to calculate several test statistics used in evaluating the NN's performance. Table 4.3 gives a summary of these test statistics for the four classes used in this section of the current research. The test statistics used are Accuracy, Recall, Precision, Prevalence, Null Error Rate, False Positive Rate, True Negative Rate, and F1_Score.

Table 4.3: Test statistic values obtained from the confusion matrix (Table 4.2)

Test statistic	Background	GM	CSF	WM	Averages
Accuracy	99.48	96.37	99.72	95.45	97.06
Recall	98.53	96.14	66.67	91.90	84.90
Precision	98.74	92.05	44.44	96.49	77.66
Prevalence	19.19	40.75	0.24	39.82	25.00(26.94)
Null error rate	80.81	60.58	99.76	60.18	75.33
False positive rate	1.26	8.31	83.33	3.34	31.66
True negative rate	98.74	91.69	16.67	96.66	68.34
F1_score	98.63	94.05	53.33	94.13	80.50

Accuracy

The first statistic considered is accuracy, this statistic expresses as a percentage the number of cases the classifier accurately classified. The calculated values for the Background, GM, CSF and WM are 99.48%, 96.37%, 99.72% and 95.45% respectively. These values show that the NN has an accuracy rate of 99.48% in classifying Background tissue pixels, 96.37% in classifying the GM, 99.72% in classifying the CSF and 95.45% in classifying WM. The average accuracy for the classification of GM, CSF and WM is calculated to be 97.06%. The accuracy of the proposed algorithm is relatively high and rival state_of_the_art NNs.

Recall

The next statistic considered is Recall (True Positive Rate), and it measures the rate at which the algorithm predicted positive, given that the correct label value is positive. The Recall values for Background, GM, CSF and WM are 98.53%, 96.14%, 66.67% and 91.90% respectively. These values show that when the actual label is Background, the algorithm predicts Background at a rate of 98.53%. Similarly, in classifying GM, the algorithm correctly predicts GM in 96.14% of the times. Also, in classifying CSF, the model correctly classified CSF, given that the actual label is CSF in 66.67% of the times. Last but not least, the algorithm correctly predicts WM in 91.90% of the cases. The average recall for the three tissue classes, that is, GM, CSF and WM is 84.90%.

Precision

The third statistic considered is Precision, and this test statistic measures how often the algorithm would be correct if it predicts positive. The Precision values for the Background, GM, CSF and WM are 98.74%, 92.05%, 44.44% and 96.49% respectively. In this case, the algorithm correctly predicts positive in 98.74% of the Background cases, 92.05% of the GM cases, 44.44% of the CSF cases, and 96.49% of the WM cases. The average Precision for the three human brain tissues being segmented is calculated to be 77.66%

Prevalence

The fourth statistic considered is Prevalence, and this statistic measures how often the positive cases appear in the sample used. The Prevalence values for the categories Background, GM, CSF and WM are 19.19%, 40.75%, 0.24%, and 39.82%, respectively. This indicated that the sample comprise of 19.19% background tissues, 40.75% GM tissues, 0.24% CSF tissue labels, and 39.82% WM tissue labels. The average Prevalence is 25.00% for the four categories and 26.94% for the three human brain tissue classes.

Null Error Rate

The fifth statistic considered is Null Error Rate, and this measures how often the algorithm would be wrong if it always predicts the majority class. The Null Error Rates for Background, GM, CSF and WM are 80.81%, 60.58%, 99.76% and 60.18% respectively. If the algorithm always predicts the majority class, then it would be wrong 80.81% of the times for Background class labels, 60.58% of the times for the GM cases, 99.76% of the times for the CSF class labels, and 60.18% of the time in the WM class labels. The average Null Error Rate is calculated to be 75.33%. These results indicate that the algorithm has to do more than just estimating the majority class for it to be effective in predicting class labels.

False Positive Rate

The sixth statistic considered is the False Positive Rate (Specificity), and it measures the rate at which the algorithm predicts positive when the correct label value is negative. The Specificity values for the Background, GM, CSF and WM are 1.26%, 8.31%, 83.33% and 3.34% respectively. The Specificity values implied that in 1.26% of the cases, it predicts Background when the correct label is not Background. When the correct label is not GM, the algorithm predicts GM in 8.31% of the cases. When the correct label is not CSF, the algorithm predicts CSF in 83.33% of the cases, and finally, WM has a Specificity of 3.34% which implies that the algorithm predicted WM when the correct response is not WM in 3.34% of the cases. The average Specificity is calculated to be 31.66%, which implies that the algorithm would wrongly predict a positive in 31.66% of the cases.

True Negative Rate

The seventh statistic considered is True Negative Rate, and the statistic measured the rate at which the algorithm predicts negative when the actual label is negative. The True Negative Rates values for Background, GM, CSF and WM were 98.74%, 91.69%, 16.67%, and 96.66%, respectively. For Background the algorithm correctly predicts the negative label in 98.74% of the cases. For GM, the algorithm correctly predicts negative in 91.69% of the cases. For CSF, the algorithm correctly predicts negative labels in 16.67% of the cases. Finally, for WM, the algorithm correctly predicts negative labels in 96.66% of the cases. The average True Negative Rate is calculated to be 68.34%.

Dice Similarity Coefficient

The final test statistic considered is F1_Score (also know as Dice Similarity coefficient), and this gives a weighted average of Recall and Precision. The values obtained for Background, GM, CSF and WM are 98.63%, 94.05%, 53.33%, and 94.13% respectively. From this data, it can be seen that 98.63% of the Background tissues are identical when comparing the actual and predicted labels. Similarly, 94.05% of the labels are similar when comparing GM tissues. In the same way, 53.33% of the tissues are similar when comparing CSF tissues. Lastly, 94.13% of the tissues are similar in the GM classification. The average F1_Score is calculated to be 80.50% for the three brain tissues in the experiment.

4.3.4 Section Conclusion

The loss and accuracy graphs showed the results that were expected of a learning model. Early stopping was used, and training stopped at a training accuracy rate of 93.37%. This result rivals state_of_the_art NN models' training accuracy rates. Additionally, a confusion Matrix was tabulated and then used to calculate several test statistics. The test statistics derived from the confusion matrix showed averagely high performance of the NN, with most values being above 80%. High percentage rates were recorded, where high values were expected except for Precision, which recorded 44.44% for CSF and True Negative rate, which recorded 16.67%. The algorithm also recorded significantly low values in cases where low values were expected

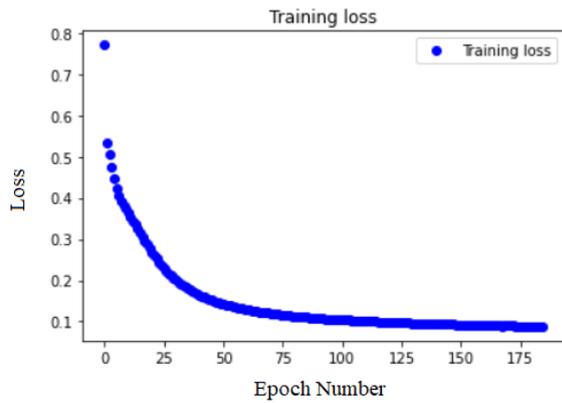
except for CSF, which recorded 83.33% for Specificity for the same tissue label. Test statistics such as Prevalence gave the general tissue label distribution for a given sample. Null Error Rate was low for the majority class but significantly high for all the other tissues indicating that the NN was learning and did not depend on guessing, especially of the majority class.

4.4 Localized Component PBS

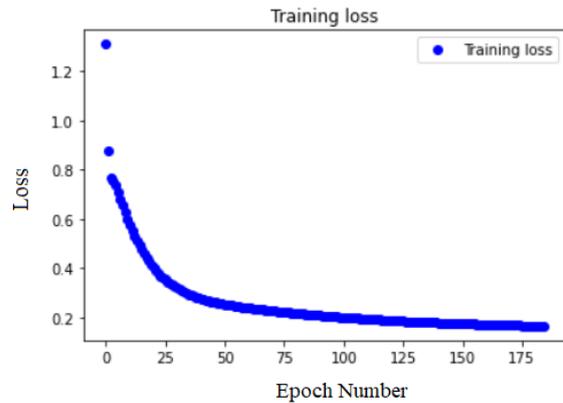
The algorithm used in the first experiment is also implemented in localized patch-based segmentation. This implementation is done in two separate experiments. The first experiment (referred to as the second experiment) segments a localized region of the brain into hippocampus, thalamus-proper, and background. The last experiment (referred to as the third experiment) segments it into background, hippocampus, thalamus-proper, and cerebellum-cortex.

4.4.1 Training Losses

Figure 4.5 shows the performance of the localized component PBS NN model during training. The graphs show the losses that are realized during the two experiments, that is, second (Figure 4.5(a)) and third experiment (Figure 4.5(b)). The charts are expected to show an exponential decrease in the losses. The decrease would indicate that the NN's loss function values are decreasing, which would imply that the NN is learning in each epoch. Also, the loss function values are expected to be asymptotic to zero. In both cases, the losses function values are exponentially decreasing and asymptotic to zero as expected.



(a) Second Experiment



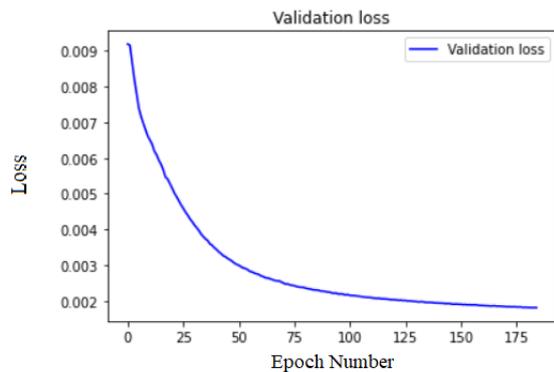
(b) Third Experiment

Figure 4.5: Training losses for the second and third experiment

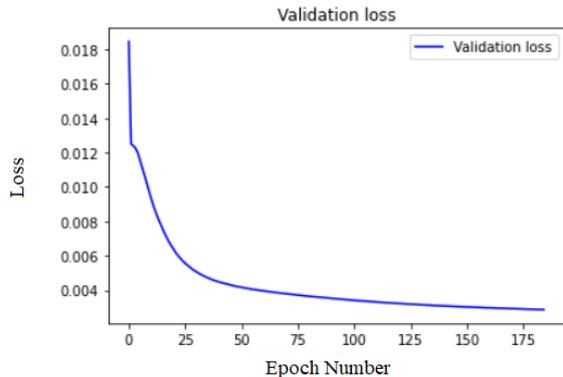
The horizontal axis represents the epoch numbers in both graphs, while the vertical axis represents the loss values at given epoch numbers. In the second experiment loss dropped from 0.7710 to 0.0881. On the other hand, in the third experiment loss dropped from 1.3097 to 0.1647. The two graphs show that the NN is learning during the training process, as evidenced by the exponential decrease in the loss function values.

4.4.2 Validation Losses

Figure 4.6 shows the NN's performance during validation in the two experiments stated above. The graphs show the validation losses incurred during the second (Figure 4.6(a)) and third experiment (Figure 4.6(b)). The charts are expected to show an exponential decrease in the validation losses. The exponential decrease is synonymous to the one for training loss.



