CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Studying the Impact of Training Algorithms of Artificial Neural Network for Segmentation Accuracy of Synthetic Brain MRIs

by

Sania Javed

A thesis submitted in partial fulfillment for the degree of Master of Science

in the

Faculty of Computing
Department of Computer Science

2021

This thesis work is wholeheartedly dedicated to my beloved parents, siblings, and my best friends who have been my source of inspiration and gave me strength when I thought of giving up, who continually provide their moral, emotional, and financial support. Special thanks to my supervisor Dr. "M. Masroor Ahmed" whose uncountable confidence enabled me to reach this milestone.

CERTIFICATE OF APPROVAL

Studying the Impact of Training Algorithms of Artificial Neural Network for Segmentation Accuracy of Synthetic Brain MRIs

by

Sania Javed

(MCS191018)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Ayyaz Hussain	QAU, Islamabd
(b)	Internal Examiner	Dr. Azhar Mahmood	CUST, Islamabad
(c)	Supervisor	Dr. M. Masroor Ahmed	CUST, Islamabad

Dr. M. Masroor Ahmed
Thesis Supervisor
December, 2021

Dr. Nayyer Masood	Dr. M. Abdul Qadir
Head	Dean
Dept. of Computer Science	Faculty of Computing
December 2021	December 2021

Author's Declaration

I, **Sania Javed** hereby state that my MS thesis titled "**Studying the Impact of Training Algorithms of Artificial Neural Network for Segmentation Accuracy of Synthetic Brain MRIs**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my MS Degree.

(Sania Javed)
MCS191018

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled "**Studying the Impact of Training Algorithms of Artificial Neural Network for Segmentation Accuracy of Synthetic Brain MRIs**" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS Degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

(Sania Javed)
MCS191018

Acknowledgement

In the Name of Allah, the Most Merciful, the Most Compassionate all praise be to Allah, the Lord of the worlds; and prayers and peace be upon Mohammad His servant and messenger. First and foremost, I must acknowledge my limitless thanks to Allah, the Ever-Magnificent; the Ever-Thankful, for His help and blessings.I am totally sure that this work would have never become truth, without His guidance. I am grateful to some people, who worked hard with me from the beginning till the completion of the present research particularly my supervisor **Dr. M. Masroor Ahmed**, who has been always generous during all phases of the research. I would like to take this opportunity to say warm thanks to all my beloved friends, who have been so supportive along the way of doing my thesis. Last but not least, I would like to express my wholehearted thanks to my family for their generous support they provided me throughout my entire life and particularly through the process of pursuing the master degree. Because of their unconditional love and prayers, I have the chance to complete this thesis.

(Sania Javed)

Abstract

Image segmentation characterized as the technique of decomposing a synthetic image into multiple constituting regions, which is an essential and demanding feature through an image processing and analysis process. The key purpose of segmentation is to represent an image into its constituents which are more significant and convenient to analyze. There are various methods in modern image processing techniques that are being utilized for carrying out a detailed analysis of these constituent regions. Artificial-Neural-Network (ANN) is the most important technique which is wide extensively used and relied upon for segmentation analysis. For using ANN it needs to be trained. This study entails, a technique for segmenting images that integrates dualistic approaches, ANN a sophisticated supervised-learning method, and un-supervised learning using a K-means clustering strategy. The anticipated cluster centers are achieved by using K means and, based on this labeled data. The proposed technique was implemented on a freely accessible simulated BrainWeb data-set. A wide range of algorithms is employed for training purposes. The primary goal of this research was to train an ANN model with different ANN training algos on the MRI brain dataset and analyze the performance to predict the algorithm that segments more accurately and provide a fast result with the minimum error and better level of accuracy. In this study, ANN models were trained using 12 different training algorithms. Quasi-Newton (BFG), Levenberg-Marquadt (LM), Bayesian-Regularization (BR), Gradient-descent (GD), Scaled-Conjugate-Gradient (SCG), Gradient-Descent with Momentum and Adaptive-Learning Rate (GDX), Gradient-Descent with Momentum Learning Rate Backpropagation (GDM), Conjugate-gradient with Powell-Beale-restarts (CGB), Fletcher-Reeves update Conjugate-Gradient algorithm (CGF), One-Step-Secant (OSS), Polak-Ribiere update Conjugate Gradient (CGP) and Resilient-Back-Propagation (RP) training algorithms were applied to the dataset. The mean, mean squared error (MSE), standard deviation, and dice similarity index will be used to determine the excellence of the segmentation. Based on the obtained outcomes this study expands its recommendations for the optimum alignment of a specific training algorithm. The findings

describe that the ‘Levenberg-Marquadt’ method performs the best for ANN with this dataset. The ‘Bayesian- Regularization ’ algorithm’s effectiveness is also noteworthy, even though not quite as well as the ‘Levenberg-Marquadt’ method. The fundamental purpose of this work is to find an optimal training method for brain MRI segmentation.

Contents

Author's Declaration iv

Plagiarism Undertaking v

Acknowledgement vi

Abstract vii

List of Figures xii

List of Tables xiv

Abbreviations xvi

Symbols xvii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Purpose . . . 2
- 1.3 Problem Statement . . . 2
- 1.4 Research Questions . . . 3
- 1.5 Proposed Solution . . . 3
- 1.6 Significance of the Solution . . . 3
- 1.7 Tools and Techniques . . . 3
- 1.8 Organization of Thesis . . . 4

2 Literature Review and Background Study 5
- 2.1 Basic Definitions . . . 5
  - 2.1.1 Digital Image . . . 5
  - 2.1.2 Image-Segmentation . . . 5
  - 2.1.3 Image-Segmentation using Pixel Data . . . 8
  - 2.1.4 Image Segmentation Based on Features . . . 9
- 2.2 Techniques of Segmentation . . . 10
  - 2.2.1 Manual Segmentation Method . . . 10
  - 2.2.2 Edge Based Methods . . . 11

2.2.3 Sobel Operator . . . 11
2.2.4 Laplacian Operator . . . 12
2.3 Region-Based Segmentation Methods . . . 13
2.3.1 Seeded Region Growing . . . 13
2.3.2 Region Splitting and Merging Method . . . 14
2.4 Thresholding Method . . . 15
2.5 Segmentation Based on Clustering Methods . . . 16
2.6 Training Algorithms of ANN . . . 17
2.7 Related Work for Image Segmentation . . . 18
2.8 Related Work for ANN Training Algorithms . . . 21

3 Fundamentals of ANN and Training Algorithms 25
3.1 Basic Structure and Strength of ANN . . . 25
3.2 Choice of Activation Functions . . . 28
3.3 Supervised and Unsupervised Learning . . . 28
3.4 Feed Forward Network and its Types . . . 29
3.5 Training Algorithms and their Description . . . 30
3.5.1 Bayesian-Regularization-Backpropagation . . . 30
3.5.2 Levenberg-Marquardt . . . 31
3.5.3 BFGS Quasi-Newton-Backpropagation . . . 31
3.5.4 Conjugate-Gradient-Backpropagation with Powell-Beale-Restarts . . . 31
3.5.5 Conjugate-Gradient-Backpropagation with Fletcher-Reeves Updates . . . 32
3.5.6 Conjugate-Gradient-Backpropagation with Polak-Rib iere Updates . . . 32
3.5.7 Batch–Gradient-Descent Training Algorithm . . . 32
3.5.8 Gradient-Descent with Momentum-Learning-Rate-Backpropagation . . . 33
3.5.9 Gradient-Descent with Momentum and Adaptive-Learning-Rate-Backpropagation . . . 33
3.5.10 Resilient-Backpropagation . . . 33
3.5.11 Scaled-Conjugate-Gradient-Backpropagation . . . 34
3.5.12 One-Step-Secant-Backpropagation . . . 34

4 Research Methodology 35
4.1 Proposed Methodology Steps . . . 37
4.2 K-means Clustering and its Pseudocode . . . 37
4.2.1 Pseudocode of K-means [75, 76] . . . 38
4.3 Steps of ANN [77, 78] . . . 38
4.4 Pseudocode of Proposed Methodology . . . 39
4.5 Experimental Setup . . . 40
4.6 Dataset Acquisition . . . 40
4.7 ANN Architecture . . . 40
4.8 Evaluation Metrices . . . 41

4.8.1 DSC 41
4.8.2 MSE 42
4.8.3 STD 43

5 Results and Evaluation **44**
5.1 Segmentation of Images using Levenberg Marquardt 44
5.2 Segmentation of Images using Bayesian Regularization 67
5.3 Segmentation of Images using BFGS Quasi-Newton 69
5.4 Segmentation of Images using Congugate Gradient with Powell/Beale Restarts 72
5.5 Segmentation of Images using Congugate Gradient with Fletcher powell 74
5.6 Segmentation of Images using Congugate Gradient with Polak-Ribiere 75
5.7 Segmentation of Images using Gradient Descent Back Propogation . 77
5.8 Segmentation of Images using Gradient Descent with Momentum . 78
5.9 Segmentation of Images using Gradient Descent with Variable Learning Rate 80
5.10 Segmentation of Images using One Step Secant Back Propagation . 82
5.11 Segmentation of Images using Resilient Back Propagation 84
5.12 Segmentation of Images using Scale Congugate Gradient 85

6 Conclusion and Future Work **90**
6.1 Conclusion 90
6.2 Future Direction 91

Bibliography **92**

List of Figures

2.1 Generic Process Flow of Brain Tumor Detection System[8]. 8
2.2 Brain Tumor Detection [15]. 8
2.3 Region Growing Method [26]. 14
2.4 Region Splitting and Merging [30]. 15

3.1 Basic Structure of ANN [37] 26
3.2 Activation Functions of ANN [52] 27
3.3 Activation Functions of ANN [52] 28

4.1 Context Diagram of Proposed Methodology. 36
4.2 Architecture of ANN. 41

5.1 Segmentation Output 45
5.2 Segmented Image 46
5.3 GT ROI 46
5.4 Segmented ROI 47
5.5 Superimposed Bar 47
5.6 Segmented Image 48
5.7 GT ROI 48
5.8 GT ROI 49
5.9 Superimposed Bars 49
5.10 Segmented ROI 50
5.11 GT ROI 50
5.12 Segmented ROI 53
5.13 Segmented ROI 53
5.14 Performance of Levenberg Marquardt 54
5.15 Probability Density Estimation 54
5.16 Standard Deviation 55
5.17 Mean for Segmented Image 55
5.18 Dice Similarity for Segmented Image 56
5.19 Segmentation Output 56
5.20 Segmentation Output 57
5.21 Segmentation Output 57
5.22 Segmented Image 57
5.23 Dice Score and MSE for Clean Images 58
5.24 Average Dice Score and MSE for Clean Images. 58

5.25 Segmented Output 0% Noise-0% INU . . . 58
5.26 Segmented Output 0% Noise-20% INU . . . 59
5.27 Segmented Output 0% Noise-40% INU . . . 59
5.28 Dice Score and MSE for Case-0 . . . 59
5.29 Average Dice Score and MSE for Case-0 . . . 60
5.30 Probability Data Distribution for Case-0 . . . 61
5.31 Dice Score and MSE for Case-1 . . . 62
5.32 Average Dice Score and MSE for Case-1 . . . 63
5.33 Dice Score and MSE for Case-3 . . . 63
5.34 Average Dice Score and MSE for Case-3 . . . 64
5.35 Dice Score and MSE for Case-5 . . . 65
5.36 Average Dice Score and MSE for Case-5 . . . 65
5.37 Dice Score and MSE for Case-7 . . . 66
5.38 Average Dice Score and MSE for Case-7 . . . 66
5.39 Performance of Bayesian Regularization Algorithm . . . 68
5.40 Performance of BFGS Quasi-Newton Algorithm . . . 70
5.41 Performance Evaluation of Congugate Gradient with Powell-Beale Restarts . . . 72
5.42 Performance of Congugate Gradient with Fletcher powell . . . 74
5.43 Performance of Congugate Gradient with Polak-Ribiere . . . 76
5.44 Performance of Gradient Descent Back Propogation . . . 78
5.45 Performance of Gradient Descent with Momentum . . . 80
5.46 Performance Evaluation of Gradient Descent with Variable Learning Rate . . . 81
5.47 Performance of One Step Secant Back Propagation . . . 82
5.48 Performance of Resilient Back Propagation . . . 84
5.49 Performance of Scale Congugate Gradient . . . 86
5.50 Comparision of All the Training Algorithms . . . 87

List of Tables

2.1 Comparison Table of Image Segmentation. 24
2.2 Comparison Table of Training Algorithms 24

4.1 Experimental Setup 40
4.2 Description of Data Set 40
4.3 Fixed Parameters of ANN 41

5.1 Performance Evaluation 45
5.2 Performance Evaluation 45
5.3 Levenberg Marquardt Performance 48
5.4 Performance Evaluation 51
5.5 Performance Evaluation 51
5.6 Performance Evaluation 51
5.7 Performance Evaluation 51
5.8 Performance Evaluation 52
5.9 Performance Evaluation 52
5.10 Performance Evaluation 52
5.11 Performance Evaluation 55
5.12 Performance Evaluation for Case-0 56
5.13 Performance Evaluation 60
5.14 Performance Evaluation for Case-1 60
5.15 Performance Evaluation for Case-1 61
5.16 Performance Evaluation for Case-3 62
5.17 Performance Evaluation for Case-3 62
5.18 Performance Evaluation for Case-5 64
5.19 Performance Evaluation for Case-5 65
5.20 Performance Evaluation for Case-7 66
5.21 Performance Evaluation for Case-7 67
5.22 Bayesian Regularization Performance 68
5.23 Performance Evaluation 68
5.24 Performance Evaluation 69
5.25 Performance Evaluation 69
5.26 Performance Evaluation 69
5.27 BFGS Performance 70
5.28 Performance Evaluation 70
5.29 Performance Evaluation 71

5.30 Performance Evaluation 71
5.31 Performance Evaluation 71
5.32 Congugate Gradient with PowellBeale Restarts Performance 72
5.33 Performance Evaluation 73
5.34 Performance Evaluation 73
5.35 Performance Evaluation 73
5.36 Performance Evaluation 74
5.37 Congugate Gradient with Fletcher Powell Performance 75
5.38 Performance Evaluation 75
5.39 Performance Evaluation 75
5.40 Congugate Gradient with Polak-Ribiere Performance 76
5.41 Performance Evaluation 76
5.42 Performance Evaluation 77
5.43 Gradient Descent Back Propagation Performance 77
5.44 Performance Evaluation 78
5.45 Performance Evaluation 79
5.46 Gradient Descent with Momentum Performance 79
5.47 Performance Evaluation 79
5.48 Performance Evaluation 79
5.49 Gradient Descent with Variable Learning Rate Performance 81
5.50 Performance Evaluation 81
5.51 Performance Evaluation 82
5.52 One Step Secant Back Propagation Performance 83
5.53 Performance Evaluation 83
5.54 Performance Evaluation 83
5.55 Resilient Back Propagation Performance 84
5.56 Performance Evaluation 85
5.57 Performance Evaluation 85
5.58 Scale Congugate Gradient Performance 86
5.59 Performance Evaluation 86
5.60 Performance Evaluation 87
5.61 Comparison of Training Parameters 87

Abbreviations

ANN	Artificial Neural Network
BR	Bayesian Regularization
CNN	Convolutional Neural Network
CSF	Cerebrospinal Fluid
CT	Computerized Tomography
DSC	Dice Similarity Coefficient
DWT	Dimensionality Wavelet Transform
FCM	Fuzzy C-means
FFNN	Feedforward Neural Network
GD	Gradient Descent
GLCM	Gray level Co-occurrence Matrix
GM	Grey Matter
INU	Intensity Non-Uniformity
LM	Levenberg-Marqardt
MATLAB	MATrix LABoratory
MLPANN	Multilayer Perceptron Artificial Neural Network
MRI	Magnetic Resonance Imaging
MSE	Mean Square Error
OSS	One Step Secant
RP	Resilient-Back-Propogation
SCG	Scale Congugate Gradient
WM	White Matter

Symbols

α	Alpha
β	Beta
δ	Delta
η	Eta
$\forall$	For All
$\geq$	Greater than or Equal to
$\cap$	Intersection
$\leq$	Less than or Equal to
$\neq$	Not Equal to
μ	Mu
∇	Nabla
∂	Partial Derivative
P	Predicate
ϕ	Phi
$\sum$	Summation
θ	Theta
$\cup$	Union

Chapter 1

Introduction

1.1 Motivation

The fields of bio-medical and computer technology have interacted together to enhance medical diagnostics features. Image segmentation is one of them, it automatically extracts the data pattern to comprehend out of a set of observations. Image processing is the technique of analyzing, interpreting, and manipulating a digital image using computer sophisticated algorithms. Image processing's actual strength can be discovered in many fields such as remote sensing, machine robot vision, microscope imaging, vehicle automation, photography and video creation, weather forecasting, optical character recognition, face and fingerprint detection, and pattern recognition[1, 2].

Accurate image segmentation is a key challenge in computer vision. Tailored to the need, Image segmentation seeks to divide an image into several non overlap-ping and consistent areas[3]. Based on the diversity and complexity of images, it is often one of the more demanding challenges in object recognition[4]. Designing robust and fast segmentation algorithms is of broad interest because it is a critical component of many imaging applications. To achieve the specific needs of the particular field various approaches, strategies, and algorithms have been proposed and investigated for image segmentation [5, 6]. Literature provides various ANN

algorithms that are being used to a variety of real-life problems to forecast the future scenario. Moreover, the performance of these training algorithms was evaluated by comparing them with one another [7, 8].
In biomedical applications, the importance of image segmentation offered a significant aspiration to work in this area. From a vast group of medical images, this study mainly focuses on brain MRI for segmentation purposes. The goal of this research is to apply various training algorithms to MRI synthetic data set and their performance will be compared to find out the best algorithm among them. that provides accurate segmentation with a minimum error rate, achieves the best level of accuracy within less time.

1.2 Purpose

The goal of ANN modeling is to minimize the predictive error rates of the record provided to the network after they have been trained. A hinder or delay in the disease diagnosis process might not be tolerated because it puts the patient's life at risk. So, selecting a robust and efficient training method will speed up the entire disease diagnosis procedure. The basic purpose of this study is to train the data using a variety of training methods and then choose the optimum training algorithm to increase network prediction accuracy. Several training algorithms are applied in the ANN modeling for this purpose, and the effect of the different training strategies is investigated, and the most effective image segmentation learning algorithms is proposed.

1.3 Problem Statement

According to best of our knowledge none of the study have been proposed that apply multiple training algorithms of ANN on synthetic brain MRIs. Biomedical imaging data has grown from Kilobytes to Tera-bytes in recent years, the manual segmentation of biomedical images may produce unreliable results which may leads to inconsistent decisions. As a result, it slows down the whole diagnosis process.

So it is necessary to find out an algorithm that is robust time saving and provide accurate results.

1.4 Research Questions

Q1: Which training algorithm provides a better level of accuracy in image segmentation?
Q2: In term of time complexity which algorithm is suitable to train ANN?

1.5 Proposed Solution

To develop an image segmentation system that has the following characteristics: low time consumption, high reliability and accuracy, and low variability and error rate in segmentation outputs. A list of training algorithms will be investigated on synthetic data, training parameters will be kept similar throughout all the processes and the best training algorithm with all the required features will be proposed.

1.6 Significance of the Solution

The proposed method has provided effective results, as it is capable of handling a large data set. The proposed solution has significantly improved accuracy, And an efficient and robust training algorithm has been proposed for the network.

1.7 Tools and Techniques

Following tools and techniques are used during this work:

1. MATLAB R2017a

2. IP Toolbox

1.8 Organization of Thesis

The remainder of the research is formulated as follows:

Chapter 2: A full introduction to image segmentation, different conventional image segmentation algorithms, and relevant work on medical image segmentation are included in this literature overview. It also describes the use of multiple training algorithms in different fields.

Chapter 3: This chapter discusses different types of learning, the principles of ANN with a focus on its power in medical image processing, and a brief discussion of multiple training algorithms.

Chapter 4: This chapter Reveals the proposed methodology, experimental setup, description of the dataset, and performance measures.

Chapter 5: Demonstrates the consequence of the segmentation results and their performance evaluation.

Chapter 6: Provides the conclusion of this thesis and recommendations for the future.

Chapter 2

Literature Review and Background Study

A brief discussion of the fundamentals of the image segmentation process is given in this chapter. Some basic techniques that are used for image segmentation are also discussed in this chapter. Moreover, training algorithms of ANN and related work about image segmentation are also part of this chapter.

2.1 Basic Definitions

2.1.1 Digital Image

A function I can be used to define an image as I(x, y) in two-dimensional space or I(x, y, z) in three-dimensional space, where x = 0, M 1, y = 1, N 1, and z = 0, D − 1 indicate spatial dimensions. I(x, y) and I(x, y, z) return values of pixel intensity [9].

2.1.2 Image-Segmentation

Image segmentation is referred to as the partitioning of an image into a group of semantically meaningful, uniform, and nonoverlapping segments with similar properties from the surrounds such as luminance, dimension, color, or structure. Segmenting images is a crucial first step in image analysis [9]. Brain tissues segmentation using MR scans is supposed to be extremely useful for diagnosing brain abnormalities, tracking diagnosis, and evaluating medication. Numerous automated or semiautomatic approaches have been promoted as a way to reduce user intervention, however, their accuracy still seems to be lower than manual segmentation in the majority of situations [10].

Medical image processing, facial detec-tion, pedestrians identification, and even more applications require image analysis techniques [11]. It's the process by which a computation turns an array of adja-cent pixels into parts containing consistent and homogenous properties from the original image analysis. Image segmentation has played and continues to play a noteworthy role in a variety of fields, from medicine to robotics. For example, segmentation followed by recognition or classification is compulsory for the auto-matic identification of malignant cells in digital mammograms [12].

Analyzing the image and retrieving information from this to execute tasks is a significant compo-nent of digital image technology, and the segmentation process is the preliminary stage. One of the hotspot areas of image processing and computer vision is image segmentation. To recognize an image segmentation of images is a precondition. It divides a test image across several related groups based on certain characteristics in terms of determining out what individuals are interested in. It also serves as a foundation for image evaluation, recognition, feature extrac-tion [11]. Large data volumes provide difficulties in medical image processing. As the volume of data expands, image processing and visualization methods must be improved. The development of scalable algorithms and advanced parallelization approaches has been enabled by the usage of graphics processing units [13]. Man-ual segmentation is effective in most circumstances but is susceptible to rater drift and bias, making it unfeasible for big datasets due to the repetitive and time-consuming nature of the process. Clinical applications provide better results with automatic segmentation methods if they have: Potential of segmenting as an ex-pert, Exceptional results from a diverse variety of datasets, Processing speed that is realistic [14]. Although expert segmentation cannot be considered 100 percent accurate in all circumstances, it can be viewed as a good starting point. Systematic perceptual faults are another potential, as are any kind of error induced by neglect that may emerge as a result of the tediousness of the segmentation operation [14].

Image segmentation, according to the current math-oriented concept, is a method for splitting down a large image in smaller pieces "I" into "n" number of subregions R1,R2,......Rn such that

$$I = U_{i=1}^{s} R \quad and \quad R_i \cap R_j = \Phi, i \neq j$$

Where

$$I = U_{i=1}^{s} R$$

$$R_i \quad Represent \quad connected \quad regions \quad \forall_i$$

$$R_i \cap R_j = \Phi \ \forall_i^j$$

$$P(R_i) = TRUE \quad \forall_i, \quad When \quad i \neq j$$

$$P(R_i \cap R_j) = FALSE \quad when \quad i \neq j$$

The uniformity predicate P(R_i) is correct for all points in the set R_i, but P($R_i \cap R_j$) is false when R_i and R_j are adjacent.

The general flow of the image segmentation process is depicted in figure 2.1 [8].

The fundamental purpose of segmenting images is to partition the image into visually distinct, identical, and significant sections based on some characteristic or evaluated attributes that are domain-independent. In addition to their local neighborhood, many segmentation methods leverage two fundamental attributes of pixels: discontinuity and similarity or uniformity.

Image segmentation would have been straightforward if image noise, weak object borders, non-uniform object field, weak contrast, and other elements impacting images haven't been presented [15].

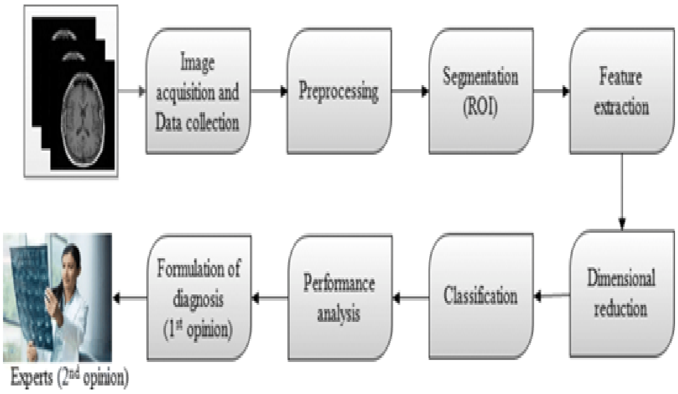

FIGURE 2.1: Generic Process Flow of Brain Tumor Detection System[8].

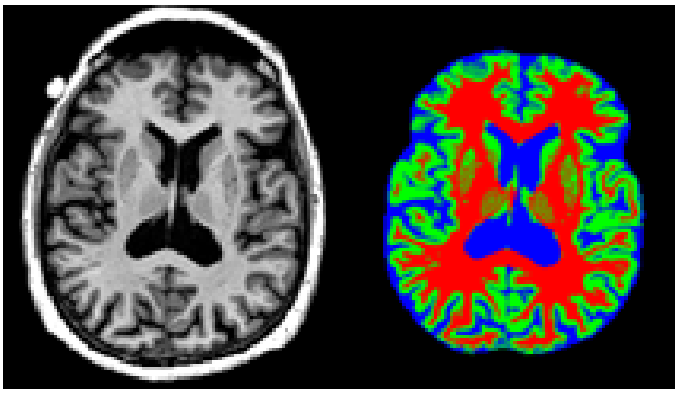

FIGURE 2.2: Brain Tumor Detection [15].

2.1.3 Image-Segmentation using Pixel Data

Several Artificial Neural Network techniques for extracting features effectively using pixels or voxel data have already been proposed. Grossberg's perception model, which is proficient in segmenting images based on their surface and shade, and Opara and Worgotter's brain-resembling networks are two examples of biologically inspired classifiers.

The goal of segmentation, when seen as a classification model, is to give labels to pixels or voxels. Many neural-based methods execute segmentation centered on the pixel information, which is acquired out of a convolution window or by-passing the data in the context of visual quality to a neural classifier [16]. To merge ANNs on multiple abstraction levels, hierarchical segmentation algorithms have been devised. The two guiding ideas of classified methodologies are concentration and bottom-up processing. Low-level feature extraction is handled through one or even more Ann models, and their results are integrated at a high level of abstraction, wherever the final image segmentation is performed by another (neural) classifier.