(a) Second Experiment



(b) Third Experiment

Figure 4.6: Validation losses for the second and third experiment

The horizontal axis represents the epoch numbers in both graphs, while the vertical axis represents the validation losses at given epochs. Figure 4.6(a) shows that the validation losses dropped from 0.0092 to 0.0018 for the second experiment. On the other hand, Figure 4.6(b) shows that validation losses dropped from 0.0185 to 0.0029 in the third experiment. In both cases, the loss function values are exponentially decreasing and asymptotic to zero. The decrease goes on to affirm the conclusion drawn on training loss analysis (that the NN was learning).

4.4.3 Training Accuracy

Figure 4.7 shows how well the NN is performing the classification task. The graphs show the accuracy rates obtained at different epochs during the second (Figure 4.7(a)) and third experiment (Figure 4.7(b)). The graphs are expected to be exponentially increasing as training of the NN progresses. Also, the accuracy rates are expected to be asymptotic to 100% in each of the two graphs. In both cases, accuracy rates are exponentially increasing and asymptotic to 100% as expected.

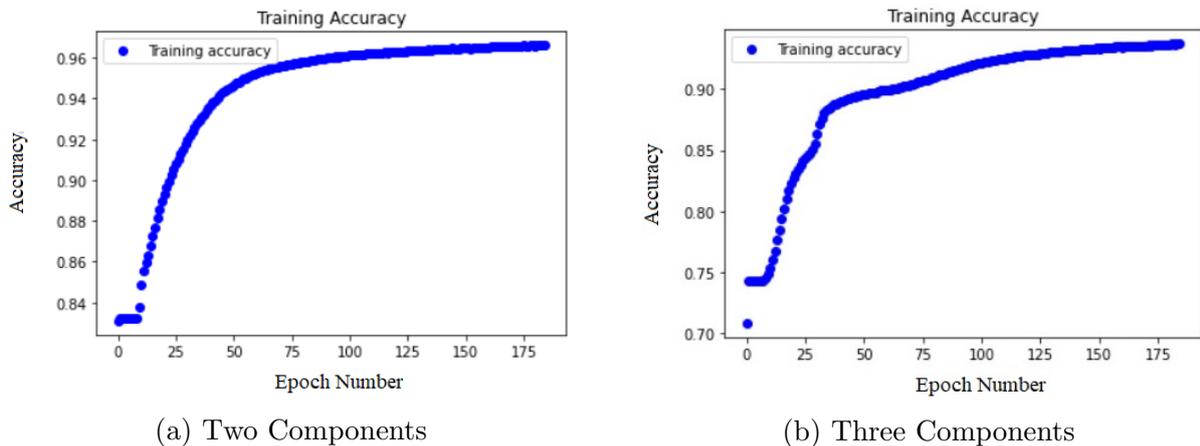


Figure 4.7: Training accuracy for the segmentation of two (a) and three components (b)

The horizontal axis represents the epoch numbers in both graphs, while the vertical axis represents the training accuracy values at given epoch numbers. During training, the accuracy rates in the second experiment increased from an initial value of 83.09% to 96.60% in the last epoch. On the other hand, in the third experiment accuracy increased from an initial value of

70.85% to a value of 93.70% in the last epoch. The two graphs show that the NN was learning during the training process and yielded results that rival state_of_the_art NNs in brain image segmentation.

4.4.4 Validation Accuracy

Figure 4.8 shows how well the NN performed the classification task during the training of the NN in the two experiments. The graphs show validation accuracy rates realised at different epochs during the second (Figure 4.8(a)) and third (Figure 4.8(b)) experiment. The graphs are expected to show exponential increase in validation accuracy and their characteristics are synonymous with training accuracy graphs. In both cases, accuracy rates are increasing and asymptotic to 100% as expected.

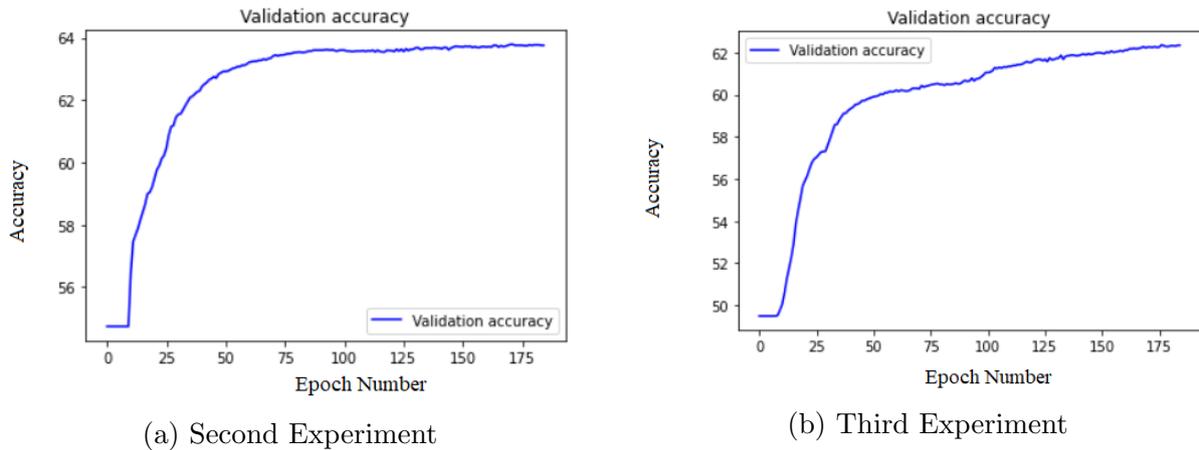


Figure 4.8: Validation accuracy for the second and third experiment

The horizontal axis represents the epoch numbers in both graphs, while the vertical axis represents the validation accuracy rates at given epochs. During training, the rates obtained the second experiment increased from an initial value of 54.74% to a value of 63.75% in the last epoch. On the other hand, in the third experiment, they increased from an initial value of 49.49% to a value of 62.35% in the last epoch. The two graphs show that NN is learning during the training process. Additionally, section 4.3.5 shows some of the segmented images and the corresponding ground truth images.

4.4.5 Segmented Images Of The Hippocampus And Thalamus-Propor

In the second experiment, the model is trained and then used to segment four 2D brain slices. Figure 4.9 shows localized segmented and ground truth slices, respectively.

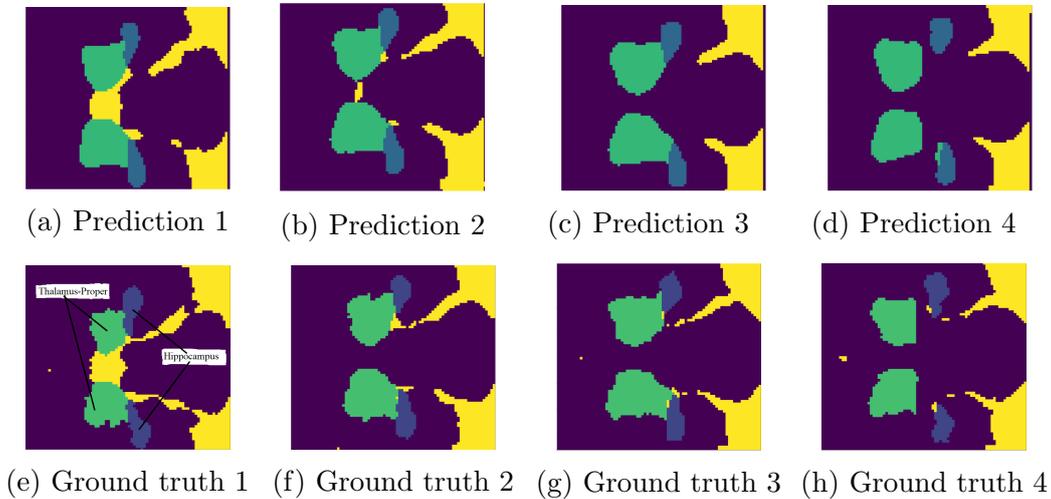


Figure 4.9: Images showing localized segmentation of the hippocampus and thalamus-proper and their corresponding ground truth images

In Figure 4.9, image (a) is the segmented image, while image (e) is its corresponding ground truth image. Additionally, image (b) is the segmented image, while image (f) is its corresponding ground truth image. Image (c) is also the segmented image, while image (g) is the corresponding ground truth image. Finally, image (d) is the segmented image while (h) is its corresponding ground truth image. Additionally, image (e) shows thalamus-proper and hippocampus. In each image, there are two Thalamuses and two hippocampuses. On the other hand, close inspection shows that the algorithm could identify the components but missed some minor details. Also, to quantify the NN's effectiveness in segmenting the human brain, several evaluation metrics are then used. The evaluation metrics are derived from the confusion matrix of each of the two experiments.

4.4.6 Confusion Matrix

After training, the algorithm was used to classify the central pixels for 3 299 patches and 6 326 patches in the second and third experiment, respectively. The results were then tabulated into confusion matrices. These matrices indicate how effective the algorithm had been in each of the classification tasks. Tables 4.4 and 4.5 show the tabulated confusion matrices for the

two experiments. In the tables, vertical labels represent the actual values, while the horizontal ones represent predicted values.

Table 4.4: Confusion matrix for the second experiment

	Actual		
Predicted	Background	Thalamus-proper	Hippocampus
Background	2573	130	15
Thalamus-proper	3	428	0
Hippocampus	9	32	109

Table 4.4 indicates that for all the Background pixels (2 585) classified, 2 573 pixels are correctly classified as Background, whereas three are misclassified as thalamus-proper and nine hippocampus. It also indicates that for all the thalamus-proper pixels (590), 428 pixels are correctly classified as thalamus-proper, while 130 pixels are classified as Background and 32 pixels are classified as hippocampus. Lastly, of all the hippocampus pixels (124), 109 pixels are correctly classified as hippocampus, while 15 are wrongly classified as Background, and none are misclassified as thalamus-proper. Additionally, Table 4.5 shows the confusion matrix for the third experiment. In the table, the vertical and horizontal labels still have the same meaning as in the confusion matrix in the second experiment.

Table 4.5: Confusion matrix for the third experiment

	Actual			
Predicted	Background	Cerebellum-cortex	Thalamus-proper	Hippocampus
Background	4607	73	39	0
Cerebellum-cortex	133	1065	0	13
Thalamus-proper	3	0	186	0
Hippocampus	48	0	17	142

In Table 4.5, of all the background pixels (4791), 4607 pixels are correctly classified as Background, whereas 133 are wrongly classified as cerebellum-cortex, 3 misclassified as thalamus-proper, and 48 misclassified as hippocampus. Of all the cerebellum-cortex pixels (1138), 1065 pixels are correctly classified as cerebellum-cortex, while 73 pixels are misclassified as Background and none as thalamus-proper or hippocampus. Also, of all the thalamus-proper pixels (242), 186 pixels are correctly classified as thalamus-proper, whereas 17 are misclassified as hippocampus, 39 misclassified as Background, and none misclassified as cerebellum-cortex. Finally, of all the hippocampus pixels (155), 142 pixels are correctly classified as hippocampus, whereas 13 are misclassified as cerebellum-cortex, and none are misclassified as either Thallamus-Proper or Background. The two confusion matrices are then used to extract the

data in Table 4.6 and Table 4.7. These tables give the values for TP, TN, FP, and FN for the second and third experiment.

The values in Tables 4.6 are calculated from Table 4.4. Table 4.6 gives the values for TP, TN, FP and FN for the second experiment. In the table, the column headings are TP, TN, FP and FN while the row headings were Background, thalamus-proper and hippocampus.

Table 4.6: Table showing TP, TN, FP and FN for the second experiment

Brain Tissue	TP	TN	FP	FN
Background	2573	567	145	12
Thalamus-proper	428	2706	3	162
Hippocampus	109	2134	31	15

In Table 4.6 the values for TP, TN, FP and FN for the classification of the background are 2573, 567, 145 and 12 respectively. The values for the same labels in the thalamus-proper classification are 428, 2706, 3, 162, respectively. Finally, the values for the same labels in the hippocampus classification are 109, 2134, 31, and 15, respectively. Additionally, Table 4.7 gives a table similar to Table 4.6 but this time for the third experiment. The column and row headings are the same but there is an additional row for the cerebellum-cortex.

Table 4.7: Table showing TP, TN, FP and FN for the third experiment

Brain tissue	TP	TN	FP	FN
Background	4607	1423	112	184
Cerebellum-cortex	1065	5042	146	73
Thalamus-proper	186	6081	3	56
hippocampus	146	6105	86	13

In Table 4.7, the values for TP, TN, FP and FN in the classification of the background are 4607, 1423, 112 and 184 respectively. Additionally, the same labels' values in the cerebellum-cortex classification are 1065, 5042, 146, and 73. Also, the values for the same labels in the classification of the thalamus-proper are 186, 6081, 3, and 56, respectively. Finally, the values

for the same labels in the hippocampus classification are 146, 6105, 86, and 13. The data from Table 4.6 and Table 4.7 are then used to calculate several test statistics used in the evaluation of the performance of the proposed algorithm.

4.4.6.1 Test Statistics Derived From The Confusion Matrix

Values from Table 4.6 are used in the calculation of values in Table 4.8 while values from Table 4.7 are used in the calculation of values in Table 4.9. The two tables include averages for all the test statistics. Most averages are calculated excluding the background class, since it is not a class of interest but just a class that is available during the classification process.

Table 4.8: Test statistic values based on Table 4.6

Test statistic	Background	Thalamus-proper	Hippocampus	Averages
Accuracy	95.18	94.998	98.30	96.16
Recall	99.53	72.54	87.90	86.66
Precision	94.66	99.3	77.86	90.61
Prevalence	78.36	17.88	3.76	33.33
Null error rate	21.58	82.11	95.94	66.54
False positive rate	20.37	0.11	0.99	7.16
True negative Rate	79.63	99.89	99.01	92.84
F1_score	97.03	83.94	82.53	87.83

Accuracy

The first statistic considered is accuracy, this statistic expressed as a percentage the number of cases the classifier accurately classified. In the second experiment, the calculated values for the Background, thalamus-proper, and hippocampus are 95.18%, 94.998%, and 98.30%, respectively (Table 4.8). This shows that the NN had an accuracy rate of 95.18% in classifying Background pixels, 94.998% in classifying the thalamus-proper, and 98.30% in classifying the hippocampus brain tissue pixels. The average accuracy for the classification of the thalamus-proper and hippocampus is calculated to be 96.16%. On the other hand, the accuracy values calculated in the third experiment are 95.32%, 96.54%, 99.07%, and 98.81% for

Background, cerebellum-cortex, thalamus-proper, and hippocampus respectively (Table 4.9). This shows that the algorithm correctly classified background tissues at an average rate of 95.32%, cerebellum-cortex at an average rate of 96.54%, thalamus-proper at an average rate of 99.07%, and hippocampus at an average rate of 98.81%. The average accuracy rate for the three brain components is calculated to be 98.14%. It is noted that the average accuracy is lower in the two tissue segmentation as compared to the three tissue segmentation. The average accuracy for the two experiments is calculated to be 97.15%. Considering the average accuracy and all the other metrics used to this point, it shows that the NN is learning and its results rival some of the state_of_the_art NNs in brain image segmentation.

Table 4.9: Test statistic values based on Table 4.7

Test statistic	Background	C-cortex	T-proper	Hippocampus	Averages
Accuracy	95.32	96.54	99.07	98.81	98.14
Recall	96.16	93.59	76.86	91.82	87.42
Precision	97.63	87.94	98.41	62.93	83.09
Prevalence	75.74	17.99	3.83	2.51	33.33(8.11)
Null error rate	24.26	82.01	96.17	97.87	92.12
False positive rate	2.34	12.83	1.24	54.09	22.72
True negative rate	97.66	87.17	98.76	45.91	77.28
F1_score	96.89	90.68	86.31	74.68	83.89

Recall

The following statistic considered is Recall (True Positive Rate), and it measures the rate at which the algorithm predicts positive, given that the correct label value is positive. It can also be defined as the ability for a model to find relevant pixels in the dataset, hence it is best to use when data is imbalanced. In the second experiment, the recall values for Background, thalamus-proper, and hippocampus are 99.53%, 72.54%, and 87.90%, respectively (Table 4.8). The Recall values show that the algorithm predicts Background when the correct label is Background in 99.53% of the cases. In classifying thalamus-proper, the algorithm correctly predicts thalamus-proper in 72.54% of the cases. In classifying hippocampus, the algorithm correctly classifies hippocampus, given that the actual label is hippocampus in

87.90% of the cases. The average Recall for the classification of thalamus-proper and hippocampus is calculated to be 86.66%. On the other hand, in the third experiment, the Recall values for Background, cerebellum-cortex, thalamus-proper, and hippocampus are 96.16%, 93.59%, 76.86%, and 91.82% respectively (Table 4.9). These values indicated that when the correct label is Background, the algorithm correctly predicts Background at an average rate of 96.16%. Given that the correct label is cerebellum-cortex, the algorithm correctly predicts the cerebellum-cortex at a rate of 93.59%. In the case where the actual label is thalamus-proper, the algorithm correctly predicts it at a rate of 76.86%. Last but not least, when the actual label is hippocampus, the algorithm correctly predicts it at a rate of 91.82%. The average Recall for the three brain tissues is calculated to be 87.42%. The Recall rate is higher in the third experiment as compared to the second experiment. The average Recall rate for the two experiments is calculated to be 87.04%. The lowest recall rate is 76.86%. This shows that for the case where data is limited, the NN network's least performance is 76.86%. There is room for improvement, but this is exceptionally high.

Precision

The third statistic considered is Precision, and this test statistic measures how often the algorithm would be correct if it predicts positive. In the second experiment, the Precision values for the Background, thalamus-proper, and hippocampus are 94.66%, 99.3%, and 77.86%, respectively (Table 4.8). In this case, the algorithm correctly predicts positive in 94.66% of the Background cases, 99.3% of the thalamus-proper cases, and 77.86% of the hippocampus cases. The average Precision for the two brain tissues being segmented is calculated to be 90.61%. On the other hand, in the third experiment the Precision values for Background, cerebellum-cortex, thalamus-proper, and hippocampus are 97.63%, 87.94%, 98.41%, and 62.93% respectively (Table 4.9). The algorithm is right in the classification of background pixels in 97.63% of the cases, in the cerebellum-cortex case, it is correct in 87.94% of the cases, in the thalamus-proper, it is correct 98.41% of the cases, and in the hippocampus, it is correct in 62.93% of the cases. The average Precision is calculated to be 83.09% for the three brain components. Precision is higher in the second experiment as compared to the third experiment. The average Precision for the two experiments is calculated to be 86.85%. This indicates that, on average, whenever the algorithm predicts positive, 86.85% of the time it would be correct. There is room for improving the Precision, but, the NN was performing well.

Prevalence

The fourth statistic considered is Prevalence, and this statistic measures how often the positive cases appear in the sample used. In the second experiment, the Prevalence values for the categories Background, thalamus-proper, and hippocampus are 78.36%, 17.88%, and 3.76%, respectively (Table 4.8). These results indicate that the sample comprises of 78.36% background tissues, 17.88% thalamus-proper tissues, and 3.76% hippocampus tissue labels. The average Prevalence is 33.33% for the four categories and 10.82% for the three brain tissue classes. On the other hand, in the third experiment, the prevalence values for Background, cerebellum-cortex, thalamus-proper, and hippocampus are 75.74%, 17.99%, 3.83%, and 2.51% respectively (Table 4.9). This implies that the labels used comprise of 75.74% background labels, 17.99% cerebellum-cortex labels, 3.83% thalamus-proper labels, and 2.51% hippocampus labels. The four labels' average population is calculated to be 25%, and the average for the three brain tissues is calculated to be 8.11%. The average Prevalence for the brain components in the two experiments is calculated to be 9.47%. The results in this paragraph indicate that the training samples used were highly imbalanced with the Background constituting the bulk of the labels.

Null Error Rate

The fifth statistic considered is Null Error Rate, and this measured how often the algorithm would be wrong if it always predicts the majority class. In the second experiment, the Null Error Rates for Background, thalamus-proper, and hippocampus are 21.58%, 82.11%, and 95.94%, respectively (Table 4.8). These rates imply that, if the algorithm always predicts the majority class, then it would be have been wrong 21.58% of the times for Background class labels, 82.11% of the times for the thalamus-proper cases, and 95.94% of the times for the hippocampus cases. The average Null Error Rate is calculated to be 66.54%. On the other hand, in the segmentation of three components, the Null Error Rate values for Background, cerebellum-cortex, thalamus-proper, and hippocampus are 24.26%, 82.01%, 96.17%, and 97.87% respectively (Table 4.9). These results indicate that if the algorithm is just to predict that every label is the majority class, then it would be wrong 24.26% for the background label, 82.01% for the cerebellum-cortex, 96.17% for the thalamus-proper labels, and 97.87% of the hippocampus labels. The average Null Error Rate for the two experiments is

calculated to be 79.33%. This average error rate indicates that if the algorithm is to always predict the majority class on average, then it would be wrong 79.33% of the time. This shows that the algorithm had to do more than just estimating the majority class for it to be useful in predicting class labels.

False Positive Rate

The sixth statistic considered is the False Positive Rate (Specificity), and it measures the rate at which the algorithm predicted positive when the correct label value is negative. The Specificity values for the Background, thalamus-proper, and hippocampus are 20.37%, 0.11%, 0.99%, and 7.16% respectively, for the second experiment (Table 4.8). The Specificity values imply that in 20.37% of the cases, it predicts Background when the correct label is not Background. The algorithm predicts thalamus-proper in error 0.11% of the time. Also, the algorithm wrongly predicts the hippocampus 0.99% of the time. The average Specificity is calculated to be 7.16%. This implies that the algorithm would wrongly predict a positive instead of a negative in 7.16% of the cases presented. On the other hand, in the third experiment, the Specificity values for Background, cerebellum-cortex, thalamus-proper, and hippocampus are 2.34%, 12.83%, 1.24%, and 54.09% respectively (Table 4.9). This implies that the algorithm wrongly predicts Background at a rate of 2.34%, it wrongly predicts cerebellum-cortex at a rate of 12.83%, it wrongly predicts thalamus-proper at a rate of 1.24%, and wrongly predicts hippocampus at a rate of 54.09%. The average rate of Specificity is calculated to be 22.72%. This shows that in 22.72% of the cases, the algorithm wrongly predicted positive instead of negative. The average Specificity for the two experiments is calculated to be 14.94%. Which implied that in 14.94% of the cases, the algorithm would predict positive instead of negative.