Modular, non-hierarchical techniques have also been proposed. Pixel-based ANNs used the following criteria to classify image content:
Texture and texture combinations, as well as local shape [16]. The structure of interest is defined by voxels with outputs greater than 0, while the outer region is defined by voxels with outputs less than 0 [17].

2.1.4 Image Segmentation Based on Features

In many image processing applications, the initial step is to extract a description of the image in terms of a set of relevant features. For image segmentation, ANNs are used in a variety of feature-based methods [18]. To conduct feature-based image segmentation, numerous forms of ANNs have been trained: SOMs, versions of radial basis functions network and CNNs, Hopfield ANNs, and a dynamic ANN are examples of feed-forward ANNs [16]. Feature-based artificial neural networks are being trained to categorize images differing in their properties, such as:

- Texture
- A texture-and-local-shape combination

The most frequent segmentation task done by the feature-based artificial neural network is texture discrimination and is dependent on:

- Matrixes of co-occurrencesWavelet features
- Gabor wavelets' multi-resolution features
- Linear scale-space spatial derivatives

Gabor and wavelet-based features provide information to the classifier at multiple levels; nevertheless, a classification algorithm must explicitly deal with scale fluctuations. Whenever it refers to coordination, Gabor and wavelet-based algorithms are highly sensitive to horizontal, vertical, and diagonal features. Rotation invariance can be produced by combining these three directions into a local orientation measure [16].

2.2 Techniques of Segmentation

Over the last few decades, imaging has become a fundamental component of most applications in everyday life, necessitating the development of precise and reliable image segmentation techniques. Image segmentation is categorieed into two types: The term "local segmentation" refers to the process of dividing a particular region or part of an image, Global segmentation, on the other hand, is focused on categorizing the entire image, that comprises many pixels. Segmentation methods can be classified into two categories centered on image properties: Discontinuity-based image segmentation (e.g., region segmentation) and similarity-based image segmentation (e.g., thresholding techniques, region growing techniques, and region splitting and merging fall into this category) [19]. This chapter will go over these segmentation approaches in detail.

2.2.1 Manual Segmentation Method

To perform manual-segmentation the radiologist has to use the multi-modality information provided by the magnetic resonance images, as well as structural and biological data acquired by training and practice. The technique requires the radiologist to observe many pieces of images, diagnosing the tumor, and precisely sketching the tumor states [20]. To extract the contour of the target structures. Typically, an experienced operator has gone through about eighty 512 * 512 images slice by slice. Manual segmentation allows users to completely use their medical knowledge, and if we trust the operator's proficiency, it is one of the most precise image segmentation algorithms. As shown by numerous intra- and inter-operator variability studies, Manual segmentation takes too long and is vulnerable to error. Experts generated segmentation results sometimes measured as the ground truth. On the other side segmentation outcomes/results generated by experts are often difficult to regenerate, since even seasoned operators show considerable inconsistency in their previous mapping [9]. Based on similarity and discontinuity, image segmentation is categorized into two classes such as edge-based segmentation and region-based segmentation. Discontinuity-based approaches are edge or boundary-based, whereas those that rely on similarity are called region-based methods [15].

2.2.2 Edge Based Methods

Edge-based algorithms used rapid fluctuation in pixels intensities of an image because a single pixel intensity is insufficient for determining edge statistics [19]. Detecting the edges creates an edge image, which is then followed sequentially by edge pixels with adjacent neighbor connectivity and procured into a list to reflect an object boundary [21]. Edge detection algorithms locate edges at which intensity's first derivatives exceeds a particular level or the second derivative has no crossings. Edge-based segmentation algorithms identify the edges firstly, then determine whether or not to combine them to construct the boundary, which is then used to segment the appropriate regions [19]. The perimeter of the recognized boundaries needs to be about equal to that of the object in the input image for segmentation precision. This led to some other presumption: that every image's region of interest has a perimeter that may be recognized using gradient or second derivative methods [15]. Edge detection approaches include the Sobel operator, canny operator, Robert's operator, Laplacian operator, and others are used to detect edges [19].

2.2.3 Sobel Operator

In digital imaging, the Sobel operator is primarily used to detect the edges of an image. This is a discontinuous differential operator that would be employed to compute the value of the image intensity function's estimation. The Sobel operator is a popular first derivative-based edge detection operator [22]. The Sobel operator has a weighted effect on the pixel's location, which is vastly inferior to the Prewitt and Roberts operators [11]. The weights can be represented as a numerical array in the form of a mask kernel or window that corresponds to the local image neighborhood [11]. The Sobel operator is comprised of two sets of 3x3 matrix refferd as diagonal and linear models, that are presented over an input image to determine the variance across horizontal and diagonal variations [22]. The following formula could be used to quantify the magnitude of the gradient by integrating the horizontal and vertical gradients estimations of each pixel in the image:

$$G = 2\sqrt{Gx^2 - Gy^2} \tag{2.1}$$

Since it is differentiated by two rows and columns, the edge elements on both sides are amplified, giving the edges a very bright and dense appearance [23]. The following formula given in equation 2.2 can be used to compute the gradient.

$$\theta = arctan\frac{G_y}{G_x} \tag{2.2}$$

The angle θ has zero value, which means the image has a longitudinal edge, and the left side of the image is darker than the right.

2.2.4 Laplacian Operator

The Laplace operator is a second-order differential operator that is isotropic. It's more effective when the edge's position is the only point of concentration, irrespective of the voxel's greyscale variation around it [22]. The immediate reaction of the Laplace operator to isolate pixel intensities is much greater than the edges or linear operator. As a result, it only succeeds to images that are free of noise. Before detecting the edge, the Laplacian-operator must implement a lowpass filter in the presence of noise. As a result, the conventional segmentation algorithm forms a new operator by incorporating the Laplacian and smoothing operators [11]. The Laplacian operator Δ^2 of a function f(a,b) is defined as below in equation 2.3

$$\Delta^2 f(a,b) = \frac{\partial^2 f(a,b)}{\partial a^2} + \frac{\partial^2 f(a,b)}{\partial a^2} \tag{2.3}$$

2.3 Region-Based Segmentation Methods

The region-based classification splits an image into segments that are identical or homogenous regions of associated pixels by implementing homogeneity requirements to all feasible groups of pixels. Some attributes, including colors, intensity, and structure, are shared by pixels in a region [15][18]. Most of the pixels inside the area of interest for the indexed segment, as well as the neighboring upper and lower sections subsequently transformed from the input data of composite spectral bands intensity values to an output vector [24]. Region-based segmentation algorithms work iteratively by grouping pixels that are next to one other and have similar values together, and detaching groups of pixels that have different values. These strategies assume that the divisions produced are related to the objects or significant areas of image. Microscopy, computer tomography (CT), Ultrasound, magnetic resonance imaging (MRI), Positron Emission Tomography (PET), and mammography are among the images that cause challenges for segmentation approaches that use global information in an image [15]. Following methods are used to segment the image such as:

2.3.1 Seeded Region Growing

The most well-known approach for segmenting images is region growing, which is commonly used for medical image segmentation [25]. Region-growing is also termed as region integration or merging, this technique is utilized to extract a region of interest of an image that is composed of collections of pixels/voxels with similar intensity. Region-growing is also identified as region merger, which is a technique for merging voxels with similar intensities. It starts with a seed point specified through an operator manually or by using an automatic seed finding technique. Then, if the intensities are close enough, the region expanding or growing adds all pixel values to the increasing region (sustaining a predefined uniformity or homogeneity criterion). This process is repeated until the area can no longer accept any more pixels/voxels [26]. Growing regions is an efficient strategy to segment volumetric images that are a complete fabrication of massive interrelated

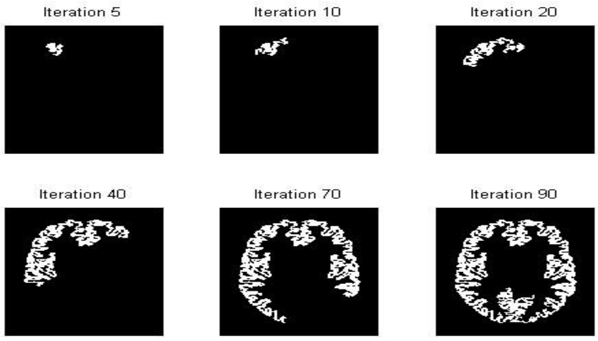

FIGURE 2.3: Region Growing Method [26].

homogeneous regions. As a result, in medical image processing, it can successfully segment various tissues, structures, and disorders from MR data. The growing regions may flow freely and overlap with non-interesting regions. if the origin point that is seed point and uniformity criteria aren't adequately defined. Furthermore, region-growing is susceptible to noise, and segmented areas may become detached or have holes in the presence of noise [27]. This technique has several advantages, including the ability to select several criteria concurrently and the capacity to produce excellent outcomes with less noise in the images. However, there are some drawbacks to this approach. For example, if the seeded area growing method is used, noise in the image may cause the seeds to be incorrectly positioned, and when the image is noisy or has strength differences, over-segmentation can occur [28]. Figure 2.3 below represents the seed growing of the MR image.

2.3.2 Region Splitting and Merging Method

Image-Segmentation techniques such as regional split and recursively merge operations. Splitting is the practice of distributing an image into segments with relatively homogeneous properties, and merging is the process of combining adjacent similar regions [19]. An image can be segmented by the users into a collection of random, disjointed regions and then combine or splitting the regions to satisfy the conditions of real image segmentation rather than selecting seed points [29]. At first, the variance of the entire image is computed. The image is divided into four quadrants if the variance exceeds the defined limit. Consequently, if any of these four quadrants' variance reaches the limit, it is subdivided into four more.

This is repeated until the entire image is made up of a series of squares of different sizes, all with variances below the limit. As a result of the splitting process, there

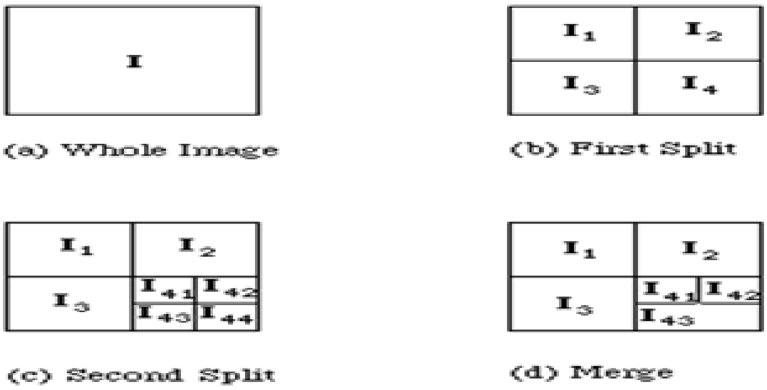

FIGURE 2.4: Region Splitting and Merging [30].

are too many regions of various sizes in the original image with identical intensity values. The splitting and merging process will continue until no more image splits are possible [30]. Figure 2.4 below represents region split and merging of MR image.

2.4 Thresholding Method

Segmentation of an image depends on thresholding seeks to split an image as input into the pixels with two or more values by analyzing the values of pixels to a specified threshold value T [15]. Throughout the images, the value of the threshold T will not change. Pixels with values less than T have been assigned to one category, while the rest were assigned to the other. On the original image, the boundaries between adjacent pixels in different categories have been superimposed in white. The output image X(y,z) can be derived from the original image P(y,z) using T as [19]:

$$X(y,z) = \begin{Bmatrix} 0, & P(y,z) < T \\ 1, & P(y,z) \geq T \end{Bmatrix} \tag{2.4}$$

Where P(y, z) denotes the pixel value at the specified point (y, z) [15]. Thresholding produces a binary image, with pixels labeled 1 corresponding to the object and pixels labeled 0 correspondings to the background. Thresholding can also be recycled in MRI segmentation to differentiate background voxels from brain tissue or to begin tissue categories in iterative subdivision approaches similar to FCM clustering. Pixels are categorized using range values or threshold values, and thresholding techniques such as global and local thresholding are used [31]. The term "global thresholding" refers to a technique in which T is solely determined by the image's grayscale qualities as well as the value of T is exclusively defined by the structure of pixels [29]. For the entire image, the global thresholding method selects only one threshold value. While different threshold values are chosen for different regions using local thresholding. Thresholding does not take into consideration the information of neighboring pixels [32]. The process of determining the value of T in image segmentation research has been a focal point [28]. In addition to thresholding entropy-based threshold segmentation, minimum error method, Otsu, optimal-thresholding, histogram-concave analysis, iterative-thresholding, entropy-based thresholding, MoM-keeping thresholding, and more approaches are available [29].

2.5 Segmentation Based on Clustering Methods

Clustering is an unsupervised research method in which data items are divided into groups called clusters by remembering two characteristics: (1) high cohesion and (2) low coupling. According to the first property, data items belonging to one cluster must have a high degree of similarity. The second property specifies that data objects in one cluster should be distinct from those in other clusters [23]. The most widely used clustering techniques appear to be K-means clustering, FCM clustering, and the expectation-maximization method [33]. The K-means clustering approach divides the inputs into K classes and sequentially evaluates the mean intensity for every class, subsequently splits the image by identifying the voxels in the group with the nearest centroids. The k-means clustering is generally referred to as a hard classification technique as every voxel must correspond to one category for each cycle. The FCM is a soft clustering technique relying on fuzzy set theory. This is a k-means clustering modification that permits each pixel/voxel to correspond to multiple groups depending on a correlation value [9]. The goal of the FCM method is to reduce the given optimization function:

$$J(m) = \sigma \left(\sum_{i=1}^{C} \sum_{j=1}^{N} \mu_{ij}^{m} D_{ij} \right) \tag{2.5}$$

Where

$$D_{ij} = \|X_i - Y_j\| \tag{2.6}$$

where N represents the total of image components that need to be split into C clusters, $\mu_(ij)$ is the association function of the ith cluster's element x_j (an input model at the jth position), m is a weighting exponent that regulates the partition's fuzziness, and D_ij is the measure of xj similarity to the ith cluster center vi [9]. And objective function for k-mean is known as squared error function, which is given by:

$$j = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| X_i^{(j)} - C_j \right\|^2 \tag{2.7}$$

Where k represents the number of clusters and n reflects the number of samples, and C_j is a given centroid [9]. The distance between data points and cluster center is represented by $\|\|$.

2.6 Training Algorithms of ANN

An Artificial Neural Network is a biologically inspired computational model which is composed of functions and neurons. The model requires training before it could solve a problem. The ANN model can be trained to produce output that is analogous to what is predicted [34]. It's more difficult to say which training method has been the most effective or precise for a given situation. Since multilayer learning machines have been known for a long time, however, their efficient implementation for practical tasks has been hindered by the lack of suitable training algorithms [35]. Following are the factor that influences the training of network including the problem's complexity, the quantity of datasets in the training phase, the network's connection weights and bias values, the error goal, or even though the network is deployed for object recognition or prediction of function [35]. ANN has investigated several training algorithms, including the Powell-Beale-Conjugate Algorithm, Polak-Ribiere-Conjugate-Gradient Algorithm, Levenberg-Marquardt, Scaled-Conjugate-Gradient Back Propagation, Resilient-Back Propagation, One-Step-Secant Back Propagation, Fletcher-Reeves-Conjugate-Gradient Algorithm, and Quasi-Newton Algorithm with (BFGS). To find the best training algorithm, the identical structural parameter setup is used to train each neural network architecture [36].

2.7 Related Work for Image Segmentation

The volume of MRI data sets is growing exponentially from day to day. As a result, it is essential to accelerate medical image processing and disease diagnosis technologies while sustaining a higher degree of accuracy. In this context, the researcher has proposed various segmentation techniques with widely differing degrees of accuracy and complexity in the literature discussed below. The author of the study [37], provided a review regarding the most often used segmentation and classification approaches for brain MR scans from the year 2014 to 2019. In addition, the author presented a hybrid strategy for Medical image processing. The authors used a feed-back pulse-coupled neural network as a front-end processor to segment images and discover ROI, then used the DWT to retrieve features from the MRI dataset. After that, the wavelet coefficients' dimensionality is reduced using PCA, leading to a further precise and effective classifier. The condensed attributes are directed to a back-propagation neural network, which uses feature selection parameters to identify inputs as normal or abnormal. The author of the study [37] performed experiments on Axial, T2-weighted, 256–256 pixel MR brain images. The dataset was obtained from the Harvard Medical School website. The model achieves a classification accuracy of 99 %, with a sensitivity rate of 100%

and a specificity rate of 92%. The study concludes that the suggested classification approach will effectively discriminate among balanced as well as pathological cases, improving the accuracy of human brain disease diagnosis. The author of the study [38], proposed a system by using FCM for segmentation and classification of a brain tumor as benign (not-spreadable) or malignant (spreadable and contains cancerous cells). FCM is a soft clustering technique -based on fuzzy set theory. Each pixel in a picture can be classified into multiple groups. pixels are consigned toward a class according to a definite association value. The algorithm's execution time(s), MSE, PSNR, and segmented area (pixels) are all found as evaluation parameters. The mean square error is the collective square error concerning the compressed and original image, while the peak signal to noise ratio indicates the amount of the peak error. The computed error will be lower when the value of MSE is lower. The excellence of the compressed or reconstructed image will be better if the value of PSNR is higher. Wavelet coefficients in an MR image are extracted using a DWT. It is a type of linear transformation that is used to convert an input data vector into another vector. It splits the data into various frequency components such as HH, HL, LL, and LH. In the study [38], it is utilized to extract image features. because its multi-resolution diagnostic property helps it to analyze an image at various resolutions. Then the PCA is used to reduce dimensionality. It's used to reduce the dimensions of data by selecting the most appropriate from a wide pool. As a consequence, a new function matrix with reduced dimensionality is retrieved. The dataset is then translated to GLCM and arithmetical features including mean, standard deviation, entropy, variance, and so on are extracted. For the SVM training process, the extracted feature database is used. The authors of the study used a private dataset of brain MRI and found that the proposed automated method had 98.82% accuracy, 100% sensitivity, 97.83% precision, and a 1.17% error rate [38]. In another study Song et al [39] , presented a modified scheme of FCM that is MRFCM. This technique incorporates the spatial information of neighborhood pixels and has strong anti-noise capability. The author of the study [39] applied the MRFCM algorithm on three types of data set such as synthetic-images, simulated brain MRI images, and real brain images, and after that, the results have been associated with typical FCM results. The results reveal that MRFCM is noise-resistant and can efficiently segment distorted mri Images. The author of the study [40] proposed a framework for fully automated segmentation of MR brain images.

A three-step segmentation approach is used in the framework. First, non-brain structures are removed which is also known as skull stripping by using the level set method. After that, the non-uniformity rectification method then uses approximations of tissue strength discrepancy to recompense for non-uniformity. Finally, it employs a statistical model relying on Markov random fields (MRF) for MRI brain image segmentation, also known as brain tissue classification. The brain tissues are categorized into the cerebrospinal fluid (CSF), white matter (WM), and gray matter (GM). The author of the study [40] experienced the anticipated framework both on simulated and real MR im-ages. Kong et al. [41] established a technique for segmenting the skull and various brain tissues. The authors used pre-processing techniques such as wavelet denoising to obtain the precise features of the skull and various tissues such as WM, GM, and CSF. CNN is executed for the automatic segmentation of images. The authors of the study [30] also used parallel computation to accelerate the entire process. The author of the study [30] measured the percentage of each brain tissue, which can be used to diagnose various disorders (e.g., cerebral atrophy), which is mostly caused by a decrease in GM or WM brain tissue. Experiments in the study [30] were carried out on a private dataset, and the findings were compared to the VCH model, yielding an accuracy of over 90%.

Khorram et al. [42] implemented the segmentation of gray matter (GM), white matter (WM), cerebrospinal fluid (CSF), and thalamus (large mass of GM) from an MR brain image. The author proposed an enhanced thresholding technique for segmenting MR brain images using the ACO technique. The researchers proposed the scheme in [31], which used textural features for heuristic details, as well as post-processing image enrichment to accomplish improved output based on homogeneity. The authors of the study [42] experimented with the BRATS-2012 dataset and achieved accuracy up to 97%. The authors compared the accuracy of work with PSO, K-means, and EM approaches using 20 MR scans.

2.8 Related Work for ANN Training Algorithms

To train ANN, an effective training algorithm must be predicted. Prediction of training algorithms to train ANN has been a major topic in research. Many attempts have been made to produce a realistic algorithm capable of accurately predicting with a high degree of accuracy. The ANN employs a variety of training functions. An optimum training function must be used in the ANN to predict the least amount of error. Training algorithms of ANN have been utilized in many fields of daily life to predict the best one training algorithm. The literature listed below contains the work of different scholars who compared training algorithms on different datasets and recommended the efficient algorithm based on their findings. The author of the study [43] proposed a model for forecasting the categorization of soil types, and the optimum neural network training algorithm was found. A repository of 120 soil prototypes was employed in the study. To train the neural net, the following algorithms were employed: back-propagation, batch-gradient-descent training, batch-gradient-descent training with-momentum, variable learning speed Scaled-conjugated gradient, Fletcher-Reeves conjugated gradient, Pollack-Ribier-conjugated-gradient, Paul-Bill-conjugate-gradient, quasi-Newton (BFGS), quasi-Newton, one-step-secant, and Levenberg-Marquardt conjugate gradient. The RMSE, CRM, and COD were used to assess the efficiency of the proposed model. By comparing actual and predicted values, the Levenberg-Marquardt algorithm with COD = 99.88% and RMSE = 0.0521 is the best training method for predicting soil type classification. Then the back-propagation method and quasi-Newton BFGS can accurately train the network. It's worth noting that the data used in this study was limited. When the quantity of data points in a neural network is increased, the accuracy of the training can greatly improve. The author of the study [44] looked at how artificial neural network (ANN) modeling techniques could be used to predict the properties of waste tire steel fiber reinforced concrete (SFRC). For training, five ANN algorithms were investigated, namely incremental-backpropagation (IBP), batch-backpropagation (BBP), quick-propagation (QP), Levenberg-Marquardt-backpropagation (LM), and genetic algorithms (GA). In descending order, the ANN training algorithms' predicting ability performance was GA >IBP >LM >BBP >QP. Constructed on statistical metrics of performance such as Root Mean (R), Root Mean Square (R2), and AFV, for analysis and forecasting the strength of concrete fiber reinforced derived out of used tires, the Genetic-Algorithm (GA) was considered as the optimum approach. According to the findings, backpropagation is not as effective as the genetic algorithm (GA) iterative method. As a result, this study finds that choosing the correct learning algorithm is crucial to simulate inputs using ANNs effectively and that using a Genetic-Algorithm to achieve high prediction efficiency is extremely suggested in ANN modeling [44]. Author of the study [36] The study's authors developed an artificial neural network model to approximation the compressive power of concrete using an optimal dataset of representative concrete mix proportions. Based on the eight distinct training techniques, eight separate neural network models are constructed; each artificial neural classifier is trained with the very same structural configurations. LM is the most feasible possible training method, with R (correlation) equal to or better than 95%, after LM, BFG seems to be another credible training function with the same architecture and design parameters on average, considerably better than 93% on average. The ANN models are repeated with other activation functions realistic to the hidden layer neurons, such as tangent-sigmoid and log-sigmoid functions. One of the best available tools for predicting concrete compressive strength is an ANN model using the learning function Levenberg–Marquardt (LM).

The author of the study [45] conducted research in which they optimized an ANN model for predicting Polymeric Inclusion Membranes (PIMs) Cr(VI) removal effectiveness by using the right training algorithm. The outputs from the created models and dataset acquired from laboratory experiments forecasting of PIMs Cr (VI) removal efficiency are assessed, compared to discover the optimal training algorithm. The predictive ability of ANN models constructed and trained using three different methods comprising Levenberg-Marquardt (LM), Bayesian-Regularization (BR), and Scaled-Conjugate-Gradient (SCG), was evaluated [45]. Based on this computation, LM was discovered to be the superlative learning algorithm for PIMs Cr (VI) removal efficiency estimates. The LM training algorithm outperformed the BR and SCG in predicting PIMs Cr (VI) removal effectiveness is calculated using six inputs: time, extraction type, extraction quantity, density, polyester type, and polyester amount, as demonstrated by R2 values of 0.97, 0.95, and 0.72 in test dataset for LM, BR, and SCG, accordingly. Lastly, each training method's precision of prediction ability was investigated, yielding the following results: For train, validate, and test data sets, LM >BR >SCG is the preferred method [45].

Jaiswal et al. [22] studied the impact of several training strategies for ANN on the feedforward backpropagation architecture. For analysis, MATLAB R2017a is used to design and run the training algorithms. Several training algorithms are used to achieve the greatest level of accuracy between expected and definite results. The study is based on the Kaggle dataset for housing price prediction. This study concluded that if network training must be finished quickly, the Scaled Conjugate Gradient technique should be employed; but, if a small error cannot be tolerated, the Levenberg-Marquardat approach should be utilized at the cost of additional time and processing power.

Tabbussum et al.[34] investigate the possibilities for employing artificial neural networking as a machine learning technique to estimate flood predictions by obtaining data from multiple sources. Flood forecasting models were constructed using five modern ANN computational approaches and 5 different training algorithms: Bayesian-Regularisation algorithm (ANNbr), Levenberg–Marquardt algorithm (ANNlm), Conjugate-Gradient algorithm (ANNcg), Resilient-Backpropagation algorithm (ANNrbp), and Scaled-Conjugate Gradient algorithm (ANNscg). All of the models were evaluated using statistical measures such as MSE, root means square error (RMSE), coefficient of correlation (R2), and absolute average deviation (AAD). The following is the sequence in which the models' MSE values increased: ANNlm <ANNscg <MLR <ANNcg <ANNrbp<ANNbr. The best model for predicting floods was the ANNlm model. The Levenberg–Marquardt model was found to be more predictive and trustworthy. When the ANN findings are compared to the MLR results, it is evident that the ANN has a significantly higher predictive capability [46].

Another study conducted by COSKUN et.al [35] found that the precision of MLP, that employed in several neural applications, is completely dependent on the learning algorithms used. Even though multi-layer perceptron is not frequently utilized for image organization, the results reveal that some MLP training algorithms are just as good as other network topologies in terms of providing a solution. The outcomes demonstrate that the 'trainlm' method outperforms MLP on feature extraction data from the vision group. The 'trainbr' algorithm is effective, nevertheless as good as the 'trainlm' approach.

TABLE 2.1: Comparison Table of Image Segmentation.