True Negative Rate

The seventh statistic considered is True Negative Rate, and the statistic measures the rate at which the algorithm predicts negative when the actual label is negative. The True Negative Rates for Background, thalamus-proper, and hippocampus are 79.63%, 99.89%, and 99.01%, respectively, for the second experiment (Table 4.8). This implied that for Background, the algorithm correctly predicts the negative label in 79.63% of the cases. For thalamus-proper,

the algorithm correctly predicts negative in 99.89% of the cases. For hippocampus, the algorithm correctly predicts negative labels in 99.01% of the cases. The average True Negative Rate is calculated to be 92.84%. On the other hand, for the third experiment, the True Negative Rates for Background, Cerebellum-Cortex, thalamus-proper, and hippocampus are 97.66%, 87.17%, 98.76%, and 45.91% respectively (Table 4.9). This implies that the algorithm correctly predicts negative, in the background segmentation 97.66% of the cases. It also correctly predicted negative in cerebellum-cortex segmentation at a rate of 87.17% and the same applied for thalamus-proper at a rate of 98.76% and hippocampus segmentation at a rate of 45.91%. The average True Negative rate is calculated to be 77.28%. Additionally, the average True Negative Rate for the two experiments is calculated to be 85.06%. This implies that the algorithm correctly predicted negative at a rate of 85.06%.

Dice Similarity Coefficient

The final test statistic considered is F1_Score (Dice Similarity coefficient), and this gives weighted averages of Recall and Precision. The values obtained for Background, thalamus-proper and hippocampus are 97.03%, 83.94%, and 82.53%, respectively, for the second experiment (Table 4.8). From this data, it is noted that when comparing actual and predicted labels, the algorithm correctly predicts background labels at a rate of 97.03%. Also, 83.94% of the thalamus-proper labels are correct when comparing classification results and actual results, and 82.53% of the tissues are similar when comparing predicted and actual hippocampus tissues. The average F1_Score is calculated to be 87.83% for the two brain tissues being segmented. On the other hand, in the third experiment, the F1_Scores for Background, cerebellum-cortex, thalamus-proper, and hippocampus are calculated to be 96.89%, 90.69%, 86.31%, and 74.68% respectively (Table 4.9). This indicates that in comparing the predicted and actuals values, the background pixels are 96.89% similar, the cerebellum-cortex pixels are 90.68% similar, the thalamus-proper pixels are 86.31% similar, and the hippocampus pixels are 74.68% similar. The average F1_Score is 83.89% for the three brain components being segmented. This implied that, on average, 83.89% of the labels predicted are similar to the actual labels. The average score for the two experiments is calculated to be 86.85%. This implies that when a classification task is carried out, 86.85% of the labels would be similar to the actual labels.

4.4.7 Section Conclusion

In the training and evaluation of the NN, the loss and accuracy graphs showed the expected results of a learning model. The training accuracy for the second and third experiment were 95.26% and 93.70%, respectively. These two give an average segmentation accuracy of 94.48%, which rivals state_of_the_art NNs in brain image segmentation. During training, early stopping was used, but training continued to the last epoch. The algorithm was then used to predict labels, and the results of the prediction were used to tabulate confusion matrices. The confusion matrices were then used to calculate several test statistics used to test how well the NN was performing the classification tasks presented to it. The test statistics derived from the confusion matrix showed high performance of the NN, with most values being above 70% where high percentage rates were expected except for Null_Error_Rate, which had a low value for the majority class. The other statistic that was below 70% was the True_Negative_Rate. Most test statistic values were above 80%, with the majority being above 90% where high values are expected. The algorithm also recorded significantly low values in cases where low values were expected, except hippocampus, which recorded 54.09% for False Positive Rate. Test statistics such as Prevalence gave the general tissue distribution for the given tissues. Null Error Rate was low for the majority classes but significantly high for all the other tissues indicating that the algorithm was learning and did not depend on guessing, especially of the majority class.

4.5 Comparison With Prior Art

A comparison is also made of the test statistics in Tables 4.3, 4.8, and 4.9. It is evident that in 4 out of the 6 test statistics (excluding Prevalence), localized segmentation produced better results. Additionally, two-component segmentation produced poor results in two test statistics compared to whole-brain segmentation, while three-component segmentation produced better results than the whole-brain segmentation. The results from the present study are also compared with results for other state_of_the_art algorithms. The predictive accuracies of these algorithms are acquired from the study done by Cui *et al.* [32]. Table 4.10 then gives the predictive accuracy rates for the second experiment, third experiment, and other state_of_the_art algorithms.

Table 4.10: Accuracy rates for the proposed CNN and state_of_the_art CNNs [32]

Neural Network	Total no. of labels	Average no. of correct labels	Accuracy rates
Proposed CNN(2 components)	3299	3110	94.27%
Proposed CNN(3 components)	6326	6000	94.84%
Cui CNN	24725	22 458	90.83%
CNN1	24 725	22 246	89.97%
CNN2	24 725	22 299	90.18%
CNN3	24 725	21 326	86.25%

Table 4.10 shows that the second and third experiment have accuracy rates of 94.27% and 94.84%, respectively. This is compared against 90.83%, 89.97%, 90.18% and 86.25% for the algorithms Cui CNN, CNN1, CNN2 and CNN3, respectively [32]. In both implementations, the proposed algorithm shows improved accuracy rates as compared to other state_of_the_art algorithms. Additionally, Table 4.11 shows the Dice similarity coefficients for the proposed algorithm compared against other state_of_the_art algorithms acquired from a study done by Zandifar *et al.* [126]. In the table, EC stands for error correction.

Table 4.11: Dice ratios for the proposed CNN and state_of_the_art CNNs [126]

Neural network	Dice ratio
Proposed CNN(2 components)	87.83%
Proposed CNN(3 components)	83.89%
FreeSurfer with EC (error correction)	86.9%
ANIMAL with label fusion	86.2%
ANIMAL with label fusion with EC	86.9%
patch-based	87.9%
patch-based with EC	88.9%
Nonlinear patch-based	88.6%
Nonlinear patch-based with EC	89.4%

Table 4.11 shows that the dice ratios for the second and third experiment are 87.83% and 83.89%, respectively. This is compared against Dice ratios 86.9%, 86.2%, 86.9%, 87.9%, 88.9%, 89.4% and 88.6% for the algorithms FreeSurfer with EC, ANIMAL with label fusion, patch-based, patch-based with EC, Nonlinear patch-based and Nonlinear patch-based with EC, respectively.

ANIMAL with label fusion with EC, patch-based, patch-based with EC, Nonlinear patch-based and Nonlinear patch-based with EC, respectively [126]. Additionally, the algorithms in Table 4.11 segmented the hippocampus only whereas the proposed algorithm segmented the hippocampus together with the thalamus-proper in one experiment and in the other it included the cerebellum-cortex. The proposed algorithm performed better than FreeSurfer and Animal (both versions). It however has almost the same Dice score with the patch-based algorithm. Lastly both versions of non-linear patch-based segmentation produced slightly better results. Also, the Dice scores for the proposed algorithm are compared with those acquired from the study carried out by Haegelen *et al.* [46]. Table 4.12 shows the average Dice scores for hippocampus and thalamus-proper.

Table 4.12: Average dice scores for the proposed CNN and state_of_the_art CNNs in segmenting the hippocampus and thalamus-proper [46]

Neural network	Dice ratio
Proposed algorithm	87.83%
Animal	82.25%
SyN	82.20%
Patch-based	71.75%

Table 4.12 shows that the average dice score for the segmentation of the hippocampus and thalamus-proper is calculated to be 87.83%. This was compared against 82.25%, 82.20% and 71.75% for the algorithms Animal, SyN and Patch-based, respectively [46]. The proposed algorithm out performs the other algorithms it is compared against. There is a difference of more than 5% between the Dice score of the proposed algorithm and the next best algorithm in Table 4.12. The proposed algorithm is also compared with other algorithms that were acquired from a study done by Zhang *et al.* [127]. The study compared the parcellation results of several algorithms in segmenting several components of the brain. Table 4.13 then shows the comparison between the Dice scores from the study against the proposed algorithm.

Table 4.13: Average dice scores for the proposed CNN and state_of_the_art CNNs in segmenting the hippocampus and thalamus-proper [127]

Proposed CNN(2 components)	87.83%
Proposed CNN(3 components)	83.89%
Majority Voting (MV)	53.60%
STAPLE [118]	56.90%
Local Weight Voting (LWV)	68.00%
Patch-Based Method (PBM)	73.60%
Sparse Patch-Based Method (SPBM)	75.60%

Table 4.13 shows that the Dice ratios for the second and third experiments are 87.83% and 83.89%, respectively. This is compared against Dice ratios 53.60%, 56.90%, 68.00%, 73.60% and 75.60% for the algorithms majority voting (MV), STAPLE, local weight voting (LWV), patch-based method (PBM) and sparse patch-based method (SPBM), respectively [127]. The proposed algorithm out performs all other algorithms. The least difference between the proposed algorithm’s Dice score and those of other state_of_the_art algorithms is 8.89%. Additionally, Table 4.14 shows the Dice scores for the proposed algorithm compared against those of state_of_the_art algorithms acquired from the study done by Cui [32].

Table 4.14: Dice ratios for the proposed CNN and state_of_the_art CNNs [32]

Neural Network	Dice Ratio
Proposed CNN (2 components)	87.83%
Proposed CNN (3 components)	83.89%
Cui CNN	95.19%
CNN1	94.12%
CNN2	94.83%
CNN3	92.62%

Table 4.14 shows that the Dice scores for the second and third experiment are 87.83% and 83.89%, respectively. This was compared against the Dice ratios 95.19%, 94.12%, 94.83% and 92.62% for the algorithms Cui CNN, CNN1, CNN2 and CNN3, respectively [32]. The table values show that the algorithm is outperformed by all the other algorithms. The least difference was 4.79% against CNN3 [32]. The low performance of the algorithm is attributed to

the fact that all the other algorithms were conducting binary classification while the proposed algorithm was segmenting at least three components. Additionally, in comparing Tables 4.11, 4.12, and 4.13, it was observed that the more components segmented, the lower the Dice ratio. The proposed algorithm rivals state_of_the_art algorithms, as shown in cases where the same number of components are segmented in which it outperforms other algorithms. Additionally, in cases where the proposed algorithm had more components to segment, the algorithm underperformed by less than 10%. Also, to pave the way for a fair comparison, the first experiment's classification results are compared with those of the third experiment. Localized segmentation shows increased rates in accuracy, Recall, Precision, Null_Error rate, False_Positive rate and F1-score of 1.08%, 2.58%, 5.52%, 16.79%, -8.94 and 3.39% respectively. These increase/decrease rates indicate that localized segmentation improved the predictive power of the algorithm.

4.6 Conclusion

The model proposed by Cui *et al.* was tweaked, and the segmentation results improved from 90.93% to 93.37%. This improvement in accuracy was realized when the brain was being segmented into WM, GM, and CSF. Several test statistics were then used to test the NN's performance, and it was concluded that in general, the NN was performing well as it rivalled state_of_the_art NNs. The NN was also used in localized component-based segmentation in which two experiments were carried out. The first experiment segmented the brain into two components, while the second segmented the brain into three components. The test statistics also showed that the NN did well in the classification task with an average accuracy rate of 94.48%. It was also noted that some of the component pixel populations were very small, resulting in some low results that were realized during the three experiments carried out during the present study. Lastly, it was also observed that the proposed algorithm rivalled state_of_the_art algorithms in segmenting the brain and generally produced better results in cases where the number of components segmented was the same.