Research Paper	**Technique**	**Accuracy**	**DataSet Used**
El-Dahshan et al, [37]	DWT, PCA	99%	Harvard Medical School
Srinivas et al, [38]	FCM	98%	Private
Khorram et al, [42]	ACO	97%	BRATS
Song et al, [39]	FCM and MRFCM	MRFCM > FCM	Synthetic, simulated, and real brain images
Kong et al, [30]	CNN	96%	Private

TABLE 2.2: Comparison Table of Training Algorithms

Research Paper	**Technique**	**Best Training Algorithm**	**DataSet Used**
Hassanneja et al, [43]	ANN	LM	Private
Awolusi et al, [44]	ANN	GA	Private
Chopra et al, [36]	ANN	LM	private
Jaiswal et al, [34]	FFBPN	LM	Kaggle
Tabbussum et al, [46]	NN	LM	Various Agencies

According to the literature, training algorithms have been used for a long time to train Artificial Neural Networks on various data sets such as concrete data, BRATS data, and a variety of fields have utilised training algorithms to train ANN for their own data types. However, we are still unable to find any research that utilised a variety of training algorithms on Synthetic MRIs.

Chapter 3

Fundamentals of ANN and Training Algorithms

The basics and architecture of ANN are covered in this chapter. It also goes through the specifics of various ANN network training algorithms and the value of employing ANN for medical image processing.

3.1 Basic Structure and Strength of ANN

An ANN is a supervised information processing system, inspired by the human nervous system. A computing system made up of several simple, densely interconnected nodes (neurons) with linear or nonlinear transfer functions is referred to as an ANN [47]. From the past few decades, the Artificial Neural Network (ANN) has played an important role in structural analysis, simulation, sensors verification, pattern recognition or development, and automatic control optimization as a black-box technique [47]. For creating an accurate and trustworthy model, choosing the proper parameters of ANN as inputs and outputs are very critical [48]. The efficiency of artificial neural networks (ANNs) for solving a variety of issues, such as pattern recognition, speech recognition, image segmentation, and image classification in digital image processing, is increasing with each day [49]. Miller

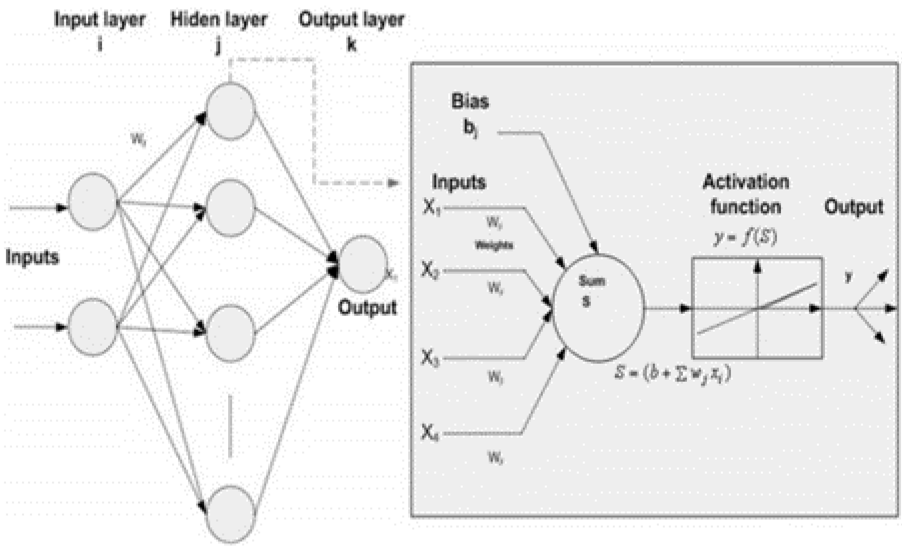

FIGURE 3.1: Basic Structure of ANN [37]

et al. [50] published a detailed analysis of ANN in 1992, predicting that the application of ANN in medical image processing will grow exponentially in the future. Their prediction conclusively demonstrated. According to their Google scholar search results, Shi et al. [51] found almost 33,000 items in the field of medical image processing with ANNs in the last 16 years. More than 500 books and even more than 20 journals have been published on the subject of ANNs [51]. Figure 3.1 represents the basic structure of ANN.

A Neural-Network is made up of the basic processing elements known as 'neurons' or 'nodes,' which have just a superficial resemblance to real neurons. Unidirectional connections of varying strength or weight connect each neuron to the different types of neurons in the network. Fig 3.2 denotes the basic structure of neurons.

Perceptron processing elements combined their weighted inputs and applied a linear transfer function to produce an output were used in the early designs [50]. The neuron calculates the weighted sum of the input signals and compares the result with a threshold value, θ. In case of the net input below a threshold value, the output of the will be -1, and if the net input is equal to or greater than the threshold value, the neuron will be triggered and its output will be +1, as shown in equation 3.1 below The activation function used by the neuron is illustrated in equation 3.2.

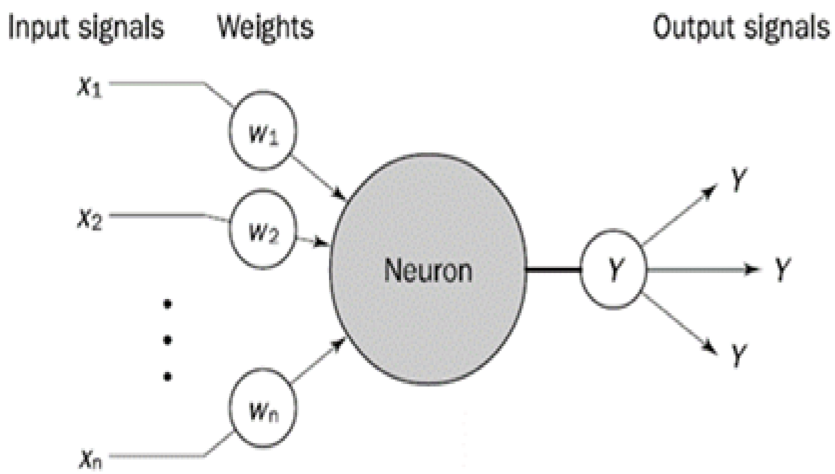

FIGURE 3.2: Activation Functions of ANN [52]

$$y = \begin{Bmatrix} +1, & if x \geq \theta \\ -1, & if x < \theta \end{Bmatrix} \tag{3.1}$$

$$X_i = \sigma \left(\sum_{j=1}^{n} Y_{ij} Z_j + T_i^{hid} \right) \tag{3.2}$$

Where 'θ' represents the activation function, n represents the input neurons, $Y_i j$ are the weights, Z_j inputs to the input neurons, and T_i^{hid} is the threshold terms of the hidden neurons. The objective of the activation function is to bind the value of the neuron so that the neural network is not disrupted by divergent neurons, in addition to bringing nonlinearity into the neural network. The sigmoid (or logistic) function is a typical activation function example. Thus, using a sign activation function, the actual output of a neuron can be defined as in equation 3.3 [53].

$$Y = \sin n \left(\sum_{i=1}^{n} w_i x_i - \theta \right) \tag{3.3}$$

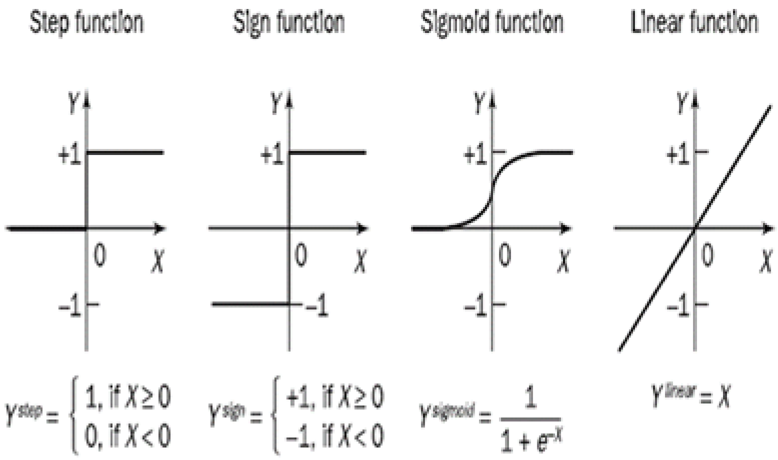

FIGURE 3.3: Activation Functions of ANN [52]

3.2 Choice of Activation Functions

Many activation functions have been investigated, but only a few have proven useful. Figure 3.3 depicts four popular functions: step, sign, linear, and sigmoid [54]. For classification and pattern recognition tasks, the step and sign activation functions, also known as hard limit functions, are frequently utilized in decision-making neurons. The value between plus and minus infinity is converted to a value between 0 and 1 using the sigmoid activation function. Back-propagation networks practice this type of activation function. [52].

3.3 Supervised and Unsupervised Learning

Adaptive approaches for machine learning enable computers to learn from examples and experiences. Improving the learning process will help an intelligent system perform better over time. Supervised and unsupervised learning are the two types of learning. In supervised learning, a network is trained with inputs and outputs in supervised learning (targets) [55]. Each training instance will have several inputs and one or even more corresponding output values, and the goal will be to reduce the network's overall output error by Using a specialized training approach, repeatedly adjusting the neurons connections weights are adjusted levels for all training cases [55]. Human interaction is essential in the supervised learning process for feature extraction, extracted feature verification, verification of generated outputs, and decision-making. In general, supervised learning produces better results when the training data available is of good quality [55] Unsupervised learning takes place without the interference of a supervisor or analyst. The training dataset in There is no objective data in unsupervised learning. In most cases, a procedure is arranged to evaluate the network's suitability or reliability [56]. This function, typically termed as an objective function, is dependent on the network's functionality and provides a current network cost. configuration by integrating the input parameters with the output of the network. Unsupervised learning usually aims to reduce or maximize the cost of all input vectors in the training set [56].

3.4 Feed Forward Network and its Types

One of the most prevalent neural network architectures utilized in biomedical fields is the feed-forward network. In a feed-forward network, the neurons in each layer are just connected to those in the same layer. in the next layer. These connections are unidirectional, which indicates that signals or information can only move through the network in one way, from the input layer to the output layer, through the hidden layer(s) [55]. The backpropagation (BP) supervised learning algorithm is often used in feed-forward networks to determine appropriate weight and bias values for each neuron in the network. Every layer's neuron is connected to the neuron in the subsequent layer. Input is sent to the input layer for training in the first step of backpropagation [57]. The input pattern is passed through the layers until it reaches the output layer, which forms the output pattern. If the obtained output does not match the required output, an error is calculated and sent backward across the network from the output layer to the input layer. The weights are updated according to the learning rule, which delivers the neuron weight modification mechanism, as the error is discovered [57]. A multilayer perceptron (MLP) is a three- or more-layer feed-forward network with nonlinear transfer functions in the hidden layer neurons. The input signals travel through the layers in a forward direction. Each layer in MLPNN has its own set of capabilities. The first (input) layer receives and distributes real-world input signals to all hidden layer neurons [58]. Because there aren't many computation neurons in the input layer, it doesn't do any computations on input patterns. The output layer receives output signals and uses the hidden layer's properties to create the overall network's output pattern. Commercially available ANNs can be three or four-layered, with one or two hidden layers and 10 to 1000 neurons in each layer. An MLPNN learns in the same way as a single perceptron. The network is provided with a set of training input patterns, and the network determines the output pattern. If there is a difference between the real and desired outputs, commonly known as error, the weights are reviewed and revised to minimize the error. A single perceptron has only one weight for each input and only one output, whereas a multilayer network has many weights, and each results in multiple outputs [58].

3.5 Training Algorithms and their Description

An ideal training function must be employed in the ANN to predict with the least amount of error. The selection of the best training algorithm can improve network prediction accuracy. Several training functions and techniques were employed in the ANN modeling for this purpose. The effects of several algorithms are explored, and the optimal learning approaches for image segmentation are proposed. Various training algorithms are explained in this chapter, here we'll look at how to solve the segmentation issue.

3.5.1 Bayesian-Regularization-Backpropagation

The challenge of selecting the best network architecture is reduced by using a training algorithm to generate networks that simplify well. Bayesian-Regularization is a network training function that uses Levenberg-Marquardat optimization to update the weight and bias value [43]. The correct combination of squared errors and weights is established, resulting in a network with higher generalization properties. The method is recognized as Bayesian regularisation [34]. The disadvantage of this technique is that it calculates performance as a mean or sum of squared errors using a Jacobian [35].

3.5.2 Levenberg-Marquardt

The Levenberg-Marquardat algorithm is a technique for iteratively solving nonlinear least-squares complications. This is a network training function that updates weight and bias values according to Levenherg-Marquardt optimization. Gradient descent and the Gauss-Newton method are combined in the Levenberg-Marquardat algorithm [34].

If the current solution is not adjacent to a local minimum, the technique acts similarly to a gradient descent approach, which converges slowly but steadily. When the present solution is close to a local minimum, it behaves like the Gauss-Newton technique and converges quickly [36]. Levenberg-Marquardt solves challenges in Gradient descent and GaussNewton methods by being fast and having steady convergence. When the training data is large, it provides a memory reduction feature for use [36]. The damped least-square approach is another name for Levenberg-Marquardat [34].

3.5.3 BFGS Quasi-Newton-Backpropagation

Newton's approach is an auxiliary to conjugate-gradient algorithms for speedy optimization. Forming the Hessian Matrix is the first stage in Newton's approach (second derivatives) [36]. It demands the maintenance of a suitable Hessian matrix and requires more effort per iteration than conjugate-gradient algorithms, but congregates in fewer iterations. The computation of the Hessian Matrix for feedforward neural networks is challenging and expensive [43].

3.5.4 Conjugate-Gradient-Backpropagation with Powell-Beale-Restarts

The Conjugate-Gradient (CGP) method was developed by Polak - Ribiere. it has a higher convergence speed on some issues but it necessitates more storing space [36]. Traincgb is a network training algorithm that updates weight and bias values using backpropagation and Powell-Beale restarts [43]. It ensures convergence to the minimal quadratic function without the need to calculate second derivatives [43].

3.5.5 Conjugate-Gradient-Backpropagation with Fletcher-Reeves Updates

Traincgf is a network training function that uses Fletcher-Reeves updates to adjust weight and bias variables based on Conjugate-Gradient-Backpropagation [59]. It requires less storage than CGP and CGB. Although the outcomes will differ from one problem to the next [36].

3.5.6 Conjugate-Gradient-Backpropagation with Polak-Rib iere Updates

Traincgp is a Polak-Ribiere-Conjugate-Gradient algorithm that keeps up-to-date weight and bias parameters using Polak-Ribiere updates. The Traincgp program is comparable to Traincgf in terms of performance [36]. It's tough to say which algorithm will perform best in a given situation. A specific issue PolakRibiere's storage requirements (four-vectors) is slightly higher than FletcherReeves' (three vectors) [60]. In certain situations, it has a better convergence, but it takes up more space [61].

3.5.7 Batch–Gradient-Descent Training Algorithm

The simplest basic training algorithm is Gradient-Descent, sometimes known as Steepest-Descent. It is a first-order approach because it requires information from the gradient vector [43]. Weights and biases are modified in the batch method once all members of the training set have been applied. The estimated gradients for each input are added together until the weights and biases are adjusted [36]. The weights and biases values are updated using the performance function's negative gradient. Gradient-descent has a slow convergence rate, which is one of its flaws. The gradient descent is used to change each variable [62].

3.5.8 Gradient-Descent with Momentum-Learning-Rate-Backpropagation

In addition to gradient changes, the network's momentum allows it to adapt to variations in error levels. By ignoring tiny errors, also makes the algorithm's route smoother. Traingdm is the training function [63]. An appropriate learning rate tries to keep the learning scale factor as large as feasible while maintaining learning stability. The learning rate is adjusted by the uncertainty of the localized error function [64].

3.5.9 Gradient-Descent with Momentum and Adaptive-Learning-Rate-Backpropagation

The 'traingdx' purpose is to integrate adaptive learning rate and momentum training into a single function. Its use is similar to 'traingda' but with the addition of momentum coefficient mc like a learning parameter [43]. Any network with derivation functions in its weights, net-input, and transfer functions can be trained with Traingdx. For the weight and bias variables X, backpropagation is employed to derive performance derivatives. The gradient descent with momentum is used to change each variable [65].

3.5.10 Resilient-Backpropagation

The Resilient-Backpropagation training technique aims to remove the negative consequences of second derivative magnitudes. The weights update's position is determined by the derivative's sign; It would make no difference what the derivative's magnitude is [66].
The size of the weight change is determined by a different update value. In most cases, resilient-backpropagation is considerably quicker than the usual steepest-descent approach. It also benefits from the fact that it only necessitates a minimal storage expansion [67].

3.5.11 Scaled-Conjugate-Gradient-Backpropagation

The algorithm for a Scaled-Conjugate-Gradient trainscg is a supervised learning strategy that utilizes a scaled-conjugate-gradient technique to update the weight and bias parameters [68]. an excellent general-purpose training process is the only conjugate-gradient algorithm that doesn't always entail linear searching [68].

3.5.12 One-Step-Secant-Backpropagation

Since In comparison to conjugate-gradient approaches, the BFGS process needs more data computational resources for each cycle, a secant estimate with smaller memory and processing requirements is needed. The OSS method intends to reduce the gap between conjugate gradient and quasi-Newton (secant) methods [69]. This approach will save time while improving accuracy over ordinary backpropagation [70].

Chapter 4

Research Methodology

This chapter goes through each phase of the proposed technique in detail. The chapter also includes a description of the dataset as well as the experimental setup. whereas Fig 4.1 depicts the proposed technique's scheme.

1. **Pre-Processing**

   Preprocessing is a data mining approach that requires converting a dataset into a format that is understandable to the user. Noise removal, contrast, and the dimension of the image are adjusted. Skull stripping (the removal of the outer ring of the brain picture) is also done. soft-tissues of the brain MR images are the output of this stage.

2. **K-Means**

   In this stage, the K-means clustering method is used to label a skull-stripped brain MR image that only contains soft tissues.

3. **ANN**

   A feedforward-back propagation ANN network is trained with 12 different training algorithms and evaluated on the labeled data that was obtained by K-means, to accurately segment brain MRI into distinct types of soft tissues, such as WM, GM, and CSF.

4. **Post-Processing**

   Following the ANN's outputs, post-processing is used to fine-tune the segmented content. The segmented image is transformed to grayscale before analyzing the performance.

5. **Performance Measures**

   Different performance metrics including DSC, MSE, mean, and STD are measured to ensure the accuracy of our results.

4.1 Proposed Methodology Steps

Step 1: Retrieve images from the database.

Step 2: Perform pre-processing of images e.g., noise removal, contrast adjust-ment, dimensions fine-tuning, and skull stripe that is the exclusion of the outer ring of the brain MRI.

Step 3: After obtaining the requisite images that are the soft tissues only and provide them to the K-means clustering algorithm for labeling the image.

Step 4: Several algorithms are applied for training and simulating the ANN network with this labeled data.
Step 5: After training the network with different algorithms, test data is provided to ANN and segmentation output is generated by ANN.

Step 6: The segmented contents generated by ANN with different algorithms are then post-processed.

Step 7: Finally, the segmentation results of all algorithms are evaluated by using different evaluation metrics.

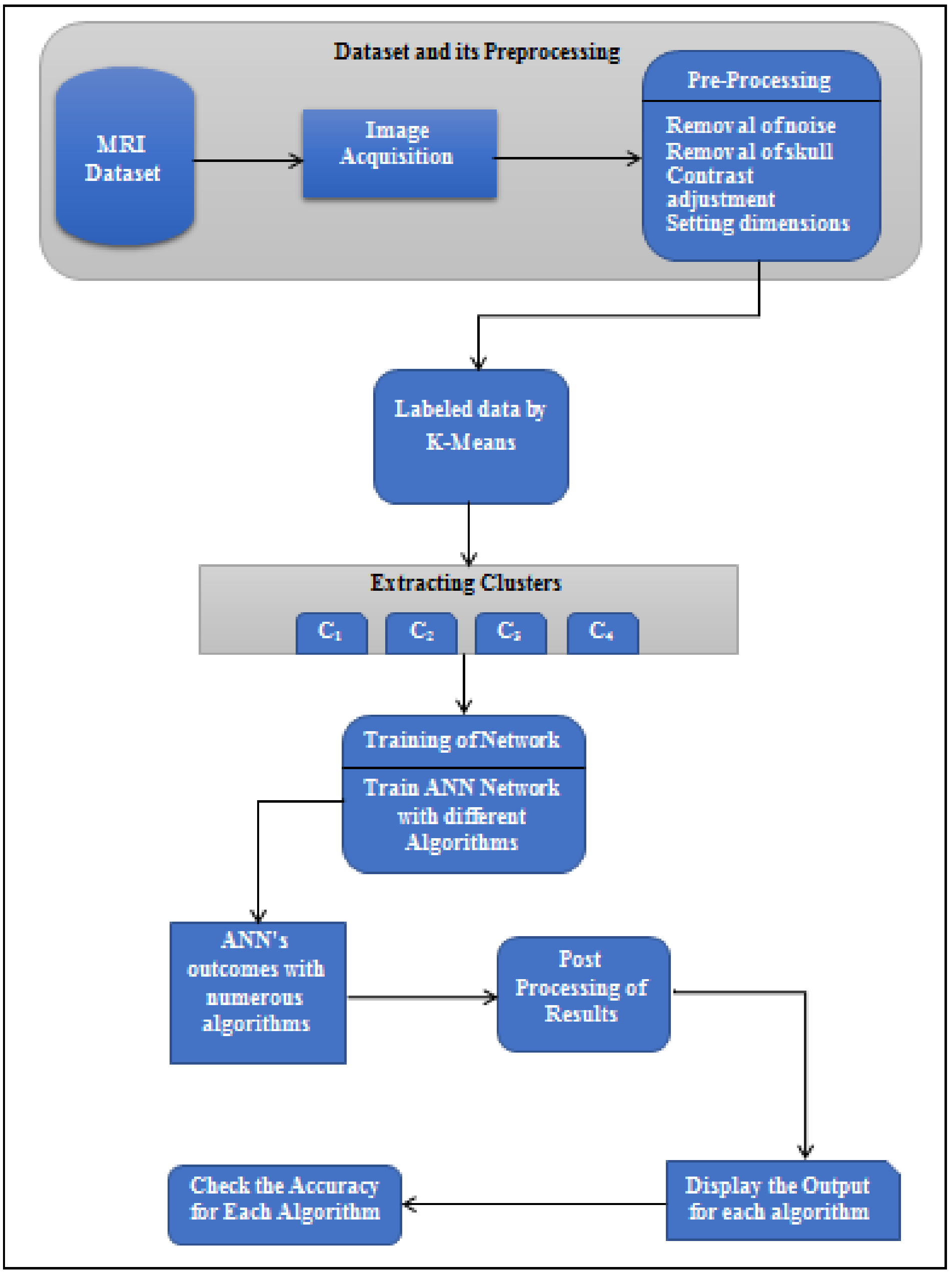

Figure 4.1: Context Diagram of Proposed Methodology.

4.2 K-means Clustering and its Pseudocode

K-Mean is one of the most frequently and straightforward unsupervised-learning methods. This is a hard clustering method [71], which means that each data point belongs to just one cluster.

K-Mean determines centroids at random for data processing, these cluster centers are the starting locations, and the optimum places are found through iterations.

As noted in step 1 begins by working with a set of data items and the number of required clusters (K) [72].

It starts by selecting centroids at random from the data elements for each cluster 'K'. Calculate the gap in between identified datasets and the rest of the items to determined to see if the initial centroids' estimation was correct or not [73]. And allocates the data element to the cluster with the shortest distance between them. It recalculates the centroids by taking the arithmetic mean of each cluster's data elements and reassigning the data elements to the cluster with the shortest distance between the data element and the cluster's centroid repeated the steps until reaching the convergence [74].

4.2.1 Pseudocode of K-means [75, 76]

Step 1: INPUT : Array [72] be the set of input data points and k, number of desired clusters.

Step 2: Initialize: Cluster centroids $C\{c_1, c_2, c_3 ... c_k\} \epsilon R_m$ randomly.

Step 3: For every a_i ϵ 1, . . . , n, set, c_i=argmin j ϵ 1, . . . ,k, $\|a_i - C_j\|^2$

$$C_j = \frac{\sum_{i=1}^{n} 1\{C_i = j\} x_i}{\sum_{i=1}^{n} 1\{C_i = j\}} \tag{4.1}$$

Step 5: Repeat steps 3 and 4 as needed to reach the maximum number of iterations or until the centroids do not change anymore.

Step 6: Output: Clusters $C\{c_1, c_2, c_3 ... c_k\}$.

4.3 Steps of ANN [77, 78]

The following fundamental steps are commonly computed by an ANN with back-propagation.

Step 1: Set the weights and biases to their initial values.

Step 2: For each input layer unit 'q' propagate the input forward.

Step 3: For any hidden or output layer unit "q" the output 'Outq' of an input unit 'q' is its real input value 'Inpq' as Outq = Inpq.

Step 4: To propagate the input forward, compute the net input of unit 'q' concerning the preceding layer 'p' as Inpq = P $\sum \omega_{qp} out_p + \gamma_q$, where '$\omega_{qp}$' is the weight of the connection from unit 'p' in the preceding layer to unit 'q'.'Outp' is the output of unit 'p' from the preceding layers and 'γ_q' represents the bias of the unit.

Step 5: Analyse each 'q' unit's output as $Out_q = \frac{1}{1+\beta^{-1}q}$.

4.4 Pseudocode of Proposed Methodology

1. **for all** training images in the database do
2. input: images
3. preprocessing: remove noise, set contrast, set dimensions
4. skull stripe the images
5. apply k-means to get labeled data for each input image I
6. **end for**
7. **for all** labeled data **do**
8. train ANN by using different algorithms
9. **end for**
10. **for all** test images, Tn **do**
11. test ANN on different algorithms
12. **output:** get a segmented image for each test image T by using numerous algorithms
13. test ANN for different algorithms
14. post-processing: convert segmented image to grayscale image

15. verify precision or accuracy for each algorithm

16. **end for**

4.5 Experimental Setup

Table 4.1: Experimental Setup

CPU	Intel(R) Core (TM) i3-4010U CPU@ 1.70GHz
RAM	4.00 GB
OS	Window
Image Processing Tool	MATLAB R2020a
Image Reading Tool	MicroDicom

4.6 Dataset Acquisition

The data was obtained from BrainWeb, an online medical imaging repository. https://brainweb.bic.mni.mcgill.ca/. It's a database of simulated brain MRI datasets with varying noise levels and INU. Multiple test scenarios are designed using images with a slice thickness of 1mm and numerous levels of noise along with various levels of INU. Table 4.2 contains the specifics of the dataset that was utilised for testing.