5. Conclusion And Recommendations

5.1 Summary

Accurate segmentation of the brain is an important prerequisite for effective diagnosis, treatment planning, and patient monitoring. The use of manual Magnetic Resonance Imaging (MRI) segmentation in treating brain medical conditions is slowly being phased out in favour of fully-automated and semi-automated segmentation algorithms, which are more efficient and objective. Manual segmentation has, however, remained the gold standard for supervised training in image segmentation.

The advent of deep learning ushered in a new era in image segmentation, object detection, and image classification. The convolutional neural network has contributed the most to the success of deep learning models. Also, the availability of increased training data when using Patch Based Segmentation (PBS) has facilitated improved neural network performance. On the other hand, even though deep learning models have achieved successful results, they still suffer from over-segmentation and under-segmentation due to several reasons, including visually unclear object boundaries. This results in deep learning algorithms having less than perfect results. Researchers have been working tirelessly to improve the accuracy of neural networks. Even though there have been significant improvements, there is still room for better results as all proposed algorithms still fall short of 100% accuracy rate.

In the present study, experiments were carried out to improve the performance of neural network models used in previous studies. The revised algorithm was then used for segmenting the brain into three regions of interest: White Matter (WM), Grey Matter (GM), and Cerebrospinal fluid (CSF). Particular emphasis is placed on localized component-based segmentation because both disease diagnosis and treatment planning require localized information, and there is a need to improve the local segmentation results, especially for small components.

In the evaluation of the segmentation results, several metrics indicate the effectiveness of the localized approach. The localized segmentation resulted in the accuracy, recall, precision, null-error, false-positive rate, true-positive and F1-score increasing by 1.08%, 2.52%, 5.43%, 16.79%, -8.94%, 8.94%, 3.39 respectively. Also, the proposed algorithm produced segmentation results that rivalled those of state_of_the_art algorithms in brain image segmentation.

5.1.1 Contributions

- In this section of the present study, a PBS algorithm is proposed based on the work done by Cui *et al.* [32]. In this algorithm, the NN proposed by Cui *et al.* [32] is tweaked to optimize output. The proposed algorithm is then used in segmenting the human brain into four classes, namely background, WM, GM, and CSF. The major challenge is insufficient data for training the algorithm to segment small brain tissues. Insufficient data results in a sub-optimum performance of the proposed algorithm. On the other hand, to mitigate class imbalance, the brain slices are cropped, ensuring that the majority of the brain's pixels are included. Also, patches that are completely made out of background pixels are eliminated. The algorithm also attempts to solve the class imbalance problem using the `class_weight()` function from a Python library called SKLearn. Class weights give weights that are inversely proportional to any given class's prevalence rate during model fitting. All these approaches are implemented to mitigate the effects of class imbalance during model training. The proposed algorithm improves the training accuracy from 90.83% to 93.37% compared with the results of the algorithm it is based on. Several test statistics are then used in evaluating the performance of the NN. All the test statistics used resulted in the conclusion that the NN rival state_of_the_art NNs in brain MRI image segmentation.
- In most cases, medical diagnosis, treatment planning, and treatment monitoring require the segmentation of a section of the brain or just a few components that are important for the neurological disorder being investigated. In this section of the present study, the proposed algorithm was implemented in localized component PBS. The proposed algorithm was implemented in two separate experiments. The first experiment segmented two components, that is, thalamus-proper and hippocampus. The second segmented

three components, that is, cerebellum-cortex, thalamus-proper, and hippocampus. In evaluating the algorithm's performance in the experiments, the same statistics used in evaluating the first algorithm are used. The averages for the two experiments were then used in judging the performance of the NN in component-based PBS. The average training accuracy for the two experimented are observed to be 95.15%, and this rate is higher than the 93.37% obtained during the segmentation in the first section of this study. The next comparison was the performance of the algorithm in segmenting four classes, that is, [WM, GM, CSF, and background] and [background, hippocampus, thalamus-proper, and cerebellum-cortex]. Training accuracy slightly increased from 93.37% to 93.70%. The accuracy, recall, precision, null error rate, false-positive rate, True positive rate and F1-score increased by 1.08%, 2.52%, 5.43%, 16.79%, -8.94%, 8.94%, 3.39% respectively. All the statistics were expected to increase (except for False_Positive rate) to increase the NN's performance. The statistics show a general increase in the NN's predictive ability when used in localized segmentation. In general, all the statistics used showed significantly high-performance rates for the algorithm with a few sub-optimum performance rates on classes with limited training data. The sub-optimum performances happened even though the same strategies addressing class imbalance used in the first implementation were implemented. The results of the algorithm were also compared to those of state_of_the_art algorithms. The algorithm performs better than some and is outperformed by others. In cases where the number of classes is the same, the algorithm seems to outperform other algorithms it was compared against

The proposed NN's implementation in localised PBS was compared against and rivalled several state_of_the_art NN. Localized segmentation improved the segmentation accuracy especially in cases where the comparison was done with NNs that segmented the same dataset. The improved results from the proposed NN were realised even though the other implementations were mainly binary classifications.

5.2 Limitations And Future Work

Deep learning algorithms have produced state_of_the_art accuracy rates in medical image segmentation. One of the significant contributors to this success were CNNs. Even though these algorithms have made headway in medical image segmentation, there is still room for

improvement. The proposed algorithm, just like all the other algorithms before it, is not without limitations. The platform used also had its limitations.

5.2.1 Limitations

Deep learning NNs, including CNNs, require large volumes of data. The need for large volumes of data compounded by the fact that medical image data is significantly large strains the computational resources of a computer. In each of the experiments carried out in the present study, only a few slices from a single patient are used. The lack of variation in slices results in a NN that does not generalize well due to the lack of variability in the samples used during the NN training and validation. Additionally, Google-Colab was chosen due to its processing abilities superior to other platforms, but this limits the training time to 12 hours. It also has user limitations, which makes extensive experimentation with training very difficult. In addition to these limitations, training the NN to segment small organs presents challenges as there is limited training data for the NN.

5.2.2 Future Work

The extensions to the present study could experiment with data augmentation for small organs. It could consider using multi-dimensional pixel voting as an augmentation strategy. Researchers could also experiment with integrating transferred learning of the little organs into medical image segmentation algorithms. Also, training does not stop before the last epoch, even though early stopping has been implemented in the algorithms proposed. For researchers with locally installed GPUs, they could experiment with training the NNs for long periods. Extensions to the study could also consider developing the proposed algorithms' hybrids with well-known state_of_the_art algorithms like U-Net and V-Net. Additionally, researchers could consider implementing the algorithm with error correction (EC). Also, one could implement the algorithm using Non-linear patch generation with or without EC. Last but not least future work could also consider developing an algorithm for automatic cropping and patch_size generation that maximizes the output of the NN.

Appendices

A. HD, MSD And DSC For OASIS, Collin27 And CANDI Datasets

Table A.1: HD, MSD and DSC for OASIS, Collin27 and CANDI datasets

Statistical test	OASIS dataset	Colin27	CANDI
Mean \pm std HD	20.837 \pm 3.124	7.115	26.561 \pm 2.623
median HD	20.475	3.913	26.739
Mean \pm std MSD	1.870 \pm 0.283	1.802	2.435 \pm 0.666
Median MSD	1.605	0.299	2.250
Mean \pm std DSC	0.716 \pm 0.017	0.593	0.594 \pm 0.045
Median DSC	0.716	0.695	0.601

B. ANOVA Analysis For OASIS, Collin27 And CANDI Datasets

Table B.1: ANOVA table for OASIS, Collin27 and CANDI datasets for HD

	Df	Sum Sq	Mean Sq	F value	Pr>F
Datasets	2	2996	01498.0	5.327	0.00867
Residuals	42	11810	281.2		

Table B.2: ANOVA table for OASIS, Collin27 and CANDI datasets for MSD

	Df	Sum Sq	Mean Sq	F value	Pr>F
Datasets	2	5.46	2.728	0.415	0.663
Residuals	42	276.22	6.577		

Table B.3: ANOVA table for OASIS, Collin27 and CANDI datasets for DSC

	Df	Sum Sq	Mean Sq	F value	Pr>F
Datasets	2	0.148	0.07373	9.538	0.000385
Residuals	42	0.3247	0.00773		

C. Commonly Used Evaluation Metrics Calculated From The Confusion Matrix

Table C.1: Commonly used evaluation metrics calculated from the confusion matrix

Metric	Description	Formula
Accuracy	The percentage of cases the classifier is actually correct	$\frac{TP + TN}{total} \times 100$ (C.0.1)
Misclassification	The percentage of cases the classifier is wrongly classifying	$1 - accuracy$ (C.0.2) OR $\frac{FP + FN}{total} * 100$ (C.0.3)
True Positive rate (Recall)	When the correct value is positive, how often does the model predict positive	$\frac{TP}{ActualPositive}$ (C.0.4)
False Positive Rate	When it is actually negative, how often does the model predict positive	$\frac{FP}{ActualPositive}$ (C.0.5)
Continued on next page		

Table C.1 – continued from previous page

Metric	Description	Formula
True Negative Rate	When the correct result is negative how often does the model predict negative	$1 - FalsePositiveRate$ (C.0.6) OR $\frac{TN}{ActualNegative}$ (C.0.7)
Precision	When it predicts positive, How often is it correct	$\frac{TP}{TP + FP}$ (C.0.8)
Prevalence	How often does the positive cases actually appear in our sample	$\frac{actualpositive}{total}$ (C.0.9)
Null Error Rate	This measures how often the model would be wrong if it always predicted the majority class	$\frac{TN + FP}{total}$ (C.0.10)
Cohen's Kappa	This metric measures how well the model performs as opposed to performance based on chance	$\frac{TP - P_c}{1 - P_c}$ (C.0.11)]
F1-score	This gives the balance between precision and recall	$\frac{2 * (precision * recall)}{precision + recall}$ (C.0.12)
ROC curve	This graph summaries the performance of the classifier over several thresholds. It shows true positive rate on the y-axis and	-

Continued on next page

Table C.1 – continued from previous page

Metric	Description	Formula
	false positive rate on the x-axis	

Where P_c refers to percentage by chance [70].