Table 4.2: Description of Data Set

Images	Noise- Level (%)	INU-Level (%)
Case-0	0	0% 20% 40%
Case-1	1	0% 20% 40%
Case-3	3	0% 20% 40%
Case-5	5	0% 20% 40%
Case-7	7	0% 20% 40%
Case-9	9	0% 20% 40%

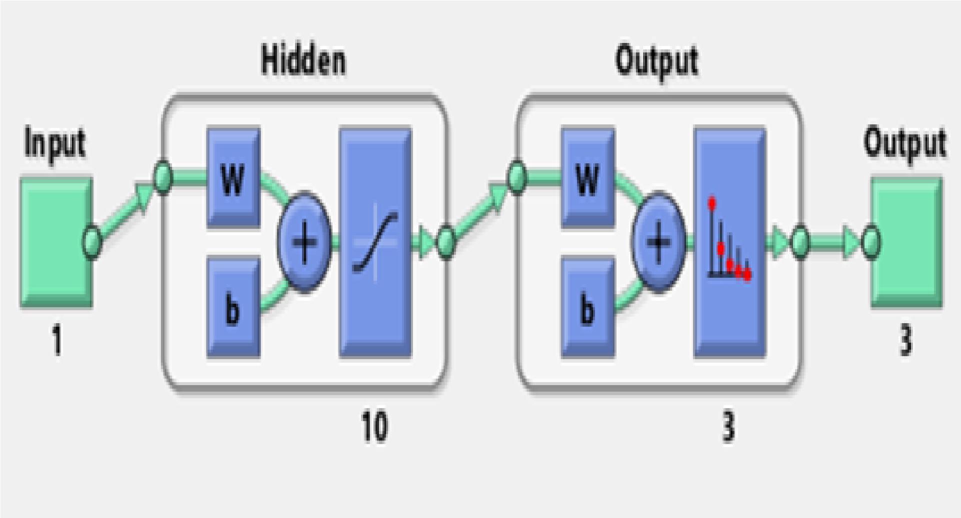

FIGURE 4.2: Architecture of ANN.

4.7 ANN Architecture

An ANN is trained with the configurations as (01-01-03), the hidden layer having 10 neurons. The tansig as a non-linear activation function is applied on each hidden and output layer. The trainlm,trainbr,traingd,traingdm,traingdx,trainscg, trainrp,trainbfg,trainoss,traincgf,traincgb,traincgp are used one by one as training algorithms to find out the fastest back-propagation algorithm. The network is trained in 1000 epochs, with a learning rate of 0.5. The architecture of ANN can be shown in fig 4.2.

TABLE 4.3: Fixed Parameters of ANN

Input Layers	**1**
Output Layers	**3**
Hidden Layers	**1**
No of Neurons	**10**
No of Epochs	**1000**
Learning Rate	**0.5**

4.8 Evaluation Metrices

To check the accuracy for results different performance matrices such as dice similarity coefficient (DSC), Mean Square Error (MSE), mean and standard deviation is calculated.

4.8.1 DSC

The dice similarity coefficient (DSC) is amongst the most widely used evaluation metrics in medical image segmentation. The DSC value is a straightforward and practical description of spatial overlap that may be used to improve image segmentation accuracy. A DSC has a value between 0 and 1, with 0 indicating no spatial overlap and 1 representing the total overlap between two sets of binary segmentation results. The DSC is defined as a measurement of the spatial overlap across two segmentations, A and B target regions.

$$DSC = (A, B) = 2\left(A \cap B\right)/(A + B) \tag{4.2}$$

Where A is the specific region in segmented output and B is the same region in GT image, ∩ is the intersection of a specific region of A and B.

4.8.2 MSE

Mean Squared Error (MSE) can be defined as the average of the squares of the errors obtained by subtracting the segmented output and GT image values. It can be defined as the average square of the errors obtained by subtracting the segmented output and GT image values. It can be expressed as

$$MSE = \frac{1}{mn} \sum_{i=0}^{m} \sum_{j=0}^{n} \left[R\left(i, j\right) - S\left(i, j\right)\right]^2 \tag{4.3}$$

Where, 'R(i,j)' represents the reference image and 'S(i,j)' represents the segmented image. 'm' represents the number of rows and 'n' represents the number of columns in segmented images and reference image. 'i= m' denotes the row-wise increment and 'j = n' denotes the column-wise increment in for loop. The MSE value for the resulting segmented image should be as lowest as possible. Reduced MSE values give a reduced occurrence of an error in segmented images.

4.8.3 STD

Standard deviation measures how far from the mean a given variable is. A lower value of STD means the process is more reliable.

$$STD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(F_i - \mu)^2} \tag{4.4}$$

Where 'F' is any variable having 'n' number of observations and the mean value of 'F' is represented 'μ'Where 'F' is any variable having 'n' number of observations and the mean value of 'F' is represented $\mu = \frac{1}{n}\sum_{i=1}^{n} F_i$.

Chapter 5

Results and Evaluation

This chapter takes into account the segmented results of brain MR images obtained by each step of the methodology. Such as pre-processing in which colored images are converted into grayscale images, dimensions, and skull stripping of the images performed. Then the soft tissues are provided to the clustering algorithm for labeling. After that, the labeled data is trained on ANN network and tested to perform the accurate segmentation of brain MRI into different types of soft tissues e.g., WM, GM, and CSF.

5.1 Segmentation of Images using Levenberg Marquardt

Segmentation of brain MR images is necessary to differentiate WM, GM, and CSF for studying anatomical structural changes and brain quantification. The segmentation results generated by ANN using the 'lm' algorithm for the test image can be shown in Figure 5.1.

The dice similarity and mean squared error for segmented parts such as WM, GM and CSF are presented in the table for the first clean image. In the second table

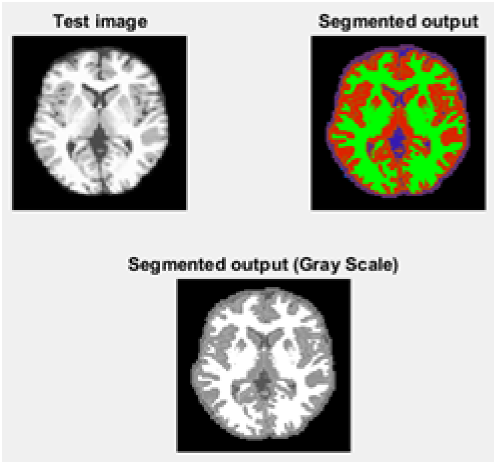

FIGURE 5.1: Segmentation Output

TABLE 5.1: Performance Evaluation

Performance Metrices	**Image**
Dice WM	0.9697
Dice GM	0.9635
Dice CSF	0.7308
MSE WM	0.0103
MSE GM	0.0251
MSE CSF	0.0671

TABLE 5.2: Performance Evaluation

	Mean		**Standard Deviation**	
Image	**GT**	**Segmented Region**	**GT**	**Segmented Region**
WM	0.2006	0.1987	0.4005	0.3991
GM	0.3044	0.2903	0.4602	0.4539
CSF	0.1546	0.1167	0.3615	0.3226

Mean and STD for the segmented ROI named as WM, GM, and CSF of the first test images are presented.

In Figure 5.2, the WM is mined and compared with its particular GT image. Figure 5.3, shows the bar graphs for GT-WM, Figure 5.4, show the bar graph for Segmented-WM, and Figure 5.5 shows the superimposed bar graph for GT-WM. It shows the segmentation difference between GT and segmented region.

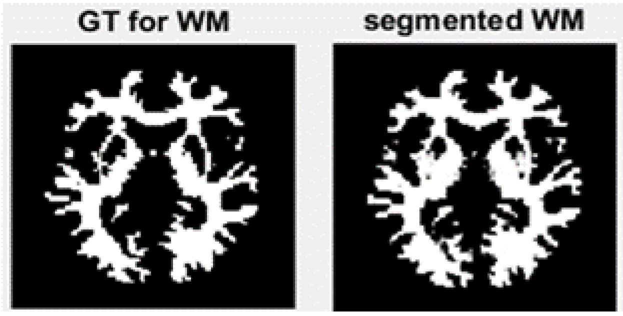

FIGURE 5.2: Segmented Image

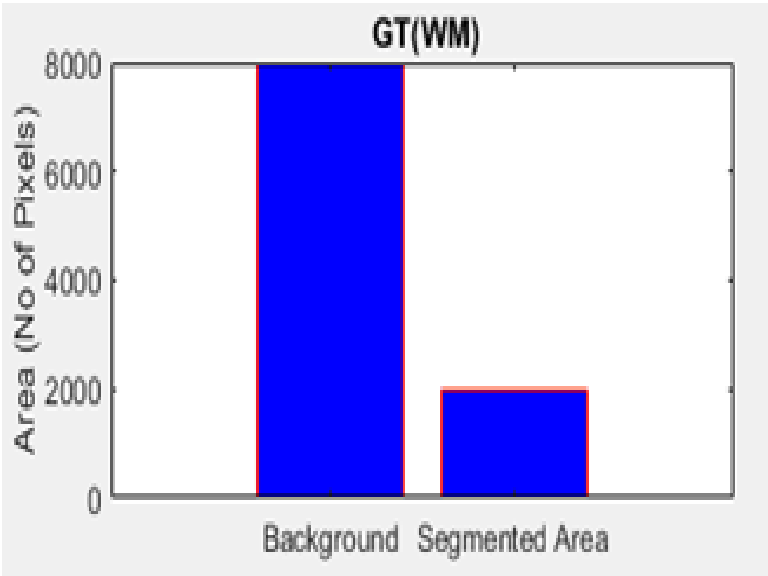

FIGURE 5.3: GT ROI

In Figure 5.6 GM is extracted and compared with its respective GT. Figure 5.7 shows the bar graphs for GT-GM, Figure 5.8, show the bar graph for Segmented-GM, and Figure 5.9 shows the superimposed bar graph for GT-GM.

In Figure 5.10 CSF is extracted and compared with its respective GT image. Figure 5.11 shows the bar graphs for GT-CSF, 5.12, show the bar graph for

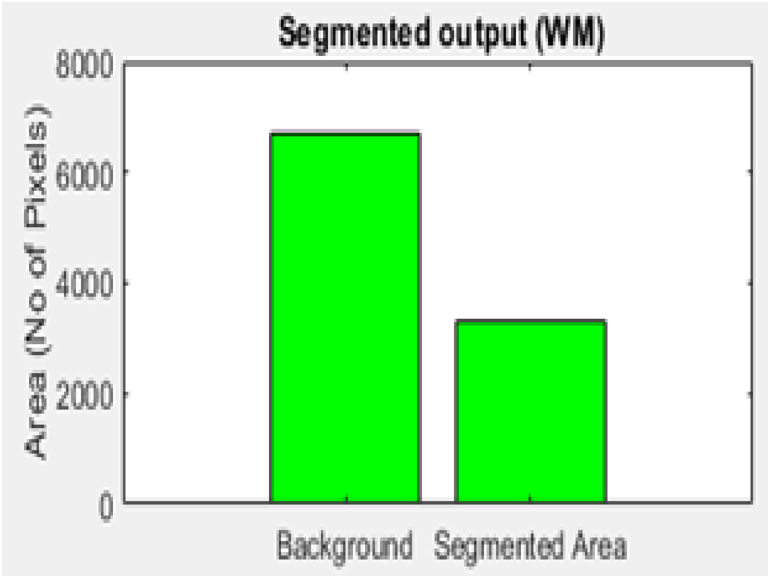

FIGURE 5.4: Segmented ROI

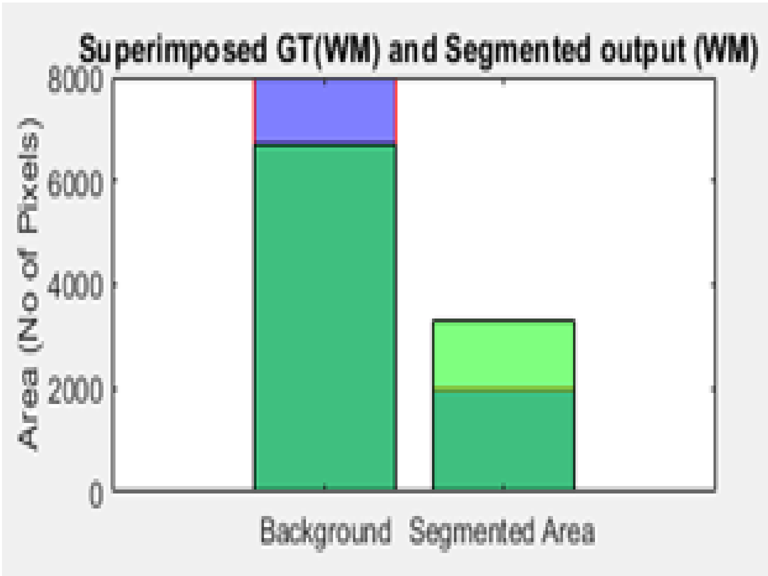

FIGURE 5.5: Superimposed Bar

Segmented-CSF, and Figure 5.13 shows the superimposed bar graph for GT-CSF and segmented CSF.

Figure 5.14 depicts the training algorithm's performance after several iterations. Table 5.3 displays the value of best validation performance, maximum reached epochs, and time was taken to reach maximum epochs.

Moreover in Figure 5.15, the probability Density Estimation of segmented images and their respective GT image is presented. it can be seen that each segmented

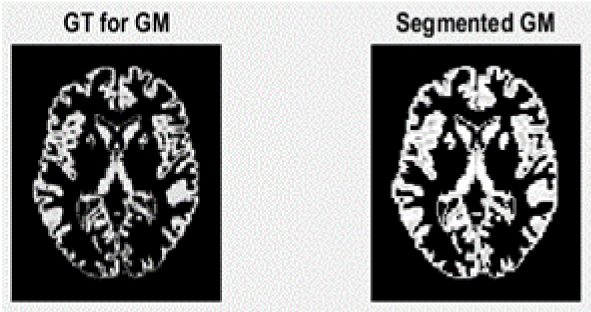

FIGURE 5.6: Segmented Image

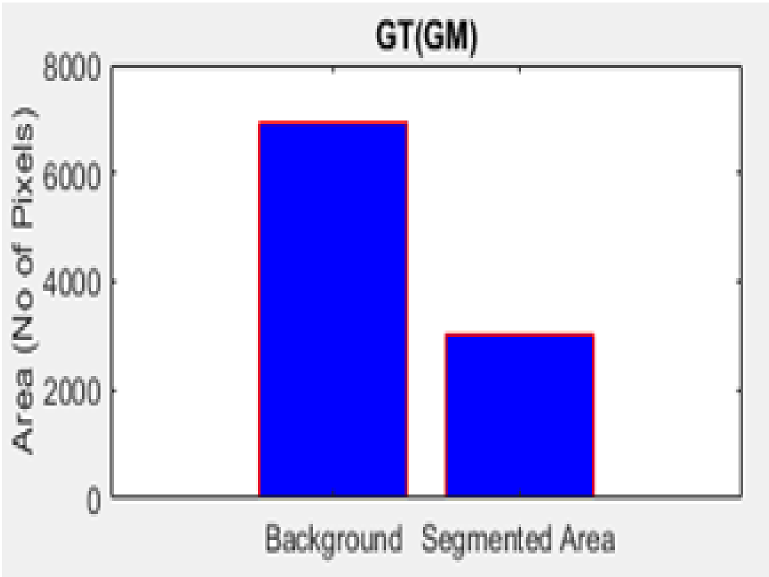

FIGURE 5.7: GT ROI

TABLE 5.3: Levenberg Marquardt Performance

Algorithm	**Levenberg-Marquardt**
Best validataion performance	0.1303
Maximum reached epochs	37
Time taken	32 s

output WM, GM, and CSF have almost the same data distribution concerning their GT image.

Figure 5.16 shows the standard deviation of segmented and non-segmented mages. Similarly Figure 5.17, 5.18 shows the graph of Mean and Dice of segmented and non-segmented images

The segmentation results generated by ANN for another test image can be shown in

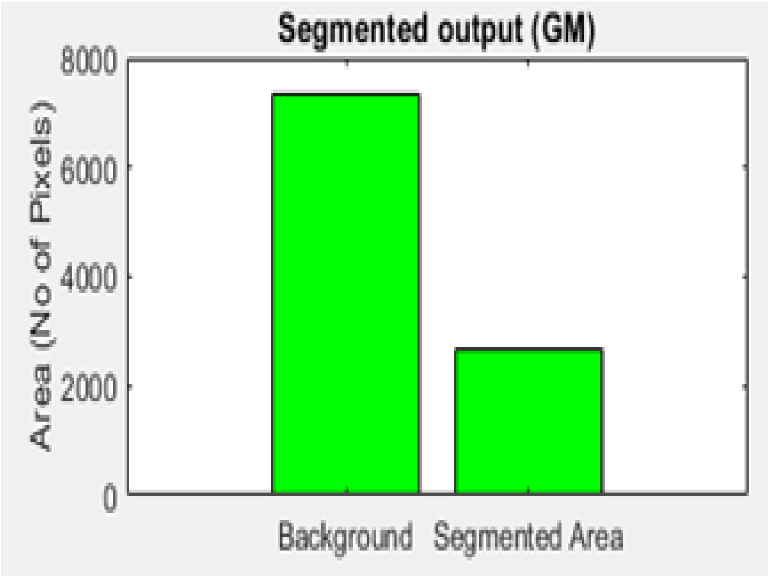

FIGURE 5.8: GT ROI

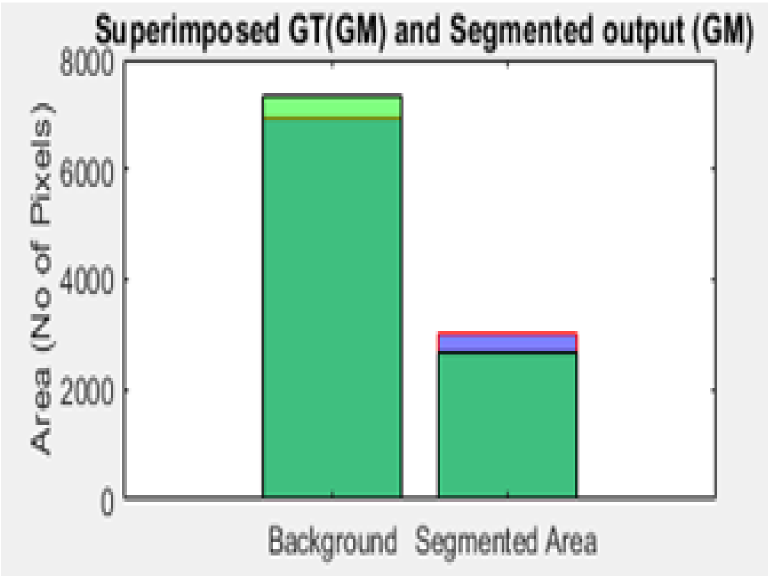

FIGURE 5.9: Superimposed Bars

the Figure 5.19. Table 5.4 provides the dice similarity and MSE for the segmented regions. Table 5.5 below Provides the Mean and STD for the segmented regions named as WM, GM, and CSF of the test image

The segmentation results generated by ANN for another test image can be shown in Figure 5.20. Table 5.6 provides the dice similarity and MSE for the segmented WM, GM, and CSF. Table 5.7 Provides the mean and STD for the segmented regions named as WM, GM, and CSF of the test images.

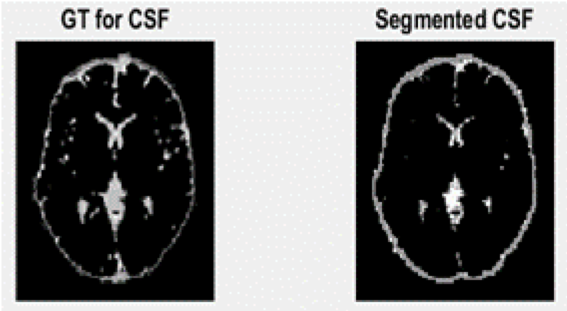

FIGURE 5.10: Segmented ROI

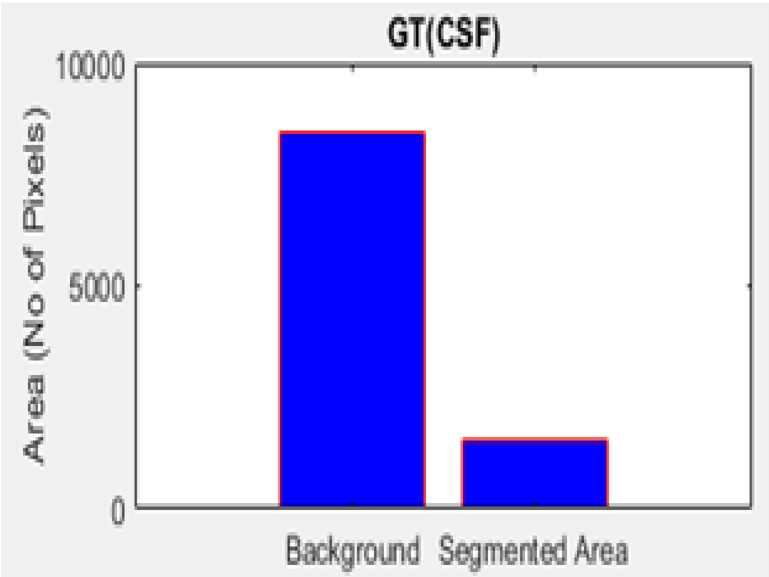

FIGURE 5.11: GT ROI

The segmentation results generated by ANN for another test image can be shown as in Figure 5.21. Table 5.8 provides the dice similarity and MSE for the segmented WM, GM, and CSF. Table 5.9.Provides the mean and STD for the segmented regions named as WM, GM, and CSF of a test image.

The segmentation results generated by ANN for another test image can be shown in Figure 5.22. Table 5.10 provides the dice similarity and MSE for the segmented WM, GM, and CSF. Table 5.11 provides the mean and STD for the segmented regions named as White Matter (WM), Grey Matter (GM), and Cerebrospinal Fluid (CSF) of the test image. Figure 5.23 represents the DSC and MSE recorded for White Matter (WM), Grey Matter (GM), and Cerebrospinal Fluid (CSF) for different clean images whereas the average DSC and MSE for WM, GM, and CSF are shown in Figure 5.24.

Table 5.4: Performance Evaluation

Performance Metrices	Image
Dice WM	0.9746
Dice GM	0.9838
Dice CSF	0.7324
MSE WM	0.0109
MSE GM	0.0069
MSE CSF	0.0511

Table 5.5: Performance Evaluation

	Mean		**Standard Deviation**	
Image	**GT**	**Segmented Region**	**GT**	**Segmented Region**
WM	0.2006	0.1983	0.4005	0.3998
GM	0.3044	0.2785	0.4602	0.4494
CSF	0.1546	0.0833	0.3615	0.3126

Table 5.6: Performance Evaluation

Performance Metrices	Image
Dice WM	0.9761
Dice GM	0.9510
Dice CSF	0.7297
MSE WM	0.0102
MSE GM	0.0241
MSE CSF	0.0512

Table 5.7: Performance Evaluation

	Mean		**Standard Deviation**	
Image	**GT**	**Segmented Region**	**GT**	**Segmented Region**
WM	0.2006	0.1983	0.4005	0.3998
GM	0.3044	0.2785	0.4602	0.4494
CSF	0.1546	0.0833	0.3615	0.3126

Table 5.8: Performance Evaluation

Performance Metrices	**Image**
Dice WM	0.9833
Dice GM	0.9809
Dice CSF	0.7270
MSE WM	0.0065
MSE GM	0.0084
MSE CSF	0.0507

Table 5.9: Performance Evaluation

	Mean		**Standard Deviation**	
Image	**GT**	**Segmented Region**	**GT**	**Segmented Region**
WM	0.2006	0.1993	0.4005	0.3991
GM	0.3044	0.2891	0.4602	0.4534
CSF	0.1546	0.1035	0.3615	0.2924

Table 5.10: Performance Evaluation

Performance Metrices	**Image**
Dice WM	0.9760
Dice GM	0.9553
Dice CSF	0.7227
MSE WM	0.0096
MSE GM	0.0214
MSE CSF	0.0516

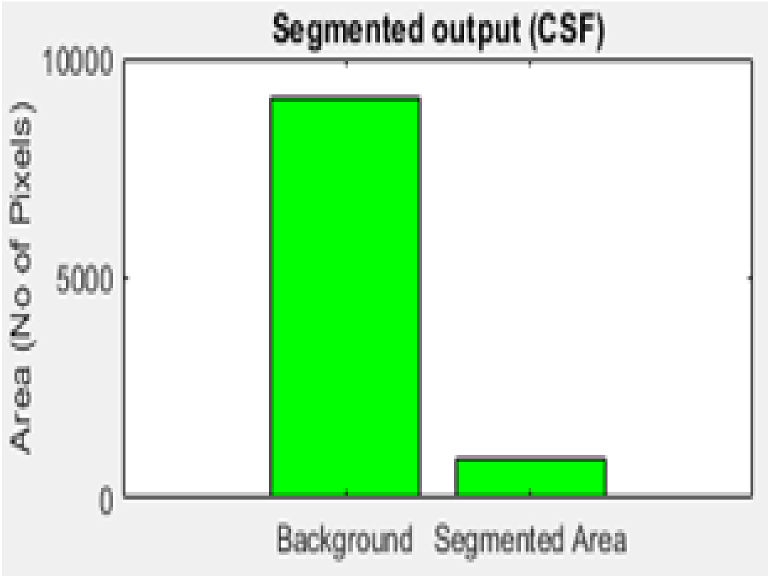

FIGURE 5.12: Segmented ROI

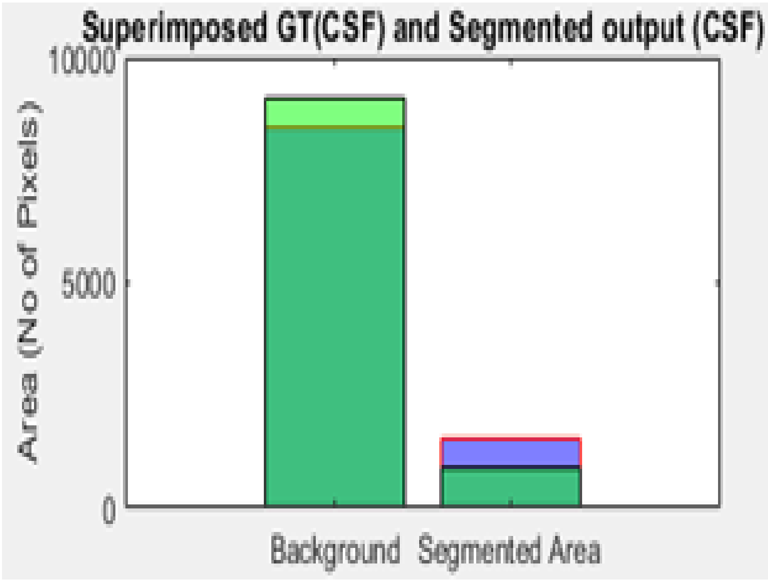

FIGURE 5.13: Segmented ROI

The performance evaluation of our segmentation outputs for various levels of noise and INU is also provided. The graphical representation is only provided for Case-0. Evaluation of other Cases is provided in tables and graphs. Test Case-0 contains images with 0% noise and 0%, 20% and 40% INU. Segmentation outputs for each of the images in Case-0 are provided in Figures 5.25, 5.26, and 5.27.