D. Brain Tissue Classes As Provided In The CANDI Website [84]

Table D.1: Brain tissue classes as provided in the CANDI website

	Number	Component
1	2	Left-Cerebral-white-matter
2	3	Left-Cerebral-Cortex
3	4	Left-Lateral-Ventricle
4	5	Left-Inf-Lat-Vent
5	7	Left-Cerebellum-White-Matter
6	8	Left-cerebellum-cortex
7	10	Left-thalamus-proper
8	11	Left-Caudate
9	12	Left-Putamen
10	13	Left-Pallidum
11	14	3rd-Ventricle
12	15	4th-Ventricle
13	16	Brain-Stem
14	17	Left-hippocampus
15	18	Left-Amygdala
16	24	CSF
17	26	Left-Accumbens-area
18	28	Left-VentralDC
19	29	Left-undetermined
20	30	Left-vessel
21	41	Right-Cerebral-White-Matter
22	42	Right-Cerebral-Cortex
Continued on next page		

Table D.1 – continued from previous page

	Number	Component
23	43	Right-Lateral-Ventricle
24	44	Right-Inf-Lat-Vent
25	46	Right-Cerebellum-White-Matter
26	47	Right-cerebellum-cortex
27	49	Right-thalamus-proper
28	50	Right-Caudate
29	51	Right-Putamen
30	52	Right-Pallidum
31	53	Right-hippocampus
32	54	Right-Amygdala
33	58	Right-Accumbens-area
34	60	Right-VentralDC
35	61	Right-undetermined
36	62	Right-vessel
37	72	5th-Ventricle
38	77	WM-hypointensities
39	85	Optic-Chiasm

References

- [1] *2D Convolution Algorithms*. Accessed on: September, 2020. [Online], Available on. URL: <https://janth.home.xs4all.nl/Publications/html/conv2d.html>.
- [2] Sang-il Ahn, Toan Duc Bui, Hyekyoung Hwang, and Jitae Shin. “Performance of Ensemble Methods with 2D Pre-trained Deep Learning Networks for 3D MRI Brain Segmentation”. In: *International Journal of Information and Electronics Engineering* 9.2 (2019).
- [3] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. “Understanding of a convolutional neural network”. In: *2017 International Conference on Engineering and Technology (ICET)*. IEEE. 2017, pp. 1–6.
- [4] *An image of a typical CNN architecture*. Accessed on: September, 2020. [Online], Available on. URL: https://www.google.com/search?sxsrf=ALeKk01qB2zfS16-_XGp4Tv9AC1c5xvbxA:1600030535037&source=univ&tbm=isch&q=images+of+convolutional+neural+networks&client=ubuntu&sa=X&ved=2ahUKEwj0r66dgufr-AhXCiFwKHVXZAsEQsAR6BAgLEAE&biw=960&bih=663.
- [5] Ananya Anand and Namrata Anand. “Fast Brain Volumetric Segmentation from T1 MRI Scans”. In: *Science and Information Conference*. Springer. 2019, pp. 402–415.
- [6] *Artificial Intelligence-Optimizing Max Pooling Algorithm*. Accessed on: September, 2020. [Online], Available on. URL: https://ai.stackex_change.com/questions/7294/optimizing-max-pooling-algorithm.
- [7] Spyridon (Spyros) Bakas. *Multimodal Brain Tumor Segmentation Challenge 2018*. Accessed on: September, 2020. [Online], Available on. URL: <https://www.med.upenn.edu/sbia/brats2018/registration.html>.
- [8] Dr Samir Kumar Bandhyopadhyay and Tuhin Utsab Paul. “Segmentation of brain MRI image—a review”. In: *International Journal of Advanced Research in Computer Science and Software Engineering* 2.3 (2012).

- [9] Siqi Bao and Albert CS Chung. “Feature sensitive label fusion with random walker for atlas-based image segmentation”. In: *IEEE Transactions on Image Processing* 26.6 (2017), pp. 2797–2810.
- [10] Siqi Bao and Albert CS Chung. “Label inference encoded with local and global patch priors”. In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3374–3378.
- [11] Abol Basher, Kyu Yeong Choi, Jang Jae Lee, Bumshik Lee, Byeong C Kim, Kun Ho Lee, and Ho Yub Jung. “Hippocampus Localization Using a Two-Stage Ensemble Hough Convolutional Neural Network”. In: *IEEE Access* 7 (2019), pp. 73436–73447.
- [12] Malav Bateriwala and Pierrick Bourgeat. “Enforcing temporal consistency in Deep Learning segmentation of brain MR images”. In: *arXiv preprint arXiv:1906.07160* (2019).
- [13] Tariq Bdair, Nassir Navab, and Shadi Albarqouni. “ROAM: Random Layer Mixup for Semi-Supervised Learning in Medical Imaging”. In: *arXiv preprint arXiv:2003.09439* (2020).
- [14] Jose Bernal, Kaisar Kushibar, Mariano Cabezas, Sergi Valverde, Arnau Oliver, and Xavier Lladó. “Quantitative analysis of patch-based fully convolutional neural networks for tissue segmentation on brain magnetic resonance imaging”. In: *IEEE Access* 7 (2019), pp. 89986–90002.
- [15] BIDS. *Brain Imaging Data Structure (BIDS)/ A simple and intuitive way to organize and describe your neuroimaging and behavioral data*. Accessed on: September, 2020. [Online], Available on: URL: <https://bids.neuroimaging.io/>.
- [16] Bigcommerce. *What is a .CSV file and what does it mean for my ecommerce business?* Accessed on: March, 2021. [Online], Available on. URL: <https://www.bigcommerce.com/ecommerce-answers/what-csv-file-and-what-does-it-mean-my-ecommerce-business/#\:\~\:\text=A\%20CSV\%20is\%20a\%20comma,Microsoft\%20Excel\%20or\%20Google\%20Spreadsheets.1>.
- [17] Léonie Borne, Jean-François Mangin, and Denis Rivière. *CNN-based segmentation with a semi-supervised approach for automatic cortical sulci recognition*. 2018. Accessed on: September, 2020. [Online], Available on. URL: https://scholar.google.com/scholar?hl=en&as_sdt=0\%2C5&q=CNN-based+segmentation+with+a+semi-supervised+approach+for+automatic+cortical+sulci+recognition&btnG=.

- [18] Alexander de Brebisson and Giovanni Montana. “Deep neural networks for anatomical brain segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2015, pp. 20–28.
- [19] Jason Brownlee. *Understand the Impact of Learning Rate on Neural Network Performance*. Accessed on: September, 2020. [Online], Available on. URL: <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/>.
- [20] Toan Duc Bui, Jitae Shin, and Taesup Moon. “3D densely convolutional networks for volumetric segmentation”. In: *arXiv preprint arXiv:1709.03199* (2017).
- [21] Diedre Carmo, Bruna Silva, Clarissa Yasuda, Letícia Rittner, and Roberto Lotufo. “Hippocampus Segmentation on Epilepsy and Alzheimer’s Disease Studies with Multiple Convolutional Neural Networks”. In: *arXiv preprint arXiv:2001.-05058* (2020).
- [22] Peter A Cerutti. “Effects of ionizing radiation on mammalian cells”. In: *The Science of Nature* 61.2 (1974), pp. 51–59.
- [23] Li Chen, Mahmud Mossa-Basha, Niranjana Balu, Gador Canton, Jie Sun, Kristi Pimentel, Thomas S Hatsukami, Jenq-Neng Hwang, and Chun Yuan. “Development of a quantitative intracranial vascular features extraction tool on 3 D MRA using semiautomated open-curve active contour vessel tracing”. In: *Magnetic resonance in medicine* 79.6 (2018), pp. 3229–3238.
- [24] Li Chen, Yanjun Xie, Jie Sun, Niranjana Balu, Mahmud Mossa-Basha, Kristi Pimentel, Thomas S Hatsukami, Jenq-Neng Hwang, and Chun Yuan. “Y-net: 3D intracranial artery segmentation using a convolutional autoencoder”. In: *arXiv preprint arXiv:1712.07194* (2017).
- [25] Jun Cheng. *figshare/brain tumor dataset*. 2017. Accessed on: September, 2020. [Online], Available on. URL: https://figshare.com/articles/brain_tumor_dataset/1512427.
- [26] Chiun-Li Chin, Bing-Jhang Lin, Guei-Ru Wu, Tzu-Chieh Weng, Cheng-Shiun Yang, Rui-Cih Su, and Yu-Jen Pan. “An automated early ischemic stroke detection system using CNN deep learning algorithm”. In: *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*. IEEE. 2017, pp. 368–372.
- [27] Francois Chollet. *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG, 2018.

- [28] Mayfield Clinic. *Anatomy of the Brain*. Accessed on September, 2020. [Online], Available on: URL: <https://mayfieldclinic.com/pe-anatbrain.htm>.
- [29] Andrew Collette. *Datasets*. 2014. Accessed on: March, 2021. [Online], Available on: URL: <https://docs.h5py.org/en/stable/high/dataset.html>.
- [30] Pierrick Coupé, Boris Mansencal, Michaël Clément, Rémi Giraud, Baudouin Denis de Senneville, Vinh-Thong Ta, Vincent Lepetit, and José V Manjon. “AssemblyNet: A large ensemble of CNNs for 3D Whole Brain MRI Segmentation”. In: *arXiv preprint arXiv:1911.09098* (2019).
- [31] *crop center portion of a numpy image*. Accessed on: September, 2020. [Online], Available on: URL: <https://stackoverflow.com/questions/39382412/crop-center-portion-of-a-numpy-image>.
- [32] Zhipeng Cui, Jie Yang, and Yu Qiao. “Brain MRI segmentation with patch-based CNN approach”. In: *2016 35th Chinese Control Conference (CCC)*. IEEE, 2016, pp. 7026–7031.
- [33] Mahsa Dadar and D Louis Collins. “BISON: Brain tISue segmentatiON pipeline using T1-weighted magnetic resonance images and a random forests classifier”. In: *bioRxiv* (2019), p. 747998.
- [34] Ivana Despotović, Bart Goossens, and Wilfried Philips. “MRI segmentation of the human brain: challenges, methods, and applications”. In: *Computational and mathematical methods in medicine 2015* (2015).
- [35] Lee R Dice. “Measures of the amount of ecologic association between species”. In: *Ecology* 26.3 (1945), pp. 297–302.
- [36] DW Dickson, P Davies, C Bevona, KH Van Hoesven, SM Factor, E Grober, MK Aronson, and HA Crystal. “Hippocampal sclerosis: a common pathological feature of dementia in very old (≥ 80 years of age) humans”. In: *Acta neuropathologica* 88.3 (1994), pp. 212–221.
- [37] Lukas Folle, Sulaiman Vesal, Nishant Ravikumar, and Andreas Maier. “Dilated deeply supervised networks for hippocampus segmentation in MRI”. In: *Bildverarbeitung für die Medizin 2019*. Springer, 2019, pp. 68–73.
- [38] Stig E Forshult. *Magnetic Resonance Imaging—MRI—An Overview*. Fakulteten för teknik- och naturvetenskap, 2007.

- [39] Wikimedia Foundation. *Tab-separated values*. 2020. Accessed on: March, 2021. [Online], Available on. URL: [https://en.wikipedia.org/wiki/Tab-separated_values#:~:text=A%20tab%2Dseparated%20values%20\(TSV,of%20exchanging%20information%20between%20databases.&text=Each%20field%20value%20of%20a,next%20by%20a%20tab%20character..](https://en.wikipedia.org/wiki/Tab-separated_values#:~:text=A%20tab%2Dseparated%20values%20(TSV,of%20exchanging%20information%20between%20databases.&text=Each%20field%20value%20of%20a,next%20by%20a%20tab%20character..)
- [40] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. “Synthetic data augmentation using GAN for improved liver lesion classification”. In: *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE. 2018, pp. 289–293.
- [41] Benjamín Garzón, Rouslan Sitnikov, Lars Bäckman, and Grégoria Kalpouzos. “Automated segmentation of midbrain structures with high iron content”. In: *Neuroimage* 170 (2018), pp. 199–209.
- [42] Evgin Goceri. “Challenges and Recent Solutions for Image Segmentation in the Era of Deep Learning”. In: *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE. 2019, pp. 1–6.
- [43] Ram Deepak Gottapu and Cihan H Dagli. “DenseNet for anatomical brain segmentation”. In: *Procedia Computer Science* 140 (2018), pp. 179–185.
- [44] Vijay PB Grover, Joshua M Tognarelli, Mary ME Crossey, I Jane Cox, Simon D Taylor-Robinson, and Mark JW McPhail. “Magnetic resonance imaging: principles and techniques: lessons for clinicians”. In: *Journal of clinical and experimental hepatology* 5.3 (2015), pp. 246–255.
- [45] R Guerrero, C Qin, O Oktay, C Bowles, L Chen, R Joules, R Wolz, M del C Valdés-Hernández, DA Dickie, J Wardlaw, et al. “White matter hyperintensity and stroke lesion segmentation and differentiation using convolutional neural networks”. In: *NeuroImage: Clinical* 17 (2018), pp. 918–934.
- [46] Claire Haegelen, Pierrick Coupé, Vladimir Fonov, Nicolas Guizard, Pierre Jannin, Xavier Morandi, and D Louis Collins. “Automated segmentation of basal ganglia and deep brain structures in MRI of Parkinson’s disease”. In: *International journal of computer assisted radiology and surgery* 8.1 (2013), pp. 99–110.
- [47] Jinjin Hai, Jian Chen, Kai Qiao, Lei Zeng, Jingbo Xu, and Bin Yan. “Fast medical image segmentation based on patch sharing”. In: *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*. IEEE. 2017, pp. 336–340.

- [48] Zhou He, Siqi Bao, and Albert Chung. “3D deep affine-invariant shape learning for brain MR image segmentation”. In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 56–64.
- [49] R. E. Hendrick. *Breast MRI-Fundamentals and Technical Aspects*. Springer Science+Business Media, LLC, 2008.
- [50] Moore e Henry Knipe annd Candace Makeda. *NIfTI (file format)*. Accessed on: March, 2021. [Online], Available on. URL: <https://radiopaedia.org/articles/nifti-file-format#:~:text=NIfTI%20is%20a%20type%20of%20file%20format%20for%20neuroimaging.&text=The%20NIfTI%2D2%20format%20is,neuroscience%20and%20even%20neuroradiology%20research..>
- [51] Yuankai Huo, Zhoubing Xu, Yunxi Xiong, Katherine Aboud, Prasanna Parvathaneni, Shunxing Bao, Camilo Bermudez, Susan M Resnick, Laurie E Cutting, and Bennett A Landman. “3D whole brain segmentation using spatially localized atlas network tiles”. In: *NeuroImage* 194 (2019), pp. 105–119.
- [52] Yuankai Huo, Zhoubing Xu, Katherine Aboud, Prasanna Parvathaneni, Shunxing Bao, Camilo Bermudez, Susan M Resnick, Laurie E Cutting, and Bennett A Landman. “Spatially localized atlas network tiles enables 3D whole brain segmentation from limited data”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 698–705.
- [53] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. “Densenet: Implementing efficient convnet descriptor pyramids”. In: *arXiv preprint arXiv:1404.1869* (2014).
- [54] Alzheimer’s Disease Neuroimaging Initiative. *Alzheimer’s Disease Neuroimaging Initiative (ADNI)*. 2003. Accessed on: September, 2020. [Online], Available on. URL: <https://ida.loni.usc.edu/login.jsp?project\\=ADNI>.
- [55] iStore. *.DICOMFile Extension*. 2021. Accessed on: March, 2021. [Online], Available on. URL: <https://fileinfo.com/extension/dicom#:~:text=A%20DICOM%20file%20is%20an,as%20an%20ultrasound%20or%20MRI.&text=NOTE%3A%20DICOM%20images%20more%20commonly,DCM%20extension.1>.
- [56] Rohini Paul Joseph, C Senthil Singh, and M Manikandan. “Brain tumor MRI image segmentation and detection in image processing”. In: *International Journal of Research in Engineering and Technology* 3.1 (2014), pp. 1–5.

- [57] Bach Ngoc Kim, Christian Desrosiers, Jose Dolz, and Pierre-Marc Jodoin. “Privacy-Net: An Adversarial Approach For Identity-obfuscated Segmentation”. In: *arXiv preprint arXiv:1909.04087* (2019).
- [58] Stefan Klein, Ua Van Der Heide, Bas W Raaymakers, Alexis NTJ Kotte, Marius Starling, and Josien PW Pluim. “Segmentation of the prostate in MR images by atlas matching”. In: *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE. 2007, pp. 1300–1303.
- [59] Ruslan Klymentiev. *Kaggle/bronze medal Brain Tumor Detection v1.0 || CNN, VGG-16*. 2019. Accessed on: September, 2020. [Online], Available on. URL: <https://www.kaggle.com/ruslankl/brain-tumor-detection-v1-0-cnn-vgg-16>.
- [60] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. “Siamese neural networks for one-shot image recognition”. In: *ICML deep learning workshop*. Vol. 2. Lille. 2015.
- [61] Arun Krishnan and Jonathan Stoeckel. *CAD (computer-aided decision) support for medical imaging using machine learning to adapt CAD process with knowledge collected during routine use of CAD system*. US Patent 7,529,394. 2009.
- [62] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. “Object recognition with gradient-based learning”. In: *Shape, contour and grouping in computer vision*. Springer, 1999, pp. 319–345.
- [63] Yuemeng li, Hongming Li, and Yong Fan. “ACEnet: Anatomical Context-Encoding Network for Neuroanatomy Segmentation”. In: *arXiv preprint arXIV:2002.05773* (2020).
- [64] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. “A survey on deep learning in medical image analysis”. In: *Medical image analysis* 42 (2017), pp. 60–88.
- [65] LORIS. *LORIS/FEATURES/DATA SHARING*. Accessed on: September, 2020. [Online], Available on. URL: <https://loris.ca/features.html#data-dissemination>.
- [66] Zhen Ma, João Manuel RS Tavares, and RM Natal Jorge. “A review on the current segmentation algorithms for medical images”. In: *Proceedings of the 1st International Conference on Imaging Theory and Applications (IMAGAPP)*. 2009.

- [67] José V Manjón, Jose E Romero, Roberto Vivo-Hernando, Gregorio Rubio-Navarro, De la Iglesia-Vaya, Fernando Aparici-Robles, Pierrick Coupé, et al. “Deep ICE: A Deep learning approach for MRI Intracranial Cavity Extraction”. In: *arXiv preprint arXiv:2001.05720* (2020).
- [68] Daniel S Marcus, Anthony F Fotenos, John G Csernansky, John C Morris, and Randy L Buckner. “Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults”. In: *Journal of cognitive neuroscience* 22.12 (2010), pp. 2677–2684.
- [69] Daniel S Marcus, Tracy H Wang, Jamie Parker, John G Csernansky, John C Morris, and Randy L Buckner. “Open Access Series of Imaging Studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults”. In: *Journal of cognitive neuroscience* 19.9 (2007), pp. 1498–1507.
- [70] Kevin Markham. *Simple guide to confusion matrix terminology*. 2014. Accessed on: September, 2020. [Online], Available on. URL: <https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/>.
- [71] Mathworks. *Know your brain: Cerebral cortex*. Accessed on: March, 2021. [Online], Available on. URL: <https://www.mathworks.com/help/matlab/apiref/matfileapi.html>.
- [72] Mayavit. *Mayavi 4.7.1 documentation/Example gallery/Volume slicer example*. Accessed on: September, 2020. [Online], Available on. URL: https://docs.enthought.com/mayavi/mayavi/auto/example_volume_slicer.html.
- [73] Arnaldo Mayer and Hayit Greenspan. “An adaptive mean-shift framework for MRI brain segmentation”. In: *IEEE transactions on medical imaging* 28.8 (2009), pp. 1238–1250.
- [74] Raghav Mehta and Jayanthi Sivaswamy. “M-net: A convolutional neural network for deep brain structure segmentation”. In: *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. IEEE. 2017, pp. 437–440.
- [75] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. “The multimodal brain tumor image segmentation benchmark (BRATS)”. In: *IEEE transactions on medical imaging* 34.10 (2014), pp. 1993–2024.

- [76] Microsoft. *Zip and unzip files*. Accessed on: March, 2021. [Online], Available on. URL: [https://support.microsoft.com/en-us/windows/zip-and-unzip-files-8d28fa72-f2f9-712f-67df-f80cf89fd4e5/#:~:text=Zipped%20\(compressed\)%20files%20take%20up,with%20uncompressed%20files%20and%20folders.1](https://support.microsoft.com/en-us/windows/zip-and-unzip-files-8d28fa72-f2f9-712f-67df-f80cf89fd4e5/#:~:text=Zipped%20(compressed)%20files%20take%20up,with%20uncompressed%20files%20and%20folders.1).
- [77] Fausto Milletari, Seyed-Ahmad Ahmadi, Christine Kroll, Annika Plate, Verena Rozanski, Juliana Maiostre, Johannes Levin, Olaf Dietrich, Birgit Ertl-Wagner, Kai Bötzel, et al. “Hough-CNN: deep learning for segmentation of deep brain regions in MRI and ultrasound”. In: *Computer Vision and Image Understanding* 164 (2017), pp. 92–102.
- [78] Pim Moeskops, Max A Viergever, Adriënne M Mendrik, Linda S De Vries, Manon JNL Benders, and Ivana Išgum. “Automatic segmentation of MR brain images with a convolutional neural network”. In: *IEEE transactions on medical imaging* 35.5 (2016), pp. 1252–1261.
- [79] Erhard T Næss-Schmidt, Anna Tietze, Irene K Mikkelsen, Mikkel Petersen, Jakob U Blicher, Pierrick Coupé, José V Manjón, and Simon F Eskildsen. “Patch-based segmentation from MP2RAGE images: comparison to conventional techniques”. In: *International Workshop on Patch-based Techniques in Medical Imaging*. Springer. 2015, pp. 180–187.
- [80] Thomas P. Naidich, Johnny C. Ng, John D. Waselus, Bradley N. Delman, and Cheuk Ying Tang. *Deep Gray Nuclei and Related Fiber Tracts*. Accessed on: September, 2020. [Online], Available on. URL: <https://radiologykey.com/deep-gray-nuclei-and-related-fiber-tracts/>.
- [81] Gilad Naor. *Read .mat files in Python*. 2009. Accessed on: September, 2020. [Online], Available on. URL: [https://stackoverflow.com/questions/874461/read-\\$mat-\\$files-\\$in-\\$python](https://stackoverflow.com/questions/874461/read-$mat-$files-$in-$python).
- [82] NBDC. *NBDC Human Database/ NBDC Research ID: hum0031.v1*. 2019. Accessed on: September, 2020. [Online], Available on. URL: <https://humandbs.biosciencedbc.jp/en/hum0031-v1>.
- [83] NITRC. *NITRC*. 2007. Accessed on: September, 2020. [Online], Available on. URL: https://www.nitrc.org/frs/shownotes.php?release_id=1607.