Figure 5.25 represents the segmentation output for images with 0% noise and 0% INU. The segmentation output for images with 0% noise and 20% INU is shown

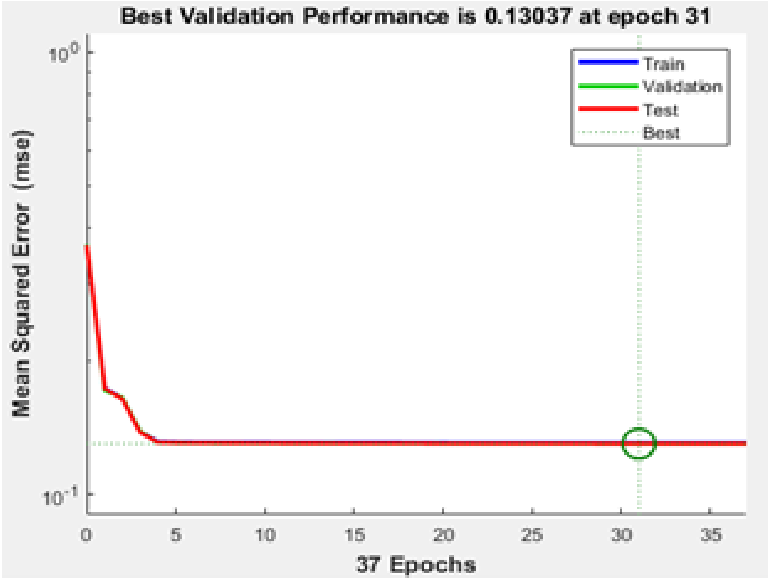

FIGURE 5.14: Performance of Levenberg Marquardt

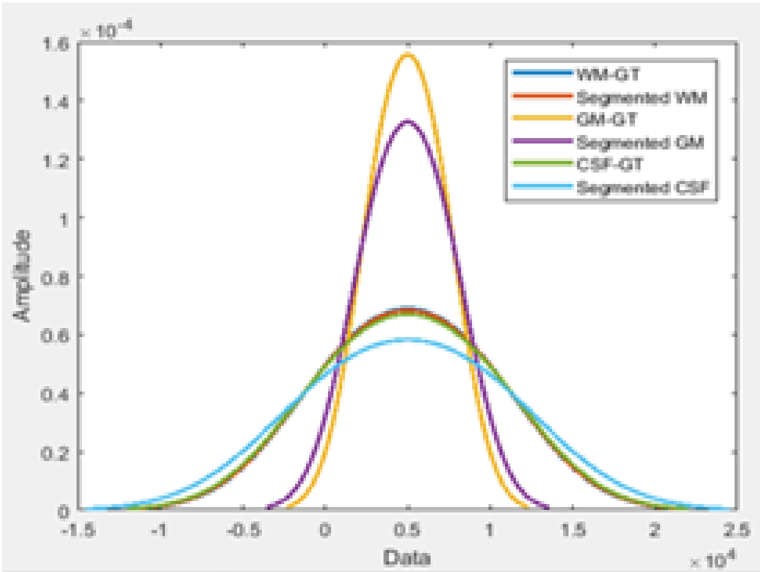

FIGURE 5.15: Probability Density Estimation

in Figure 5.26. Figure 5.27 illustrates the segmentation results for images with 0% noise and 40% INU.

Table 5.12 provides the results for Case-0 in terms of Dice and MSE for the segmented regions such as WM, GM, and CSF of test images. Table 5.13 provides the results for Case-0 in terms of Mean and STD for the segmented regions such as WM, GM, and CSF of test images.

Figure 5.28 shows the DSC and MSE recorded for WM, GM, and CSF of each

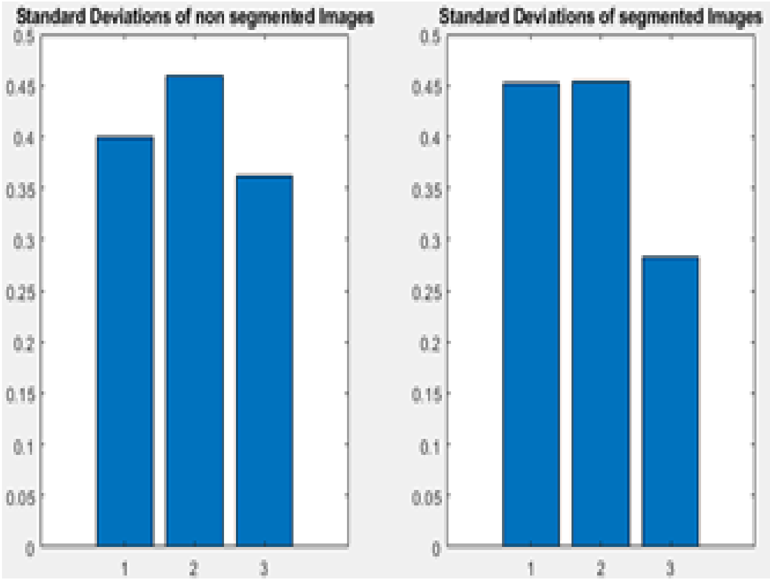

FIGURE 5.16: Standard Deviation

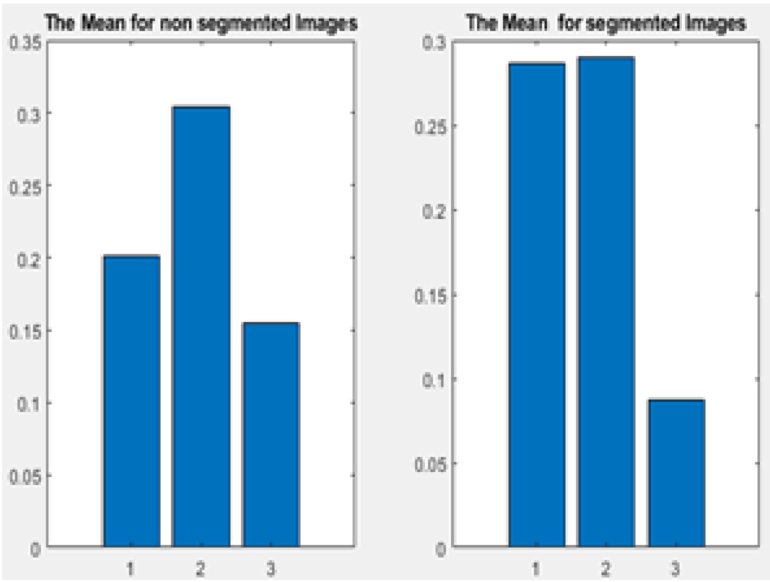

FIGURE 5.17: Mean for Segmented Image

TABLE 5.11: Performance Evaluation

	Mean		**Standard Deviation**	
Image	**GT**	**Segmented Region**	**GT**	**Segmented Region**
WM	0.2006	0.1985	0.4005	0.3997
GM	0.3044	0.2876	0.4602	0.4478
CSF	0.1546	0.0899	0.3615	0.3126

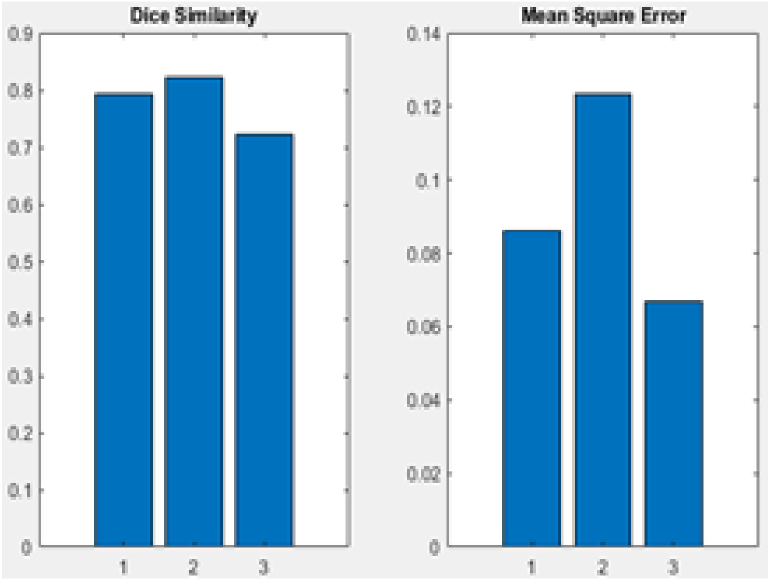

FIGURE 5.18: Dice Similarity for Segmented Image

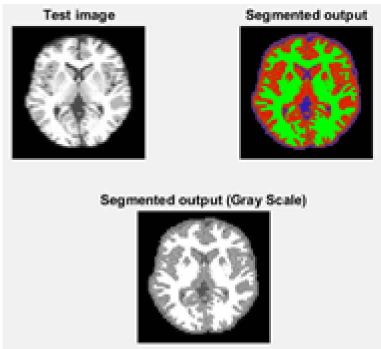

FIGURE 5.19: Segmentation Output

TABLE 5.12: Performance Evaluation for Case-0

Test Case-0			
Performance Metrices	**Image** 0%- 0%	**Image** 0%-20%	**Image** 0%-40%
Dice WM	0.9783	0.948	0.9122
Dice GM	0.9865	0.9721	0.9629
Dice CSF	0.7505	0.7419	0.7306
MSE WM	0.0127	0.0215	0.0372
MSE GM	0.014	0.0165	0.0218
MSE CSF	0.0681	0.0704	0.0725

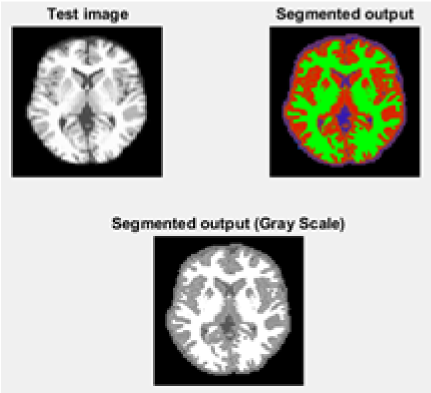

FIGURE 5.20: Segmentation Output

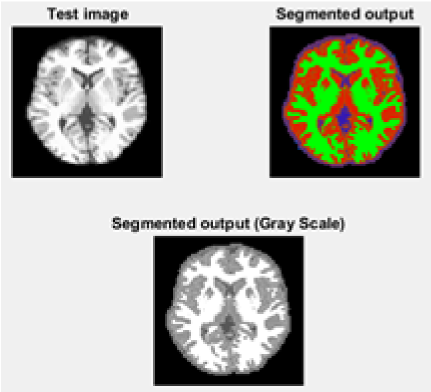

FIGURE 5.21: Segmentation Output

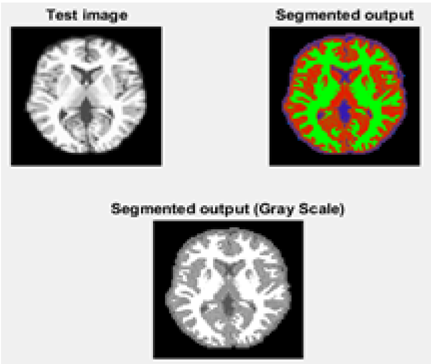

FIGURE 5.22: Segmented Image

image in Case-0 that contains 0% noise and 0%,20%,40% INU, whereas the average DSC and MSE of WM, GM and CSF for Case-0 is shown in Figure 5.29. In Figure 5.30 the data distribution of images with 0% noise and 40% INU is shown. It can be seen that each segmented output WM, GM, and CSF have almost similar data distribution concerning their GT.

Test Case-1 contains images with 1% noise and 0%, 20% and 40% INU. Table

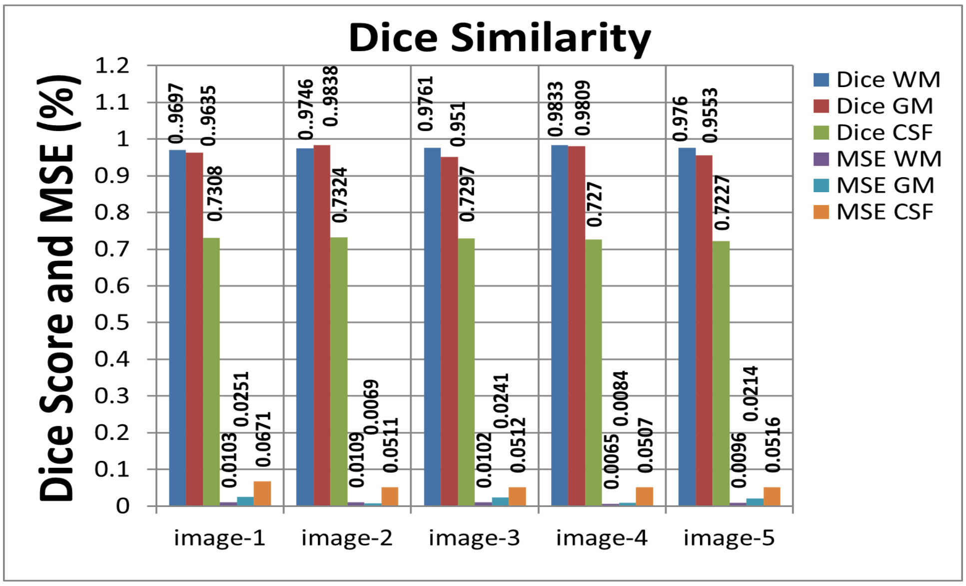

FIGURE 5.23: Dice Score and MSE for Clean Images

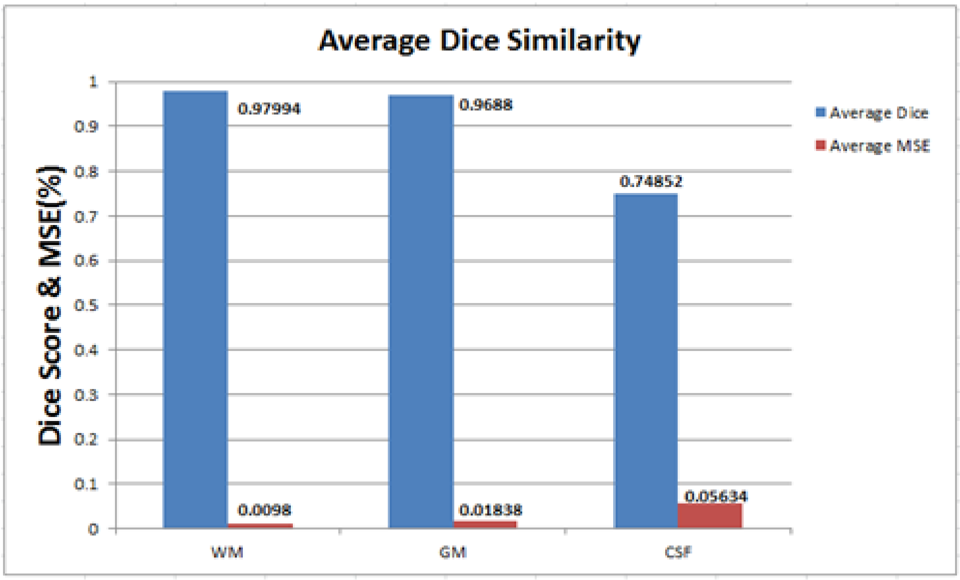

FIGURE 5.24: Average Dice Score and MSE for Clean Images.

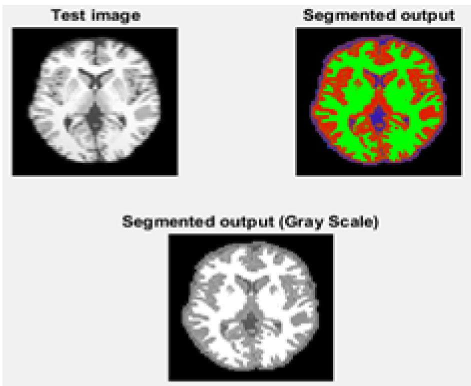

FIGURE 5.25: Segmented Output 0% Noise-0% INU

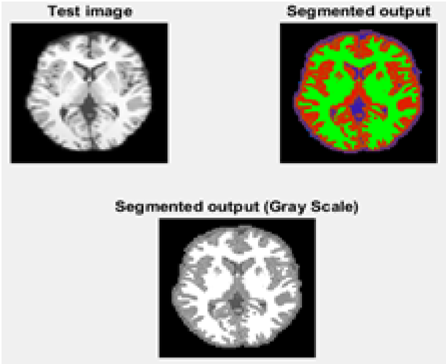

FIGURE 5.26: Segmented Output 0% Noise-20% INU

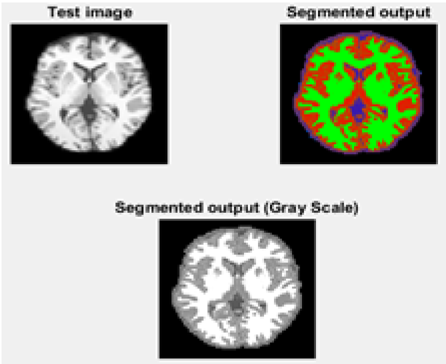

FIGURE 5.27: Segmented Output 0% Noise-40% INU

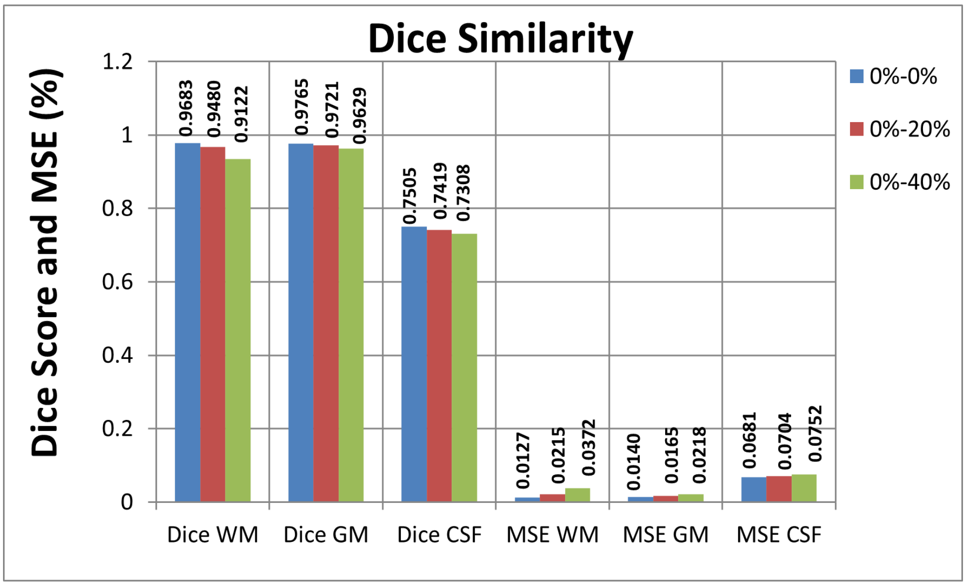

FIGURE 5.28: Dice Score and MSE for Case-0

TABLE 5.13: Performance Evaluation

		Mean		Standard Deviation	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.1998	0.4005	0.3993
Image 0% -0%	GM	0.3044	0.2994	0.4602	0.4586
	CSF	0.1546	0.117	0.3615	0.3256
	WM	0.2006	0.2896	0.4005	0.4501
Image 0% -20%	GM	0.3044	0.2803	0.4602	0.4539
	CSF	0.1546	0.1154	0.3615	0.3223
	WM	0.2006	0.2229	0.4005	0.4159
Image 0% -40%	GM	0.3044	0.3041	0.4602	0.4501
	CSF	0.1546	0.1144	0.3615	0.3116

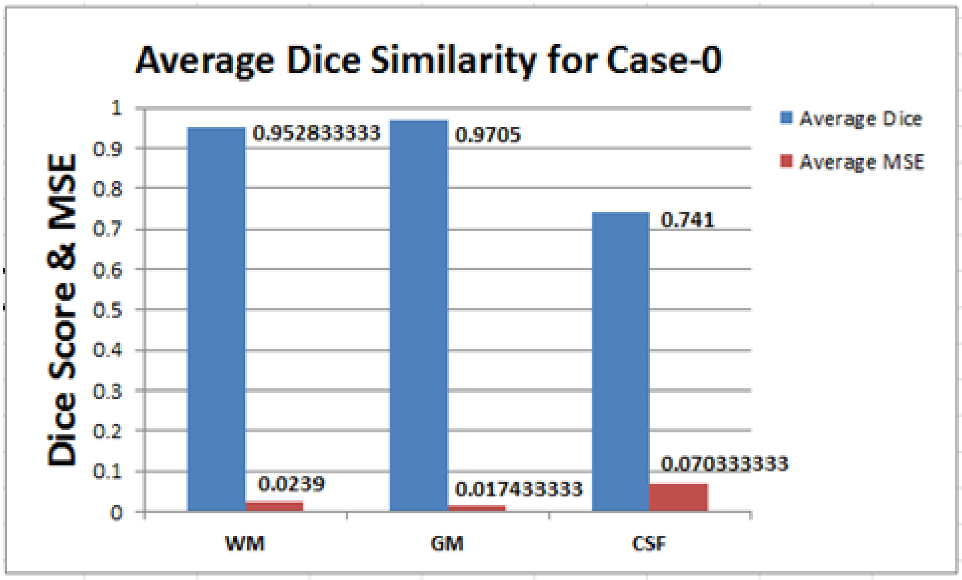

FIGURE 5.29: Average Dice Score and MSE for Case-0

TABLE 5.14: Performance Evaluation for Case-1

Test Case-1			
Performance Metrices	**Image** 1%-0%	**Image** 1%-20%	**Image** 1%-40%
Dice WM	0.9679	0.9474	0.9117
Dice GM	0.9759	0.9717	0.9625
Dice CSF	0.7493	0.7424	0.7497
MSE WM	0.0132	0.021	0.0364
MSE GM	0.0135	0.0164	0.0209
MSE CSF	0.0677	0.0704	0.0719

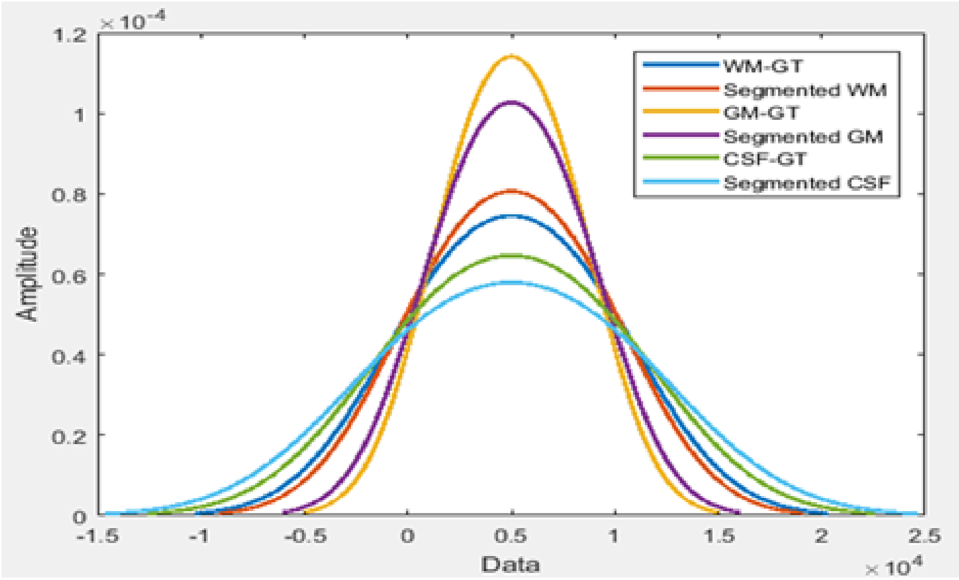

FIGURE 5.30: Probability Data Distribution for Case-0

5.14 provides the DSC and MSE and the mean and standard deviation for the segmented regions of images in test Case-1 are shown in Table 5.15.

Figure 5.31 shows the DSC and MSE recorded for WM, GM, and CSF of each

TABLE 5.15: Performance Evaluation for Case-1

		Mean		**Standard Deviation**	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.2004	0.4005	0.4004
Image 1% -0%	GM	0.3044	0.2894	0.4602	0.4578
	CSF	0.1546	0.119	0.3615	0.3249
	WM	0.2006	0.2196	0.4005	0.4093
Image 1% -20%	GM	0.3044	0.2801	0.4602	0.4531
	CSF	0.1546	0.115	0.3615	0.3203
	WM	0.2006	0.2224	0.4005	0.4151
Image 1% -40%	GM	0.3044	0.2995	0.4602	0.4499
	CSF	0.1546	0.1138	0.3615	0.3109

image in Case-1 that contains 1% noise and 0%,20%, and 40% INU, whereas the average DSC and MSE of WM, GM, and CSF for Case-1 are shown in Figure 5.32.

Test Case-3 contains images with 3% noise and 0%, 20% and 40% INU. Table 5.16 provides the DSC and MSE and Table 5.17 displays the standard deviation and mean for the segmented regions of images in test Case-3. Table 5.17 provides the results for Case-3 in terms of mean and STD for the Segmented regions such as WM, GM, and CSF of test images.

TABLE 5.16: Performance Evaluation for Case-3

Test Case-3			
Performance Metrices	**Image** 3%-0%	**Image** 3%-20%	**Image** 3%-40%
Dice WM	0.9654	0.946	0.9126
Dice GM	0.9769	0.9702	0.9617
Dice CSF	0.7412	0.7354	0.7411
MSE WM	0.0142	0.0218	0.036
MSE GM	0.0131	0.0169	0.0215
MSE CSF	0.0697	0.0722	0.0734

TABLE 5.17: Performance Evaluation for Case-3

		Mean		**Standard Deviation**	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.1996	0.4005	0.3989
Image 3% -0%	GM	0.3044	0.2904	0.4602	0.4581
	CSF	0.1546	0.1182	0.3615	0.3241
	WM	0.2006	0.2199	0.4005	0.4099
Image 3% -20%	GM	0.3044	0.2791	0.4602	0.4527
	CSF	0.1546	0.1156	0.3615	0.3209
	WM	0.2006	0.2238	0.4005	0.4159
Image 3% -40%	GM	0.3044	0.2973	0.4602	0.4495
	CSF	0.1546	0.1149	0.3615	0.3174

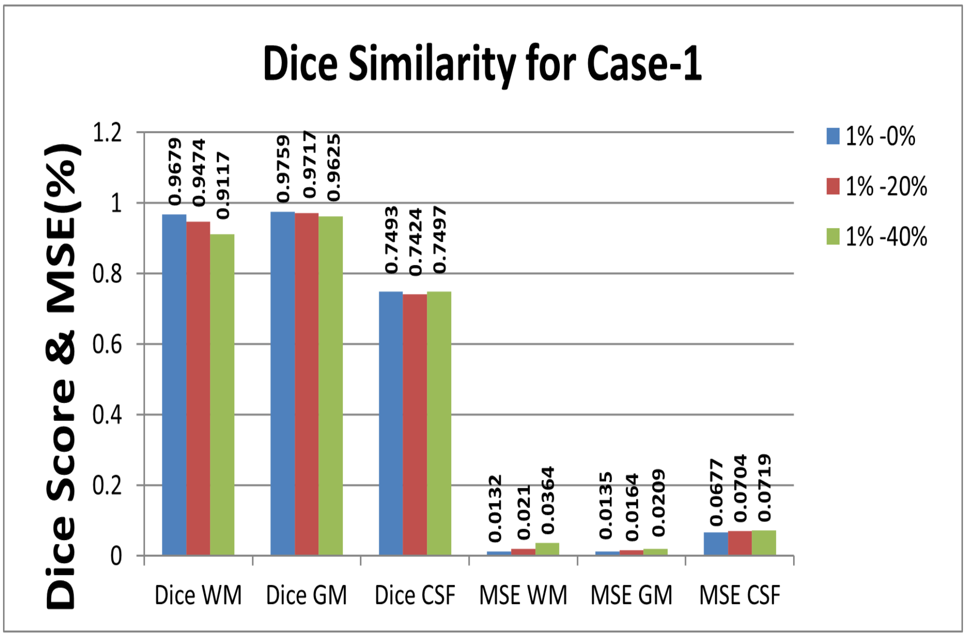

FIGURE 5.31: Dice Score and MSE for Case-1

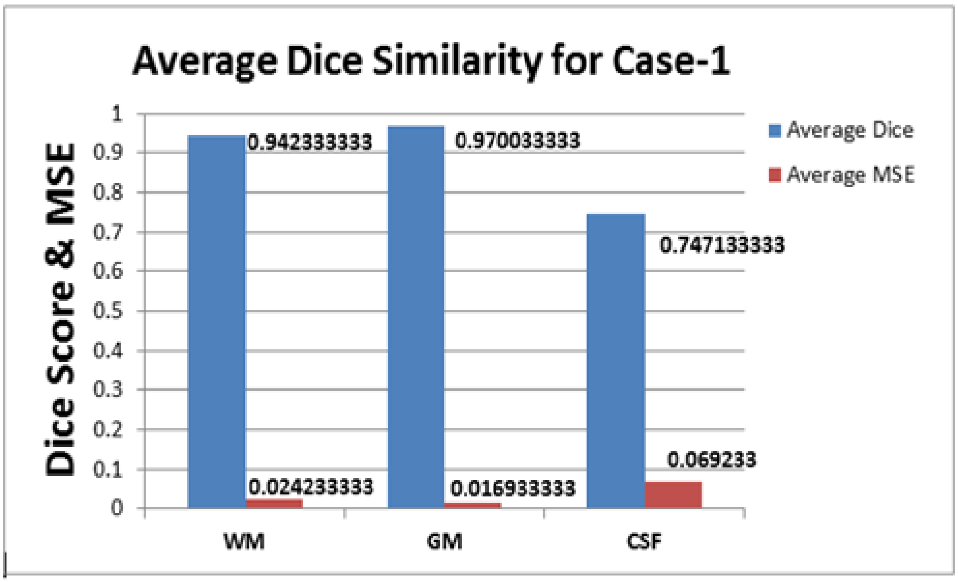

FIGURE 5.32: Average Dice Score and MSE for Case-1

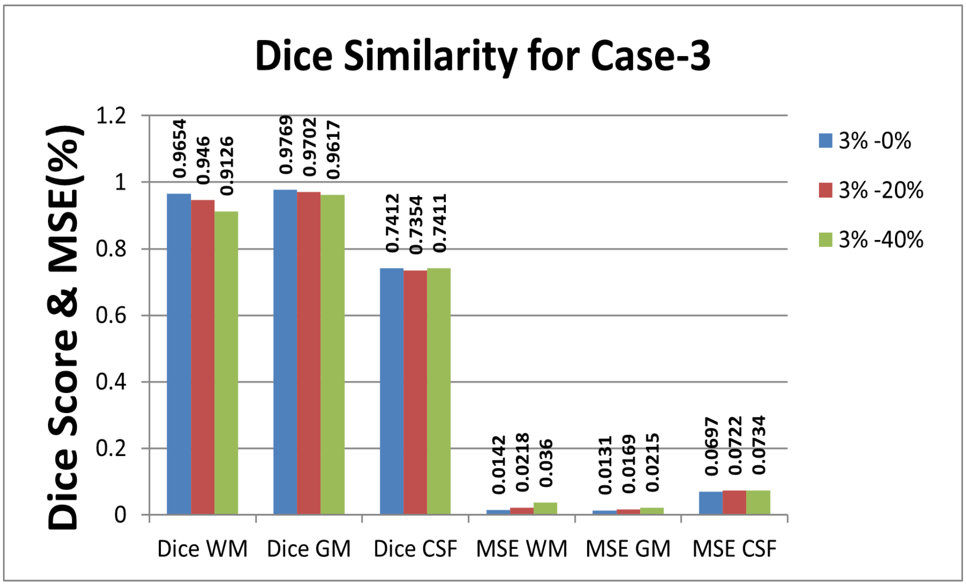

FIGURE 5.33: Dice Score and MSE for Case-3

image in Case-3 that contains 3% noise and 0%,20%, and 40% INU, whereas the average DSC and MSE of WM, GM, and CSF for Case-3 are shown in Figure 5.34. Test Case-5 contains images with 5% noise and 0%, 20% and 40% INU. Table 5.18 provides the dice similarity and MSE and Table 5.19 provides the mean and standard deviation for the segmented regions of images in test Case-5.

Figure 5.35 shows the DSC and MSE recorded for WM, GM, and CSF of each image in Case-5 that contains 5% noise and 0%,20%, and 40% INU, whereas the

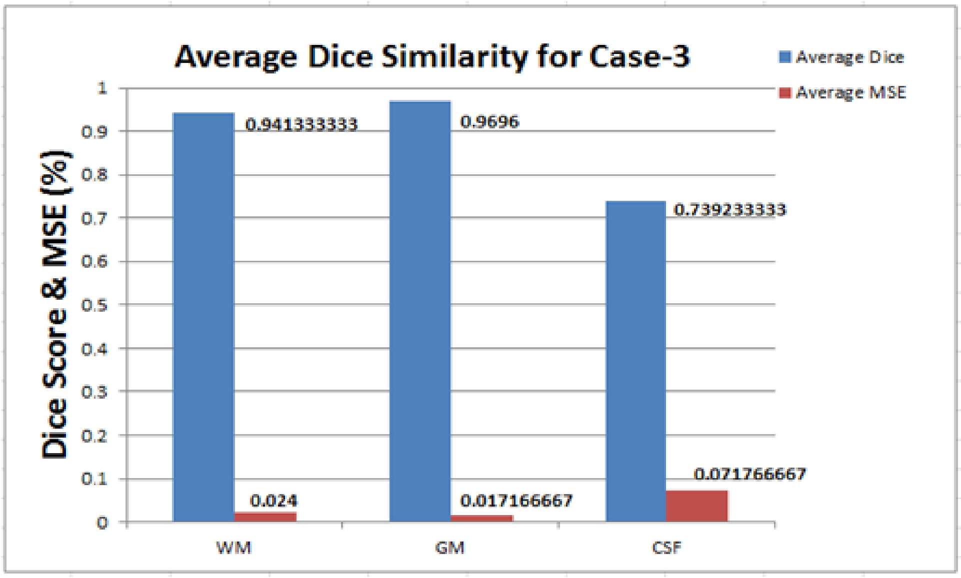

FIGURE 5.34: Average Dice Score and MSE for Case-3

average DSC and MSE of WM, GM, and CSF for Case-5 are shown in Figure 5.36. Test Case-7 contain images with 7% noise and 0%, 20% and 40% INU. Table 5.20 provides the dice similarity and MSE and Table 5.21 provides the mean and The standard deviation for the segmented regions of images in test Case-7. Figure 5.37 shows the DSC and MSE recorded for WM, GM, and CSF of each image in Case-7 that contains 7% noise and 0%,20%, and 40% INU, whereas the average DSC and MSE of WM, GM, and CSF for Case-7 are shown in Figure 5.38.

TABLE 5.18: Performance Evaluation for Case-5

	Test Case-5		
Performance Metrices	**Image** 5%-0%	**Image** 5%-20%	**Image** 5%-40%
Dice WM	0.9584	0.9342	0.9096
Dice GM	0.9776	0.972	0.9604
Dice CSF	0.7312	0.7272	0.7311
MSE WM	0.0175	0.0269	0.0387
MSE GM	0.0127	0.0176	0.0219
MSE CSF	0.0724	0.0732	0.0748

It is observed throughout the experimentation that, with the increasing noise levels such as 0%, 1%, 3%, 5%, and 7% combined with different INU levels like 0%, 20%, and 40%, there is a slight decrease in the performance of proposed scheme for brain

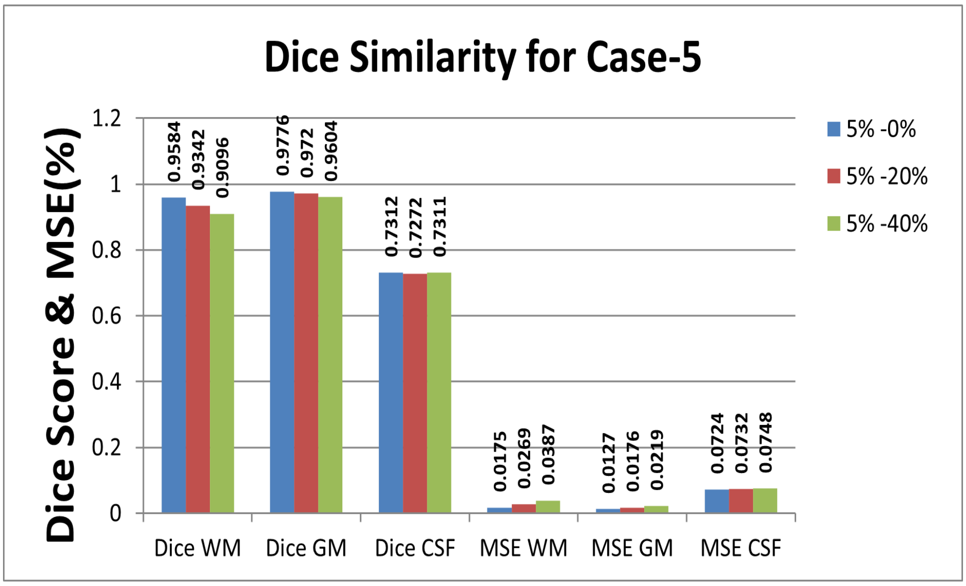

FIGURE 5.35: Dice Score and MSE for Case-5

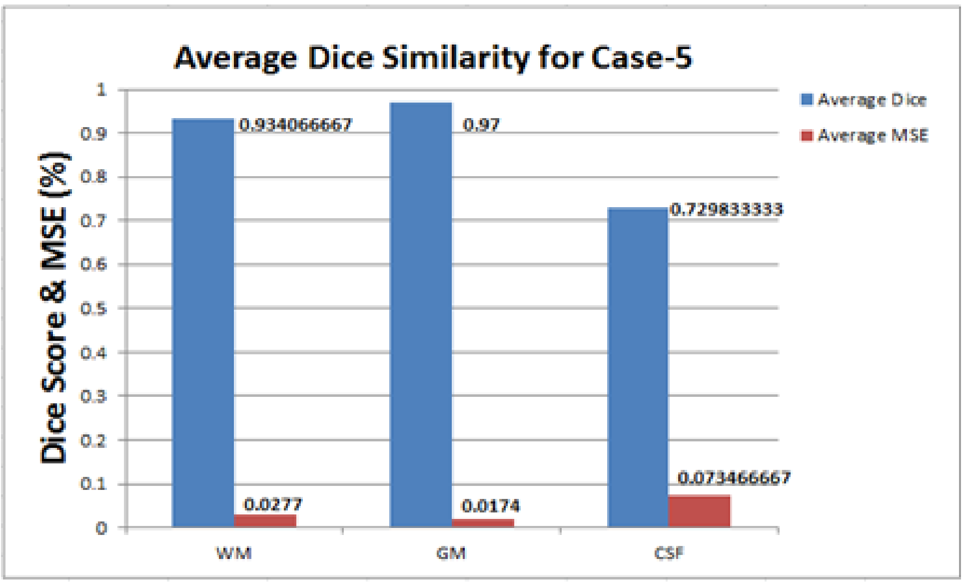

FIGURE 5.36: Average Dice Score and MSE for Case-5

TABLE 5.19: Performance Evaluation for Case-5

		Mean		**Standard Deviation**	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.2001	0.4005	0.3997
Image 5% -0%	GM	0.3044	0.2909	0.4602	0.4593
	CSF	0.1546	0.1142	0.3615	0.3201
	WM	0.2006	0.2201	0.4005	0.4179
Image 5% -20%	GM	0.3044	0.2797	0.4602	0.4532
	CSF	0.1546	0.1123	0.3615	0.3191
	WM	0.2006	0.2248	0.4005	0.4172
Image 5% -40%	GM	0.3044	0.2943	0.4602	0.4489
	CSF	0.1546	0.1139	0.3615	0.4274

TABLE 5.20: Performance Evaluation for Case-7

Performance Metrices	Test Case-7 Image 7%-0%	Image 7%-20%	Image 7%-40%
Dice WM	0.9576	0.9436	0.9117
Dice GM	0.9748	0.9689	0.9633
Dice CSF	0.7355	0.7203	0.7109
MSE WM	0.0217	0.0201	0.0367
MSE GM	0.0159	0.0188	0.0234
MSE CSF	0.0713	0.0757	0.0752

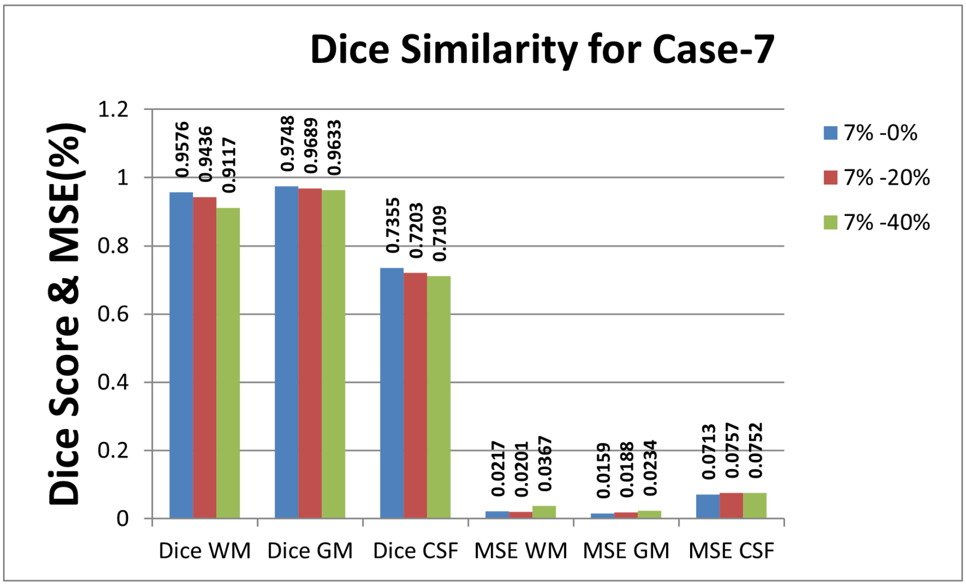

FIGURE 5.37: Dice Score and MSE for Case-7

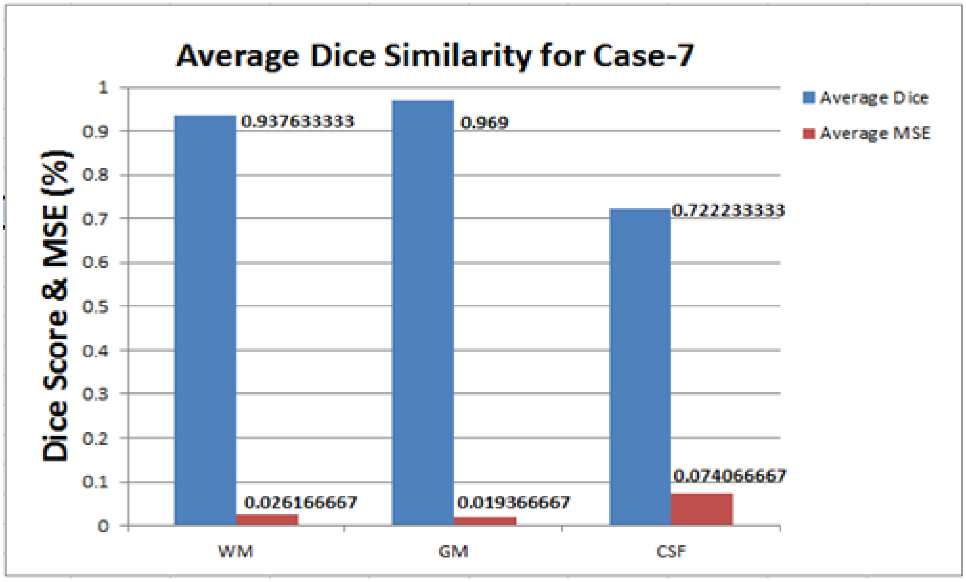

FIGURE 5.38: Average Dice Score and MSE for Case-7

TABLE 5.21: Performance Evaluation for Case-7

		Mean		**Standard Deviation**	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.2001	0.4005	0.4001
Image 7% -0%	GM	0.3044	0.2885	0.4602	0.4575
	CSF	0.1546	0.1191	0.3615	0.3232
	WM	0.2006	0.2172	0.4005	0.4053
Image 7% -20%	GM	0.3044	0.2991	0.4602	0.4519
	CSF	0.1546	0.1126	0.3615	0.3199
	WM	0.2006	0.223	0.4005	0.4159
Image 7% -40%	GM	0.3044	0.2979	0.4602	0.4592
	CSF	0.1546	0.1092	0.3615	0.3103

MRI segmentation. In Case-1, the average recorded DCS for WM is 0.9423, for GM is 0.97333 and for CSF it is 0.7471 whereas with the increase in noise intensity for every Case, finally for Case-7 the average recorded DCS for WM is 0.9376, for GM is 0.9690 and for CSF it is 0.7222. However, the overall performance achieved for all the Cases (Case-0 to Case-7) is quite better.

5.2 Segmentation of Images using Bayesian Regularization

Figure 5.39 depicts the performance graph of the training algorithm 'Br,' which displays the training algorithm's performance over multiple iterations. Table 5.22 displays the value of best validation performance, maximum reached epochs, and time were taken to reach maximum epochs.

The segmentation results generated by ANN using the 'br' algorithm for the test image are given below. Table 5.23 provides the dice similarity and MSE for the WM, GM, and CSF that have been segmented, and Table 5.24 provides the mean and Standard deviation for the WM, GM, and CSF segments with 0% noise and 0%, 20%, and 40% INU. Table 5.25 provides the dice similarity and MSE for the WM, GM, and CSF segments.

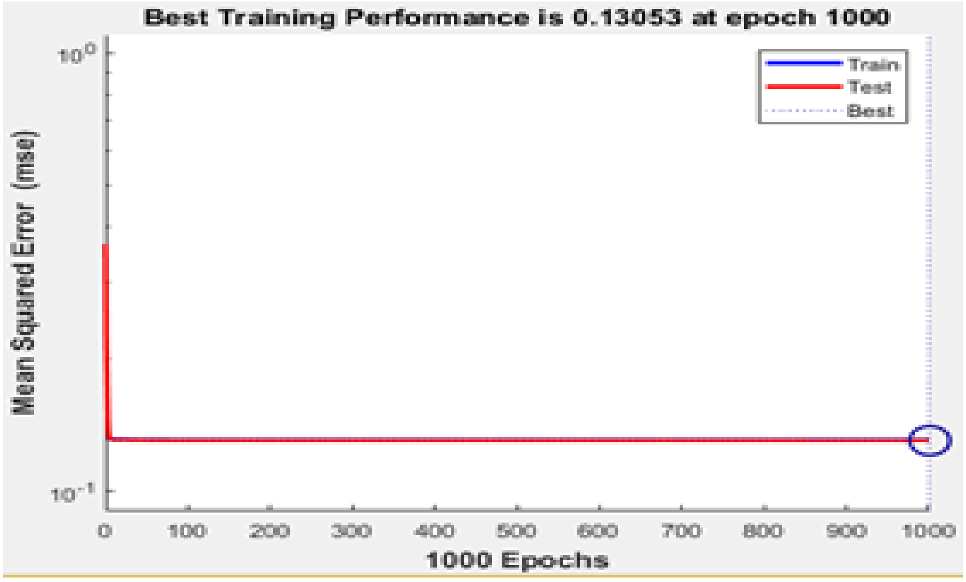

FIGURE 5.39: Performance of Bayesian Regularization Algorithm

TABLE 5.22: Bayesian Regularization Performance

Algorithm	**Bayesian-Regularization**
Best validation performance	0.13053
Maximum reached epochs	1000
Time taken	9:59 s

and Table 5.26 provides the mean and Standard deviation for WM, GM, and

TABLE 5.23: Performance Evaluation

Performance Metrices	**Image** 0%-0%	**Image** 0%-20%	**Image** 0%-40%
Dice WM	0.7923	0.7637	0.7392
Dice GM	0.8214	0.8303	0.8262
Dice CSF	0.7228	0.7053	0.7343
MSE WM	0.105	0.1241	0.1415
MSE GM	0.1088	0.1009	0.1017
MSE CSF	0.0671	0.0804	0.071

CSF, all are segmented with 9% noise and 0%, 20%, and 40% INU.

TABLE 5.24: Performance Evaluation

		Mean		Standard Deviation	
	ROI	GT	Segmented	GT	Segmented
Image 0% -0%	WM	0.2006	0.305	0.4005	0.4604
	GM	0.3044	0.3048	0.4602	0.4603
	CSF	0.1546	0.0857	0.3615	0.2826
	WM	0.2006	0.3245	0.4005	0.4682
Image 0% -20%	GM	0.3044	0.2901	0.4602	0.4538
	CSF	0.1546	0.1182	0.3615	0.3229
	WM	0.2006	0.3419	0.4005	0.4744
Image 0% -40%	GM	0.3044	0.2807	0.4602	0.4494
	CSF	0.1546	0.1126	0.3615	0.3161

TABLE 5.25: Performance Evaluation

Performance Metrices	**Image** 9%-0%	**Image** 9%-20%	**Image** 9%-40%
Dice WM	0.6198	0.6026	0.6023
Dice GM	0.7114	0.7145	0.7132
Dice CSF	0.7021	0.7225	0.7218
MSE WM	0.1948	0.2054	0.2096
MSE GM	0.1774	0.1766	0.1742
MSE CSF	0.0643	0.0651	0.0665

TABLE 5.26: Performance Evaluation

		Mean		Standard Deviation	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.2234	0.4005	0.4633
Image 9% -0%	GM	0.3044	0.3102	0.4602	0.4626
	CSF	0.1546	0.1167	0.3615	0.3176
	WM	0.2006	0.3162	0.4005	0.4651
Image 9% -20%	GM	0.3044	0.3142	0.4602	0.4642
	CSF	0.1546	0.1185	0.3615	0.3137
	WM	0.2006	0.3264	0.4005	0.4689
Image 9% -40%	GM	0.3044	0.3031	0.4602	0.4596
	CSF	0.1546	0.1183	0.3615	0.2826

5.3 Segmentation of Images using BFGS Quasi-Newton

Figure 5.40 shows the performance graph of the training algorithm'Bfg,' which depicts the training algorithm's performance across many iterations. Table 5.27 displays the value of best validation performance, maximum reached epochs, and time were taken to reach maximum epochs.

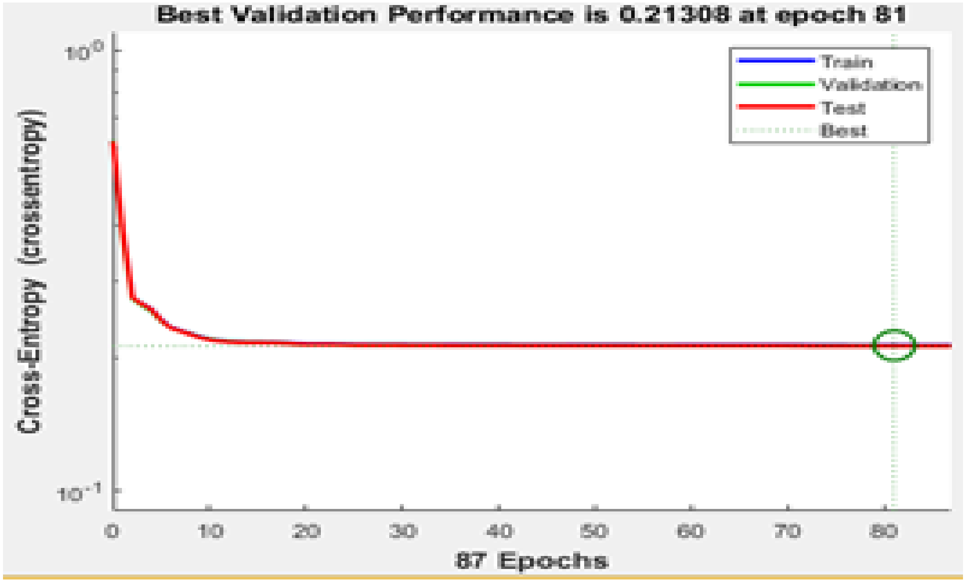

FIGURE 5.40: Performance of BFGS Quasi-Newton Algorithm

TABLE 5.27: BFGS Performance

Algorithm	**BFGS Quasi-Newton**
Best validation performance	0.21308
Maximum reached epochs	87
Time taken	41 s

TABLE 5.28: Performance Evaluation

Performance Metrices	**Image** 0%-0%	**Image** 0%-20%	**Image** 0%-40%
Dice WM	0.8261	0.7939	0.7668
Dice GM	0.7807	0.7833	0.7718
Dice CSF	0.7221	0.7125	0.7212
MSE WM	0.0843	0.1041	0.1219
MSE GM	0.1229	0.1186	0.1219
MSE CSF	0.064	0.0678	0.0683

The segmentation results generated by ANN using the ‘Bfg’ algorithm for the test image are given below. The dice similarity and MSE for the segmented WM, GM, and CSF are shown in Table 5.28, and the mean and standard deviation for

TABLE 5.29: Performance Evaluation

		Mean		Standard Deviation	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.2843	0.4005	0.4511
Image 0% -0%	GM	0.3044	0.2561	0.4602	0.4365
	CSF	0.1546	0.1059	0.3615	0.3276
	WM	0.2006	0.3045	0.4005	0.4602
Image 0% -20%	GM	0.3044	0.2428	0.4602	0.4288
	CSF	0.1546	0.1176	0.3615	0.3279
	WM	0.2006	0.3221	0.4005	0.4673
Image 0% -40%	GM	0.3044	0.2297	0.4602	0.4207
	CSF	0.1546	0.1143	0.3615	0.2827

TABLE 5.30: Performance Evaluation

Performance Metrices	**Image** 9%-0%	**Image** 9%-20%	**Image** 9%-40%
Dice WM	0.6324	0.6186	0.6022
Dice GM	0.6639	0.6655	0.6655
Dice CSF	0.7121	0.7138	0.7245
MSE WM	0.1838	0.1851	0.2087
MSE GM	0.1911	0.1186	0.185
MSE CSF	0.0643	0.0678	0.0693

segmented WM, GM, and CSF with 0% noise and 0%, 20%, and 40% INU are provided in Table 5.29.