- [84] NITRC. *NITRC: NeuroImaging Tools & resources Collaboratory/Documents*. Accessed on: September, 2020. [Online], Available on. URL: https://www.nitrc.org/docman/\?group_id=377.
- [85] NITRC. *NITRC/About the NITRC Team*. 2007. Accessed on: September, 2020. [Online], Available on. URL: <https://www.nitrc.org/include/team.php>.
- [86] NITRC. *NITRC:NeuroImaging Tools and Resources Collaboratory*. Accessed on: September, 2020. [Online], Available on. URL: https://www.nitrc.org/frs/?group_id=48.
- [87] NITRC. *NITRC/Open-discusion*. 2007. Accessed on: September, 2020. [Online], Available on. URL: https://www.nitrc.org/forum/forum.php?thread_id=3729&forum_id=1515.
- [88] NITRC. *NITRC/Open-discusion*. 2007. Accessed on: September, 2020. [Online], Available on. URL: https://www.nitrc.org/forum/forum.php?thread_id=3654&forum_id=1515.
- [89] NITRC. *NITRC/What is NITRC?* 2007. Accessed on: September, 2020. [Online], Available on. URL: https://www.nitrc.org/include/about_us.php.
- [90] OASIS. *OASIS: Open Access Series of Imaging Studies*. 2019. Accessed on: September, 2020. [Online], Available on. URL: <https://www.oasis-brains.org/>.
- [91] OASIS. *OASIS open Access Series of Imaging Studies*. 2007. Accessed on: September, 2020. [Online], Available on. URL: <http://www.oasis-brains.org/>.
- [92] Keiron O’Shea and Ryan Nash. “An introduction to convolutional neural networks”. In: *arXiv preprint arXiv:1511.08458* (2015).
- [93] *Parkinsons Disease*. 2018. Accessed on: September, 2020. [Online], Available on. URL: https://www.wikiwand.com/en/Substantia_nigra.
- [94] Magdalini Paschali, Stefano Gasperini, Abhijit Guha Roy, Michael Y-S Fang, and Nassir Navab. “3DQ: Compact Quantized Neural Networks for Volumetric Whole Brain Segmentation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 438–446.
- [95] Prabhu Ramachandran and Gaël Varoquaux. “Mayavi: 3D visualization of scientific data”. In: *Computing in Science & Engineering* 13.2 (2011), pp. 40–51.

- [96] Reshare. *UK Data Services/Reshare/A neuroimaging dataset of brain tumour patients*. 2016. Accessed on: September, 2020. [Online], Available on. URL: <http://reshare.ukdataservice.ac.uk/851861/>.
- [97] Reviversoft. *.OTL File Extension*. 2021. Accessed on: March, 2021. [Online], Available on. URL: <https://www.reviversoft.com/file-extensions/otl/#:~:text=Files%20with%20the%20.,its%20excellent%20options%20for%20editing.&text=otl%20file%20extension%20are%20also,and%20Z%20Soft%20Type%20Foundry.l>.
- [98] Anne-Marie Rickmann, Abhijit Guha Roy, Ignacio Saarasua, and Christian Wachinger. “Recalibrating 3D ConvNets with Project & Excite”. In: *IEEE Transaction on Medical Imaging* (2020).
- [99] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.
- [100] Abhijit Guha Roy, Sailesh Conjeti, Nassir Navad, and Christian Waxhinger. “Quick-NAT: segmenting MRI neuroanatomy in 20 seconds”. In: *arXiv preprint arXiv:1801.04161* (2018).
- [101] David W Shattuck, Mubeena Mirza, Vitria Adisetiyo, Cornelius Hojatkashani, Georges Salamon, Katherine L Narr, Russell A Poldrack, Robert M Bilder, and Arthur W Toga. “Construction of a 3D probabilistic atlas of human cortical structures”. In: *Neuroimage* 39.3 (2008), pp. 1064–1080.
- [102] Dinggang Shen, Guorong Wu, and Heung-Il Suk. “Deep learning in medical image analysis”. In: *Annual review of biomedical engineering* 19 (2017), pp. 221–248.
- [103] Hoo-Chang Shin, Neil A Tenenholtz, Jameson K Rogers, Christopher G Schwarz, Matthew L Senjem, Jeffrey L Gunter, Katherine P Andriole, and Mark Michalski. “Medical image synthesis for data augmentation and anonymization using generative adversarial networks”. In: *International workshop on simulation and synthesis in medical imaging*. Springer. 2018, pp. 1–11.
- [104] SIMON. *SIMON Dataset/MEDICS*. 2003. Accessed on: September, 2020. [Online], Available on. URL: http://fcon_1000.projects.nitrc.org/indi/retro/SIMON.html.

- [105] itk SNAP. *itk-SNAP/ ITK-SNAP Image Data Download*. 2018. Accessed on: September, 2020. [Online], Available on: URL: <http://www.itksnap.org/pmwiki/pmwiki.php?n=Downloads.Data>.
- [106] Larry W Swanson. *Brain architecture: understanding the basic plan*. Oxford University Press, 2012.
- [107] Siqin Tao, Tao Zhang, Jun Yang, Xueqian Wang, and Weining Lu. “Bearing fault diagnosis method based on stacked autoencoder and softmax regression”. In: *2015 34th Chinese Control Conference (CCC)*. IEEE. 2015, pp. 6331–6335.
- [108] *The Canadian Dementia Imaging Protocol: Harmonizing National Cohorts*. Accessed on: September, 2020. [Online], Available on. URL: <https://onlinelibrary.wiley.com/doi/full/10.1002/jmri.26197>.
- [109] *The earth rotating about its axis*. Accessed on: September, 2020. [Online], Available on. URL: https://www.google.com/search?rlz=1C1GCEU_enZA912ZA912&sxsrf=ALeKk022eBqmjqcYuGBvyiFEZZGNxX7tog:159963712-1656&source=univ&tbm=isch&q=earth+rotating+about+its+axis+images&sa=X&ved=2ahUKEwjLnp_TyNvrAhXc-URUIHeT2B0IQsAR6BAGMEAE&biw=880&bih=875.
- [110] *The MRI scanner*. Accessed on: September, 2020. [Online], Available on. URL: https://www.google.com/search?q=mri+machine+labelled+diagram\&tbm=isch\&ved=2ahUKEwiT0JTQxNvrAhU8AmMBHWSkCu0Q2-cCegQ-IABAA\&oq=mri+machine+labelled+diagram\&gs_lcp=CgNpbWcQAzICCAA6BAGjECc6B-ggAEAUQHjoECAAQHjo-GCAAQCBAe0gQIABAYUKbMEFjWqxFgoa4RaABwAHgAgAHGAogBmi2S-AQYyLTE4LjSYAQCg-AQGqAQtd3Mtd216LWltZ8ABAQ\&scclient=img\&ei=SYJYX90HG7yEj-LsP5Miq6A4\&bih=920\&biw=1920\&rlz=1C1GCEU_enZA912ZA912.
- [111] Tran Anh Tuan, Jin Young Kim, João Manuel RS Tavares, et al. “White Matter, Gray Matter and Cerebrospinal Fluid Segmentation from Brain 3D MRI Using B-UNET”. In: *ECCOMAS Thematic Conference on Computational Vision and Medical Image Processing*. Springer. 2019, pp. 188–195.
- [112] Juan P Viguera-Guillén, Busra Sari, Stanley F Goes, Hans G Lemij, Jeroen van Rooij, Koenraad A Vermeer, and Lucas J van Vliet. “Fully convolutional architecture vs sliding-window CNN for corneal endothelium cell segmentation”. In: *BMC Biomedical Engineering* 1.1 (2019), pp. 1–16.

- [113] Jianzhong Wang, Jun Kong, Yinghua Lu, Jingdan Zhang, and Baoxue Zhang. “Segmentation of 3d MRI brain images using information propagation”. In: *International Workshop on Medical Imaging and Virtual Reality*. Springer. 2006, pp. 348–355.
- [114] Li Wang, Dong Nie, Guannan Li, Élodie Puybureau, Jose Dolz, Qian Zhang, Fan Wang, Jing Xia, Zhengwang Wu, Jia-Wei Chen, et al. “Benchmark on automatic six-month-old infant brain segmentation algorithms: the iSeg-2017 challenge”. In: *IEEE transactions on medical imaging* 38.9 (2019), pp. 2219–2230.
- [115] Liansheng Wang, Cong Xie, and Nianyin Zeng. “RP-Net: a 3D convolutional neural network for brain segmentation from magnetic resonance imaging”. In: *IEEE Access* 7 (2019), pp. 39670–39679.
- [116] Monan Wang and Pengcheng Li. “Labelfusion method combining pixel greyscale probability for brain MR segmentation”. In: *Scientific Reports* 9.1 (), pp. 1–10.
- [117] Shuxin Wang, Shilei Cao, Dong Wei, Renzhen Wang, Deyu Wang kai Ma andMeng, Yefeng Zheng, et al. “LT-Net; Label Transfer by Learning Reversible Voxel-wise Correspondence for One-shot Medical Image Segmentation”. In: *arXiv preprint arXiv:2003.07072* (2020).
- [118] Simon K Warfield, Kelly H Zou, and William M Wells. “Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation”. In: *IEEE transactions on medical imaging* 23.7 (2004), pp. 903–921.
- [119] Rob GJ Wijnhoven and PHN de With. “Fast training of object detection using stochastic gradient descent”. In: *2010 20th International Conference on Pattern Recognition*. IEEE. 2010, pp. 424–427.
- [120] Fei Wu, YiFei Li, Jian Zhu, Bin Zhou, Youyong Kong, and Huazhong Shu. “Brain Tissue Segmentation Integrating Multi-level Features”. In: *2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)*. IEEE. 2019, pp. 235–239.
- [121] Jianxin Wu. “Introduction to convolutional neural networks”. In: *National Key Lab for Novel Software Technology. Nanjing University. China* 5 (2017), p. 23.
- [122] Jiong Wu, Yue Zhang, and Xiaoying Tang. “Simultaneous Tissue Classification and Lateral Ventricle Segmentation via a 2D U-net Driven by a 3D Fully Convolutional Neural Network”. In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2019, pp. 5928–5931.

- [123] Zhongliu Xie and Duncan Gillies. “Near real-time hippocampus segmentation using patch-based canonical neural network”. In: *arXiv preprint arXiv:1807.05482* (2018).
- [124] Botian Xu, Yaqiong Chai, Cristina M Galarza, Chau Q Vu, Benita Tamrazi, Bilwaj Gaonkar, Luke Macyszyn, Thomas D Coates, Natasha Lepore, and John C Wood. “Orchestral fully convolutional networks for small lesion segmentation in brain MRI”. In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE. 2018, pp. 889–892.
- [125] Li Yi and Gao Zhijun. “A review of segmentation method for MR image”. In: *2010 International Conference on Image Analysis and Signal Processing*. IEEE. 2010, pp. 351–357.
- [126] Azar Zandifar, Vladimir Fonov, Pierrick Coupé, Jens Pruessner, D Louis Collins, Alzheimer’s Disease Neuroimaging Initiative, et al. “A comparison of accurate automatic hippocampal segmentation methods”. In: *NeuroImage* 155 (2017), pp. 383–393.
- [127] Daoqiang Zhang, Qimiao Guo, Guorong Wu, and Dinggang Shen. “Sparse patch-based label fusion for multi-atlas segmentation”. In: *International Workshop on Multimodal Brain Image Analysis*. Springer. 2012, pp. 94–102.
- [128] Xi Zhang, Yanwei Fu, Andi Zang, Leonid Sigal, and Gady Agam. “Learning classifiers from synthetic data using a multichannel autoencoder”. In: *arXiv preprint arXiv:1503.03163* (2015).
- [129] Chenzhuo Zhu, Song Han, Huizi Mao, and William J Dally. “Trained ternary quantization”. In: *arXiv preprint arXiv:1612.01064* (2016).