TABLE 5.31: Performance Evaluation

		Mean		Standard Deviation	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.2994	0.4005	0.458
Image 9% -0%	GM	0.3044	0.2592	0.4602	0.4382
	CSF	0.1546	0.1156	0.3615	0.2826
	WM	0.2006	0.2847	0.4005	0.4513
Image 9% -20%	GM	0.3044	0.2669	0.4602	0.4424
	CSF	0.1546	0.1166	0.3615	0.3157
	WM	0.2006	0.3241	0.4005	0.4681
Image 9% -40%	GM	0.3044	0.2486	0.4602	0.4322
	CSF	0.1546	0.1159	0.3615	0.2829

Table 5.30 provides the dice similarity and MSE for the WM, GM, and CSF that have been segmented. And table 5.31 provides the mean and Standard deviation for the classified WM, GM, and CSF with 9% noise and 0%, 20%, and 40% INU.

5.4 Segmentation of Images using Congugate Gradient with Powell/Beale Restarts

Figure 5.41 demonstrates the performance graph of the learning algorithm 'cgb,' which depicts the training algorithm's efficiency as a result of the sum of epochs it has gone through. Table 5.32 displays the value of best validation performance, maximum reached epochs, and time were taken to reach maximum epochs

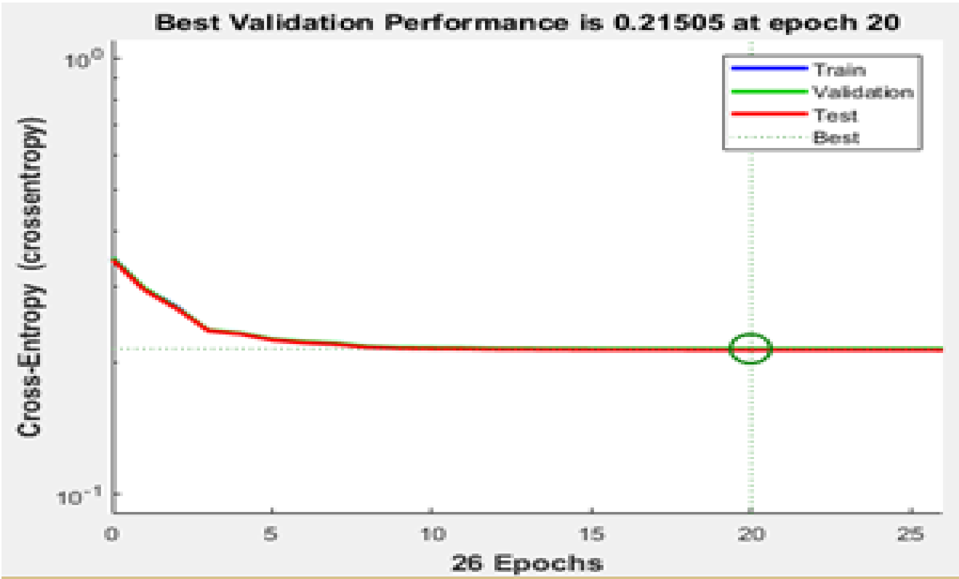

FIGURE 5.41: Performance Evaluation of Congugate Gradient with Powell-Beale Restarts

TABLE 5.32: Congugate Gradient with PowellBeale Restarts Performance

Algorithm	**CGB**
Best validation performance	0.21505
Maximum reached epochs	26
Time taken	12 s

The segmentation results generated by ANN using the 'cgb' algorithm for the test image are given below. The dice similarity and MSE for the segmented WM, GM, and CSF are shown in Table 5.33, and Table 5.34 provides the mean and Standard deviation for the segmented WM, GM, and CSF with 0% noise and 0%, 20%, and 40% INU. Table 5.35 provides the dice similarity and MSE for the WM, GM, and

Table 5.33: Performance Evaluation

Performance Metrices	**Image** 0%-0%	**Image** 0%-20%	**Image** 0%-40%
Dice WM	0.8248	0.8075	0.8208
Dice GM	0.8147	0.821	0.7683
Dice CSF	0.6245	0.6342	0.7143
MSE WM	0.0764	0.0956	0.1209
MSE GM	0.1141	0.1078	0.1064
MSE CSF	0.054	0.0579	0.0673

Table 5.34: Performance Evaluation

		Mean		**Standard Deviation**	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.2543	0.4005	0.3997
Image 0% -0%	GM	0.3044	0.3116	0.4602	0.4631
	CSF	0.1546	0.943	0.3615	0.3577
	WM	0.2006	0.296	0.4005	0.4565
Image 0% -20%	GM	0.3044	0.2978	0.4602	0.4573
	CSF	0.1546	0.1098	0.3615	0.309
	WM	0.2006	0.3211	0.4005	0.4669
Image 0% -40%	GM	0.3044	0.2894	0.4602	0.4535
	CSF	0.1546	0.1097	0.3615	0.2821

Table 5.35: Performance Evaluation

Performances Metrices	**Image** 9%-0%	**Image** 9%-20%	**Image** 9%-40%
Dice WM	0.6329	0.6242	0.6168
Dice GM	0.7078	0.7082	0.7083
Dice CSF	0.7109	0.7117	0.7132
MSE WM	0.1834	0.1807	0.1938
MSE GM	0.1807	0.1115	0.1787
MSE CSF	0.0632	0.0665	0.0618

CSF segments. And table 5.36 provides the mean and Standard deviation for the segmented WM, GM, and CSF with 9% noise and 0%, 20%, and 40% INU.

TABLE 5.36: Performance Evaluation

		Mean		Standard Deviation	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.299	0.4005	0.4578
Image 9% -0%	GM	0.3044	0.3141	0.4602	0.4642
	CSF	0.1546	0.1276	0.3615	0.2831
	WM	0.2006	0.2803	0.4005	0.4492
Image 9% -20%	GM	0.3044	0.3175	0.4602	0.4655
	CSF	0.1546	0.1138	0.3615	0.3197
	WM	0.2006	0.3052	0.4005	0.4605
Image 9% -40%	GM	0.3044	0.3083	0.4602	0.4618
	CSF	0.1546	0.1117	0.3615	0.2956

5.5 Segmentation of Images using Congugate Gradient with Fletcher powell

Figure 5.42 depicts the cgf training algorithm's graph, which demonstrates the cgf's performance over multiple iterations. Table 5.37 displays the value of best validation performance, maximum reached epochs, and time were taken to reach maximum epochs with cgf.

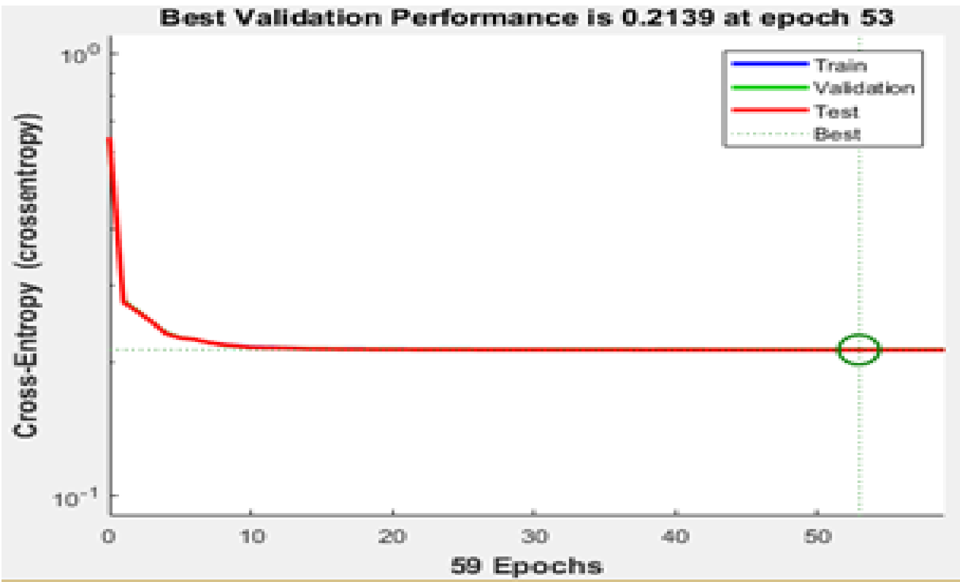

FIGURE 5.42: Performance of Congugate Gradient with Fletcher powell

The segmentation results generated by ANN using the 'cgf' algorithm for the test image are given below. Table 5.38 provides the dice similarity and MSE for WM, GM, and CSF segments and Table 5.39 provides the mean and Standard deviation

Table 5.37: Congugate Gradient with Fletcher Powell Performance

Algorithm	**CGF**
Best validation performance	0.2139
Maximum reached epochs	59
Time taken	24 s

Table 5.38: Performance Evaluation

Performances Metrices	**Image 0%-0%**	**Image 7%-20%**	**Image 9%-40%**
Dice WM	0.8091	0.7767	0.6014
Dice GM	0.7797	0.768	0.636
Dice CSF	0.7152	0.7187	0.7228
MSE WM	0.0945	0.1151	0.2084
MSE GM	0.1189	0.1213	0.1908
MSE CSF	0.0643	0.0665	0.0677

Table 5.39: Performance Evaluation

		Mean		**Standard Deviation**	
	ROI	GT	Segmented	GT	Segmented
Image 0% -0%	WM	0.2006	0.2945	0.4005	0.4558
	GM	0.3044	0.2353	0.4602	0.4242
	CSF	0.1546	0.1265	0.3615	0.2821
Image 7% -20%	WM	0.2006	0.3149	0.4005	0.4645
	GM	0.3044	0.2185	0.4602	0.4132
	CSF	0.1546	0.1187	0.3615	0.3256
Image 9% -40%	WM	0.2006	0.3222	0.4005	0.4673
	GM	0.3044	0.2198	0.4602	0.4141
	CSF	0.1546	0.1142	0.3615	0.2942

for the segmented white matter, grey matter, and cerebrospinal fluid with 0% noise 0% INU,7% noise 20%, and 9% noise 40% INU.

5.6 Segmentation of Images using Congugate Gradient with Polak-Ribiere

Figure 5.43 presents the efficiency graph of the learning algorithm'cgp' which indicates the performance of the training algorithm with numerous iterations it has passed through. Table 5.40 displays the value of best validation performance, maximum reached epochs, and time were taken to reach maximum epochs with cgp.

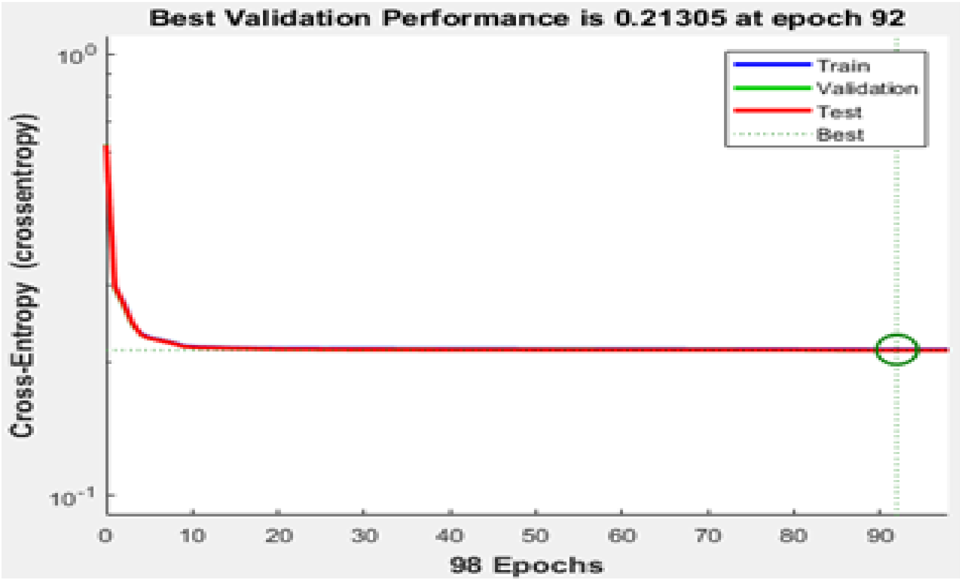

FIGURE 5.43: Performance of Congugate Gradient with Polak-Ribiere

TABLE 5.40: Congugate Gradient with Polak-Ribiere Performance

Algorithm	**CGP**
Best validation performance	0.21305
Maximum reached epochs	98
Time taken	45s

TABLE 5.41: Performance Evaluation

Performances Metrices	**Image** 0%-20%	**Image** 3%-20%	**Image** 5%-20%
Dice WM	0.8149	0.8104	0.8034
Dice GM	0.8202	0.816	0.8261
Dice CSF	0.7252	0.7145	0.7224
MSE WM	0.0911	0.0938	0.0981
MSE GM	0.107	0.1093	0.1037
MSE CSF	0.0649	0.0658	0.067

The segmentation results generated by ANN using the 'cgp' algorithm for the test image are given below. Table 5.41 depicts the dice similarity and MSE for

Table 5.42: Performance Evaluation

	Mean			Standard Deviation	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.2915	0.4005	0.4545
Image 0% -20%	GM	0.3044	0.2906	0.4602	0.4541
	CSF	0.1546	0.1135	0.3615	0.3228
	WM	0.2006	0.2942	0.4005	0.4557
Image 3% -20%	GM	0.3044	0.2897	0.4602	0.4536
	CSF	0.1546	0.1153	0.3615	0.3241
	WM	0.2006	0.2985	0.4005	0.4576
Image 5% -20%	GM	0.3044	0.2919	0.4602	0.4547
	CSF	0.1546	0.1157	0.3615	0.2964

the segregated WM, GM, and CSF, whereas Table 5.42 indicates the mean and Standard deviation for the segment WM, GM, and CSF with 0% noise 20% INU, 3% noise 20% INU, and 5% noise 20% INU.

5.7 Segmentation of Images using Gradient Descent Back Propogation

Figure 5.44 display the 'gd' training algorithm's evaluation graph which shows the achievement of the training algorithm with the many iterations it has departed and the total time that it consumes. Table 5.43 displays the value of best validation performance, maximum reached epochs, and time were taken to reach maximum epochs with gd.

Table 5.43: Gradient Descent Back Propagation Performance

Algorithm	**GD**
Best validation performance	0.26834
Maximum reached epochs	1000
Time taken	1:53s

The segmentation results generated by ANN using the 'gd' algorithm for the test image are given below. Table 5.44 provides the dice similarity and MSE for the segmentation of white matter, grey matter, CSF, and Table 5.45 provides the

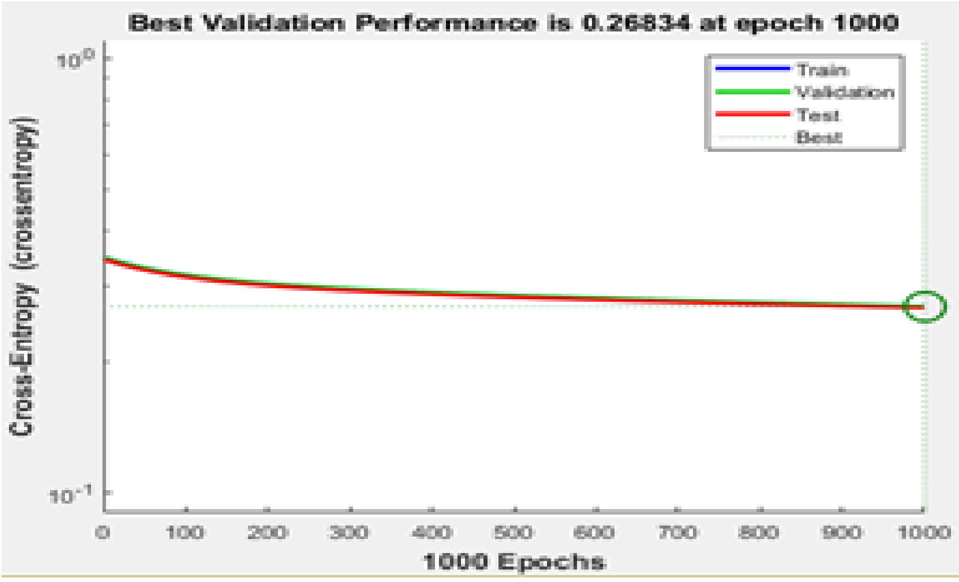

FIGURE 5.44: Performance of Gradient Descent Back Propogation

TABLE 5.44: Performance Evaluation

Performances Metrices	**Image** 0%-40%	**Image** 1%-40%	**Image** 7%-40%
Dice WM	0.6149	0.6054	0.541
Dice GM	0.6469	0.6451	0.6418
Dice CSF	0.7143	0.7138	0.7059
MSE WM	0.219	0.2188	0.2047
MSE GM	0.3047	0.304	0.3041
MSE CSF	0.0657	0.0615	0.0603

mean and Standard deviation for the WM, GM, and CSF segments with 0% noise 40% INU,1% noise 40% INU, and 7% noise 40% INU.

5.8 Segmentation of Images using Gradient Descent with Momentum

Figure 5.45 shows the effectiveness of the training algorithm 'gdm,' which depicts the learning method's performance. With several iterations and the total time that it consumes. Table 5.46 displays the value of best validation performance, maximum reached epochs, and time were taken to reach maximum epochs with gdm.

Table 5.45: Performance Evaluation

	Mean			Standard Deviation	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.4196	0.4005	0.4935
Image 0% -40%	GM	0.3044	0.3856	0.4602	0.4856
	CSF	0.1546	0.1195	0.3615	0.3487
	WM	0.2006	0.4194	0.4005	0.4935
Image 1% -40%	GM	0.3044	0.3287	0.4602	0.4746
	CSF	0.1546	0.1343	0.3615	0.3791
	WM	0.2006	0.4253	0.4005	0.4944
Image 7% -40%	GM	0.3044	0.4319	0.4602	0.4847
	CSF	0.1546	0.1196	0.3615	0.3509

Table 5.46: Gradient Descent with Momentum Performance

Algorithm	**GDM**
Best validation performance	0.26301
Maximum reached epochs	1000
Time taken	1:56s

Table 5.47: Performance Evaluation

Performances Metrices	**Image** 0%-0%	**Image** 3%-0%	**Image** 9%-0%
Dice WM	0.6646	0.6646	0.6046
Dice GM	0.6689	0.6598	0.6568
Dice CSF	0.7253	0.7256	0.7219
MSE WM	0.2025	0.2025	0.2301
MSE GM	0.3044	0.3044	0.3044
MSE CSF	0.0676	0.0673	0.0659

Table 5.48: Performance Evaluation

	Mean			Standard Deviation	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.4049	0.4005	0.4909
Image 0% -0%	GM	0.3044	0.3516	0.4602	0.4839
	CSF	0.1546	0.1166	0.3615	0.3439
	WM	0.2006	0.4031	0.4005	0.4905
Image 3% -0%	GM	0.3044	0.3293	0.4602	0.4736
	CSF	0.1546	0.1359	0.3615	0.3751
	WM	0.2006	0.3813	0.4005	0.4857
Image 9% -0%	GM	0.3044	0.4331	0.4602	0.4791
	CSF	0.1546	0.1476	0.3615	0.3521

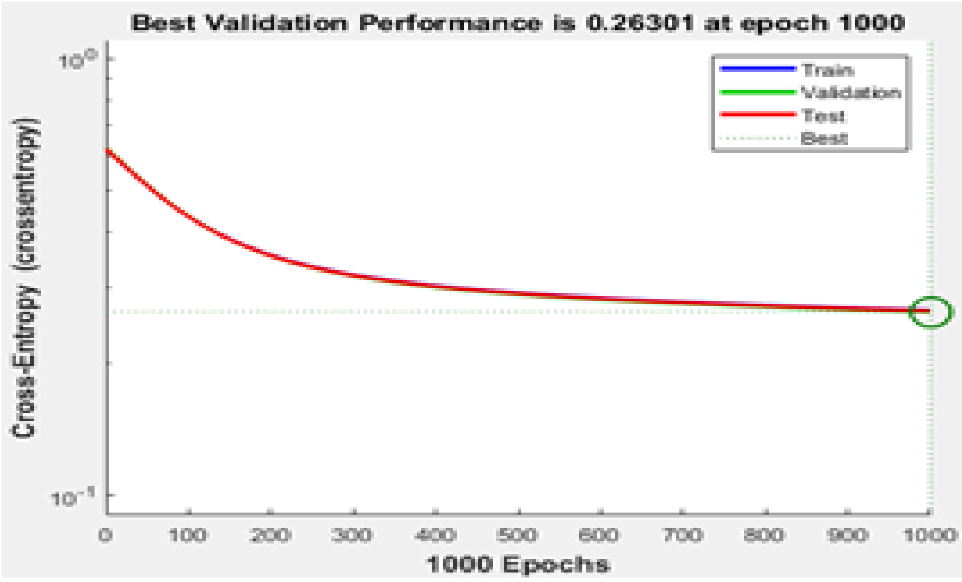

FIGURE 5.45: Performance of Gradient Descent with Momentum

The segmentation results generated by ANN using the 'gdm' algorithm for the test image are given below. Table 5.47 provides the dice similarity and MSE for the WM, GM, and CSF, and Table 5.48 provides the mean and Standard deviation for the segmented WM, GM, and CSF with 0% noise 0% INU,3% noise 0% INU, and 9% noise 0% INU.

5.9 Segmentation of Images using Gradient Descent with Variable Learning Rate

Figure 5.46 displays the evaluation of the training algorithm'gdx' which depicts the training algorithm's performance after multiple iterations and the total time that it consumes. Table 5.49 displays the value of best validation performance, maximum reached epochs, and time were taken to reach maximum epochs with gdm.

The segmentation results generated by ANN using the 'gdx' algorithm for the test image are given below. Table 5.50 provides the dice similarity and MSE for the WM, GM, and CSF, and Table 5.51 provides the mean and Standard deviation

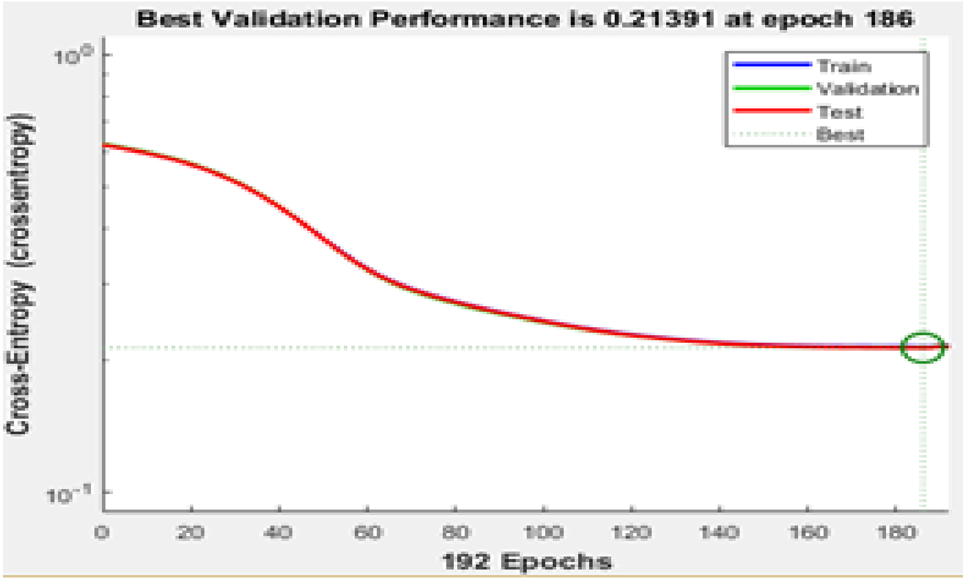

FIGURE 5.46: Performance Evaluation of Gradient Descent with Variable Learning Rate

TABLE 5.49: Gradient Descent with Variable Learning Rate Performance

Algorithm	**GDX**
Best validation performance	0.21391
Maximum reached epochs	192
Time taken	23s

TABLE 5.50: Performance Evaluation

Performance Metrices	**Image** 1%-20%	**Image** 5%-20%	**Image** 7%-20%
Dice WM	0.803	0.796	0.7942
Dice GM	0.7816	0.7764	0.7785
Dice CSF	0.7193	0.7218	0.7228
MSE WM	0.0984	0.1028	0.1038
MSE GM	0.1162	0.1181	0.1182
MSE CSF	0.0652	0.0653	0.0675

for the WM, GM, and CSF segments with 1% noise 20% INU,5% noise 20% INU, and 7% noise 20% INU.

Table 5.51: Performance Evaluation

		Mean		Standard Deviation	
	ROI	GT	Segmented	GT	Segmented
Image 1% -20%	WM	0.2006	0.2988	0.4005	0.4578
	GM	0.3044	0.2276	0.4602	0.4193
	CSF	0.1546	0.1139	0.3615	0.3426
	WM	0.2006	0.3032	0.4005	0.4597
Image 5% -20%	GM	0.3044	0.2237	0.4602	0.4167
	CSF	0.1546	0.1365	0.3615	0.3541
	WM	0.2006	0.3038	0.4005	0.4599
Image 7% -20%	GM	0.3044	0.2292	0.4602	0.4203
	CSF	0.1546	0.1376	0.3615	0.3526

5.10 Segmentation of Images using One Step Secant Back Propagation

Figure 5.47 presents the performance of the learning algorithm 'oss,' which depicts the training algorithm's performance across multiple iterations as well as the overall time it takes. Table 5.52 displays the value of best validation performance, maximum reached epochs, and time were taken to reach maximum epochs with oss.

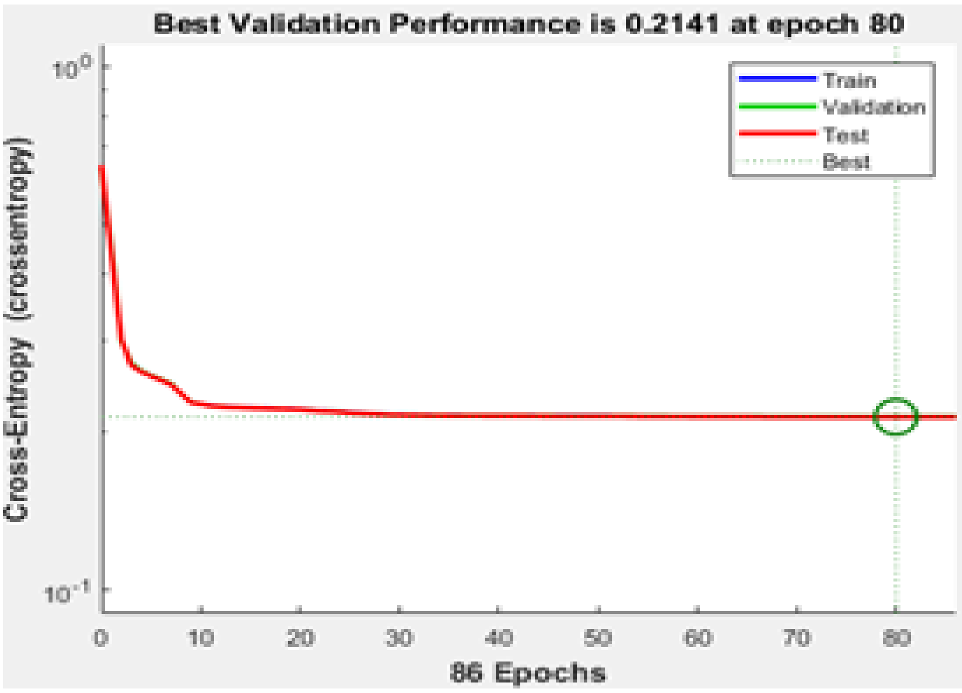

Figure 5.47: Performance of One Step Secant Back Propagation

TABLE 5.52: One Step Secant Back Propagation Performance

Algorithm	**OSS**
Best validation performance	0.2141
Maximum reached epochs	86
Time taken	35s

TABLE 5.53: Performance Evaluation

Performances Metrices	**Image** 1%-40%	**Image** 5%-40%	**Image** 7%-40%
Dice WM	0.8039	0.8005	0.7951
Dice GM	0.6943	0.6949	0.6946
Dice CSF	0.7331	0.7283	0.7236
MSE WM	0.0977	0.0999	0.1032
MSE GM	0.1492	0.1177	0.1489
MSE CSF	0.0621	0.0613	0.0635

TABLE 5.54: Performance Evaluation

		Mean		**Standard Deviation**	
	ROI	GT	Segmented	GT	Segmented
Image 1% -40%	WM	0.2006	0.2977	0.4005	0.4573
	GM	0.3044	0.1836	0.4602	0.3872
	CSF	0.1546	0.1119	0.3615	0.3484
Image 5% -40%	WM	0.2006	0.3001	0.4005	0.4583
	GM	0.3044	0.1837	0.4602	0.3873
	CSF	0.1546	0.1388	0.3615	0.3459
Image 7% -40%	WM	0.2006	0.303	0.4005	0.4596
	GM	0.3044	0.1831	0.4602	0.3868
	CSF	0.1546	0.1381	0.3615	0.3587

The segmentation results generated by ANN using the 'oss' algorithm for the test image are given below. Table **??** provides the dice similarity and MSE for the WM, GM, and CSF segments, and Table 5.54 provides the mean and Standard deviation for the segmented white matter, grey matter, and cerebrospinal fluid with 1% noise 40% INU,5% noise 40% INU, and 7% noise 40% INU.

5.11 Segmentation of Images using Resilient Back Propagation

Figure 5.48 depicts the training algorithm's performance graph, which displays the training algorithm's performance in terms of the number of iterations completed and total time consumed. Table 5.55 displays the value of best validation performance, maximum reached epochs, and time were taken to reach maximum epochs with rp.

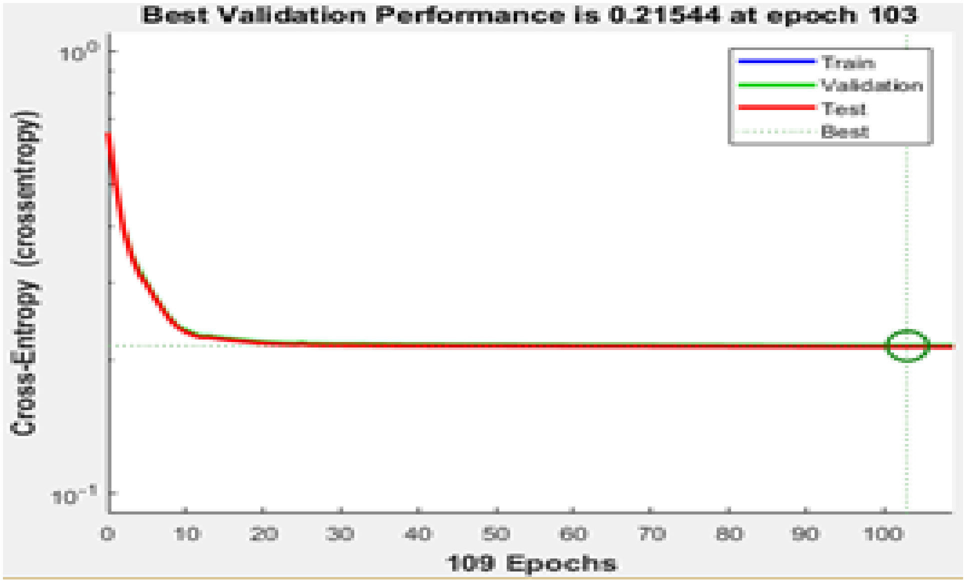

FIGURE 5.48: Performance of Resilient Back Propagation

TABLE 5.55: Resilient Back Propagation Performance

Algorithm	**RP**
Best validation performance	0.21544
Maximum reached epochs	109
Time taken	12s

The segmentation results for the test image generated by ANN using the 'rp' algorithm are shown below. Table 5.56 shows the dice similarity and MSE for segmented WM, GM, and CSF, while Table 5.57 shows the mean and standard

Table 5.56: Performance Evaluation

Performances Metrices	**Image** 0%-40%	**Image** 3%-40%	**Image** 9%-40%
Dice WM	0.7806	0.7808	0.6152
Dice GM	0.751	0.747	0.6384
Dice CSF	0.7127	0.726	0.7249
MSE WM	0.1127	0.1124	0.1944
MSE GM	0.1275	0.1293	0.1903
MSE CSF	0.0679	0.0637	0.0655

Table 5.57: Performance Evaluation

		Mean		**Standard Deviation**	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.3131	0.4005	0.4638
Image 0% -40%	GM	0.3044	0.2077	0.4602	0.4057
	CSF	0.1546	0.1124	0.3615	0.2852
	WM	0.2006	0.3122	0.4005	0.4634
Image 3% -40%	GM	0.3044	0.2067	0.4602	0.405
	CSF	0.1546	0.1259	0.3615	0.2826
	WM	0.2006	0.3046	0.4005	0.4603
Image 9% -40%	GM	0.3044	0.2219	0.4602	0.4155
	CSF	0.1546	0.1352	0.3615	0.355

deviation for segmented WM, GM, and CSF with 0% noise 40% INU, % noise 40% INU, and 9 percent noise 40% INU.

5.12 Segmentation of Images using Scale Congugate Gradient

The performance graph of the training algorithm‘scg’ is shown in Figure 5.49, which shows the performance of the training algorithm relating to the number of iterations it has gone through and the overall time it has taken. The value of best validation performance, maximum reached epochs, and time is taken to reach maximum epochs with scg are displayed in table 5.58.

The segmentation results for the test image generated by ANN using the ‘scg’ algorithm are shown below. Table 5.59 shows the dice similarity and MSE for the WM, GM, and CSF segments, while Table 5.60 shows the mean and standard

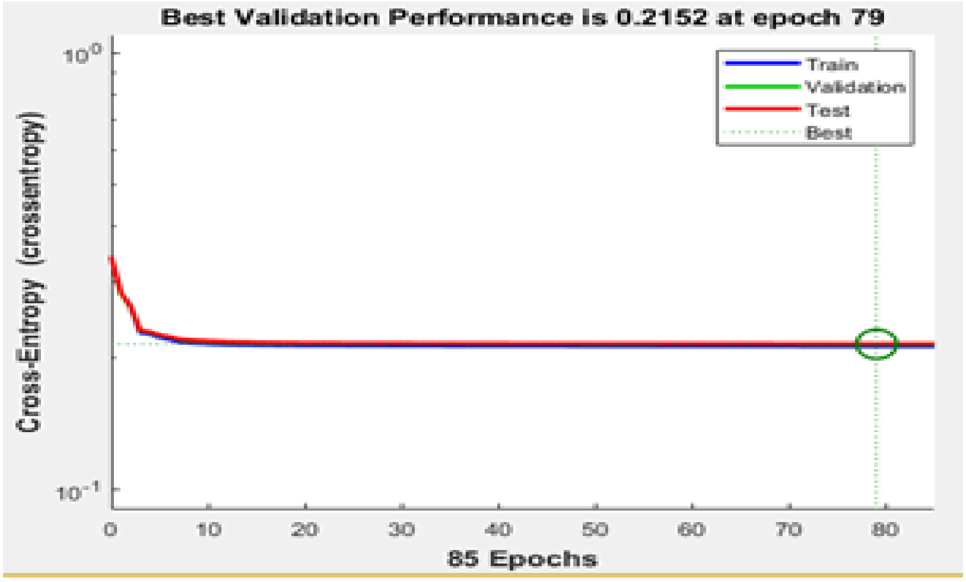

FIGURE 5.49: Performance of Scale Congugate Gradient

TABLE 5.58: Scale Congugate Gradient Performance

Algorithm	**SCG**
Best validation performance	0.2152
Maximum reached epochs	85
Time taken	18s

TABLE 5.59: Performance Evaluation

Performances Metrices	**Image** 1%-40%	**Image** 7%-40%	**Image** 9%-40%
Dice WM	0.765	0.758	0.6036
Dice GM	0.7904	0.7885	0.6809
Dice CSF	0.7227	0.7272	0.7253
MSE WM	0.1231	0.1278	0.2078
MSE GM	0.1144	0.115	0.1803
MSE CSF	0.068	0.0648	0.0632

deviation of WM, GM, and CSF with 1% noise 40% INU, 7% noise 40% INU, and 9% noise 40% INU.

The table given below displays the comparison of training algorithms.

TABLE 5.60: Performance Evaluation

		Mean		Standard Deviation	
	ROI	GT	Segmented	GT	Segmented
	WM	0.2006	0.3233	0.4005	0.4678
Image 1% -40%	GM	0.3044	0.2416	0.4602	0.4281
	CSF	0.1546	0.1148	0.3615	0.2891
	WM	0.2006	0.3276	0.4005	0.4694
Image 7% -40%	GM	0.3044	0.2394	0.4602	0.4267
	CSF	0.1546	0.1282	0.3615	0.2834
	WM	0.2006	0.3236	0.4005	0.4679
Image 9% -40%	GM	0.3044	0.2607	0.4602	0.439
	CSF	0.1546	0.1308	0.3615	0.3254

The comparision graph of all the training algorithms is shown in Figure 5.50, which shows the performance of all the training algorithms.

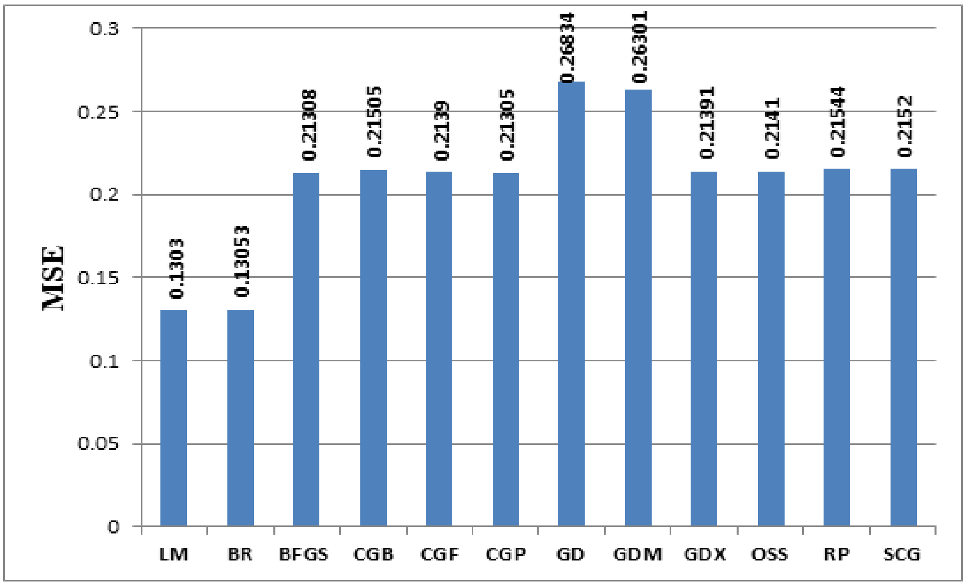

FIGURE 5.50: Comparision of All the Training Algorithms

TABLE 5.61: Comparison of Training Parameters

S.no	Algorithm	Best Performance	Achieved at	Maximum Reached Epoch	Time Taken
1	Levenberg-Marquardt	0.1303	31	37	32s

2	Bayesian Regularization	0.13053	1000	1000	9:59s
3	BFGS Quasi-Newton	0.21308	81	87	41s
4	Congugate Gradient with Powell/Beale Restarts	0.21505	20	26	12s
5	Congugate Gradient with Fletcher powell	0.2139	53	59	24s
6	Congugate Gradient with Polak-Ribiere	0.21305	92	98	45s
7	Gradient Descent Back Propogation	0.26834	1000	1000	1:53s
8	Gradient Descent with Momentum	0.26301	1000	1000	1:56s
9	Gradient Descent with Variable Learning Rate	0.21391	186	192	23s

10	One Step Secant Back Propagation	0.2141	80	86	35s
11	Resilient back Propagation	0.21544	103	109	12s
12	Scale Congugate Gradient	0.2152	79	85	18s

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This chapter deals with the conclusion and future directions of the study. The test results show that the Bayesian Regularization technique minimizes the inaccuracy to a significant degree, however, it will take a long time and a lot of computing resources to accomplish this. Furthermore, the Bayesian Regularization algorithm's performance tends to get worse as it over-fits the network with training data.

The Gradient Descent algorithm, on the other hand, is quick since it accomplishes 1000 epochs in a short period, as well as the error, keeps dropping with each of the it-erations. However, the rate of descent slows down significantly, so it takes longer to reduce the more errors. The One Step Secant Backpropagation method outperforms Gradient Descent in terms of reliability because OSS Backpropagation is faster and less error-prone.

The SCG technique outperforms OSS Backpropaga-tion in terms of performance, time makes it superior. It takes almost half of them as long as the OSS technique. The Broyden-Fletcher-Goldfarb-Shannon method (BFGS) outperforms others, but it takes much longer to run, due to which t has a lower chance of being adopted. The LM algorithm provides more accurate findings in less time as compared to the BFGS algorithm. Furthermore, this approach is more efficient because it only takes 31 epochs to get the result. The Scaled Conjugate Gradient strategy should be utilized if network training is completed rapidly, but if a tiny amount of error cannot be accepted, the Levenberg-Marquardat technique should be used at the cost of more time and processing power. The table given below displays the comparison of training algorithms.

6.2 Future Direction

Some of the aspects will be considered in the future study, as stated below.
A simulated brain MRI dataset was used in this investigation. The algorithms can be applied to real MRI datasets.

Bibliography

[1] M. Bahrami, M. Akbari, S. A. Bagherzadeh, A. Karimipour, M. Afrand, and M. Goodarzi, "Develop 24 dissimilar anns by suitable architectures & training algorithms via sensitivity analysis to better statistical presentation: Measure mses between targets & ann for fe–cuo/eg–water nanofluid," *Physica A: Statistical Mechanics and Its Applications*, vol. 519, no. 9, pp. 159–168, 2019.

[2] A. Ghosh and B. Soni, "An automatic tumor identification process to classify mri brain images," in *Data Science*, vol. 26, pp. 315–327, Springer, 2021.

[3] G. Mohan and M. M. Subashini, "Mri based medical image analysis: Survey on brain tumor grade classification," *Biomedical Signal Processing and Control*, vol. 39, no. 7, pp. 139–161, 2018.

[4] T. Lei, X. Jia, Y. Zhang, L. He, H. Meng, and A. K. Nandi, "Significantly fast and robust fuzzy c-means clustering algorithm based on morphological reconstruction and membership filtering," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 3027–3041, 2018.

[5] C. P. Junior, G. V. Cavallieri, F. A. da Silva, G. L. Fernandes, G. A. Nai, A. K. M. Salge, J. G. Puhle, D. T. d. R. e Silva, D. R. Pereira, F. de Azevedo Mello, *et al.*, "Digital image processing: a useful tool in the analysis of lung injuries caused by chronic inhalation of agricultural herbicides," *Environmental Science and Pollution Research*, vol. 16, no. 8, pp. 1–7, 2021.

[6] E. Hamuda, M. Glavin, and E. Jones, "A survey of image processing techniques for plant extraction and segmentation in the field," *Computers and Electronics in Agriculture*, vol. 125, no. 5, pp. 184–199, 2016.

[7] P. ANUSHKA and R. UPAKA, "Comparison of different artificial neural network (ann) training algorithms to predict the atmospheric temperature in tabuk, saudi arabia," *Mausam*, vol. 71, no. 2, pp. 233–244, 2020.

[8] X. Wang, F. Liu, Y. Gao, C.-h. Xue, R. W. Li, and Q.-j. Tang, "Transcriptome analysis revealed anti-obesity effects of the sodium alginate in high-fat diet-induced obese mice," *International journal of biological macromolecules*, vol. 115, no. 7, pp. 861–870, 2018.

[9] I. Despotović, B. Goossens, and W. Philips, "Mri segmentation of the human brain: challenges, methods, and applications," *Computational and mathematical methods in medicine*, vol. 2015, no. 6, 2015.

[10] A. Mahbod, M. Chowdhury, Ö. Smedby, and C. Wang, "Automatic brain segmentation using artificial neural networks with shape context," *Pattern Recognition Letters*, vol. 101, no. 8, pp. 74–79, 2018.

[11] S. Yuheng and Y. Hao, "Image segmentation algorithms overview," *arXiv preprint arXiv:1707.02051*, vol. 76, no. 6, 2017.

[12] C. Amza, "A review on neural network-based image segmentation techniques," *De Montfort University, Mechanical and Manufacturing Engg., The Gateway Leicester, LE1 9BH, United Kingdom*, vol. 63, no. 11, pp. 1–23, 2012.

[13] I. Scholl, T. Aach, T. M. Deserno, and T. Kuhlen, "Challenges of medical image processing," *Computer science-Research and development*, vol. 26, no. 1, pp. 5–13, 2011.

[14] A. J. Worth and D. N. Kennedy, "Segmentation of magnetic resonance brain images using analogue constraint satisfaction neural networks," *Image and Vision Computing*, vol. 12, no. 6, pp. 345–354, 1994.

[15] T. Zuva, O. O. Olugbara, S. O. Ojo, and S. M. Ngwira, "Image segmentation, available techniques, developments and open issues," *Canadian Journal on Image Processing and Computer Vision*, vol. 2, no. 3, pp. 20–29, 2011.

[16] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks—a review," *Pattern recognition*, vol. 35, no. 10, pp. 2279–2301, 2002.

[17] M. J. Moghaddam and H. Soltanian-Zadeh, "Medical image segmentation using artificial neural networks," *Artificial Neural Networks-Methodological Advances and Biomedical Applications*, vol. 22, no. 2, pp. 121–138, 2011.

[18] M. N. Ahmed and A. A. Farag, "Two-stage neural network for volume segmentation of medical images," *Pattern Recognition Letters*, vol. 18, no. 11-13, pp. 1143–1151, 1997.

[19] D. Kaur and Y. Kaur, "Various image segmentation techniques: a review," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 5, pp. 809–814, 2014.

[20] A. Işın, C. Direkoğlu, and M. Şah, "Review of mri-based brain tumor image segmentation using deep learning methods," *Procedia Computer Science*, vol. 102, no. 7, pp. 317–324, 2016.

[21] D. J. Withey and Z. J. Koles, "Medical image segmentation: Methods and software," in *2007 Joint Meeting of the 6th International Symposium on Noninvasive Functional Source Imaging of the Brain and Heart and the International Conference on Functional Biomedical Imaging*, vol. 82, pp. 140–143, IEEE, 2007.

[22] Y. Song and H. Yan, "Image segmentation techniques overview," in *2017 Asia Modelling Symposium (AMS)*, vol. 28, pp. 103–107, IEEE, 2017.

[23] D. J. Bora and A. K. Gupta, "A novel approach towards clustering based image segmentation," *arXiv preprint arXiv:1506.01710*, vol. 19, no. 6, 2015.

[24] W. E. Reddick, J. O. Glass, E. N. Cook, T. D. Elkin, and R. J. Deaton, "Automated segmentation and classification of multispectral magnetic resonance images of brain using artificial neural networks," *IEEE Transactions on medical imaging*, vol. 16, no. 6, pp. 911–918, 1997.

[25] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Interactive image segmentation by maximal similarity based region merging," *Pattern Recognition*, vol. 43, no. 2, pp. 445–456, 2010.

[26] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 63, pp. 7014–7023, 2018.

[27] D. Q. Zeebaree, H. Haron, A. M. Abdulazeez, and D. A. Zebari, "Machine learning and region growing for breast cancer segmentation," in *2019 International Conference on Advanced Science and Engineering (ICOASE)*, vol. 51, pp. 88–93, IEEE, 2019.

[28] W. Tao, H. Jin, and Y. Zhang, "Color image segmentation based on mean shift and normalized cuts," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1382–1389, 2007.

[29] W.-X. Kang, Q.-Q. Yang, and R.-P. Liang, "The comparative research on image segmentation algorithms," in *2009 First International Workshop on Education Technology and Computer Science*, vol. 2, pp. 703–707, IEEE, 2009.

[30] T. Pavlidis and Y.-T. Liow, "Integrating region growing and edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 3, pp. 225–233, 1990.

[31] M. Abd Elaziz, D. Oliva, A. A. Ewees, and S. Xiong, "Multi-level thresholding-based grey scale image segmentation using multi-objective multi-verse optimizer," *Expert Systems with Applications*, vol. 125, no. 3, pp. 112–129, 2019.

[32] T. Y. Goh, S. N. Basah, H. Yazid, M. J. A. Safar, and F. S. A. Saad, "Performance analysis of image thresholding: Otsu technique," *Measurement*, vol. 114, no. 13, pp. 298–307, 2018.

[33] R. B. Aghdam, A. S. B. Ghiasi, P. Fatemi, N. S. Hashemi, *et al.*, "Challenges in brain magnetic resonance image segmentation," *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, vol. 27, no. 1, pp. 122–138, 2017.

[34] P. Jaiswal, N. K. Gupta, and A. Ambikapathy, "Comparative study of various training algorithms of artificial neural network," in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, vol. 207, pp. 1097–1101, IEEE, 2018.

[35] N. Coskun and T. Yildirim, "The effects of training algorithms in mlp network on image classification," in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 2, pp. 1223–1226, IEEE, 2003.

[36] P. Chopra, R. K. Sharma, M. Kumar, *et al.*, "Artificial neural networks for the prediction of compressive strength of concrete," *International journal of applied science and engineering*, vol. 13, no. 3, pp. 187–204, 2015.

[37] E.-S. A. El-Dahshan, H. M. Mohsen, K. Revett, and A.-B. M. Salem, "Computer-aided diagnosis of human brain tumor through mri: A survey and a new algorithm," *Expert systems with Applications*, vol. 41, no. 11, pp. 5526–5545, 2014.

[38] B. Srinivas and G. S. Rao, "Performance evaluation of fuzzy c means segmentation and support vector machine classification for mri brain tumor," in *Soft computing for problem solving*, vol. 20, pp. 355–367, Springer, 2019.

[39] J. Song and Z. Zhang, "A modified robust fcm model with spatial constraints for brain mr image segmentation," *Information*, vol. 10, no. 2, p. 74, 2019.

[40] P. Lin, Y. Yang, C.-X. Zheng, and J.-W. Gu, "An efficient automatic framework for segmentation of mri brain image," in *The Fourth International*

Conference onComputer and Information Technology, 2004. CIT'04., vol. 17, pp. 896–900, IEEE, 2004.

[41] P. Lin, Y. Yang, C.-X. Zheng, and J.-W. Gu, "An efficient automatic framework for segmentation of mri brain image," in *The Fourth International Conference onComputer and Information Technology, 2004. CIT'04.*, vol. 11, pp. 896–900, IEEE, 2004.

[42] B. Khorram and M. Yazdi, "A new optimized thresholding method using ant colony algorithm for mr brain image segmentation," *Journal of digital imaging*, vol. 32, no. 1, pp. 162–174, 2019.

[43] H. Hassannejad, M. S. Pakbaz, and R. Mehdizadeh, "Comparison and evaluation of artificial neural network (ann) training algorithms in predicting soil type classification," *Pharmacology and Life Sciences Bull. Env. Pharmacol. Life Sci*, vol. 4, no. 5, pp. 212–218, 2015.

[44] T. Awolusi, O. Oke, O. Akinkurolere, A. Sojobi, and O. Aluko, "Performance comparison of neural network training algorithms in the modeling properties of steel fiber reinforced concrete," *Heliyon*, vol. 5, no. 1, p. e01115, 2019.

[45] B. Eren, M. Yaqub, and V. Eyüpoğlu, "Assessment of neural network training algorithms for the prediction of polymeric inclusion membranes efficiency," *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, vol. 20, no. 3, pp. 533–542, 2016.

[46] R. Tabbussum and A. Q. Dar, "Comparative analysis of neural network training algorithms for the flood forecast modelling of an alluvial himalayan river," *Journal of Flood Risk Management*, vol. 13, no. 4, p. e12656, 2020.

[47] H. Asgari, X. Chen, and R. Sainudiin, "Analysis of ann-based modelling approach for industrial systems," *International Journal of Innovation, Management and Technology*, vol. 4, no. 1, p. 165, 2013.

[48] H. Asgari, X. Q. Chen, M. B. Menhaj, and R. Sainudiin, "Ann-based system identification, modelling and control of gas turbines–a review," in *Advanced Materials Research*, vol. 622, pp. 611–617, Trans Tech Publ, 2013.

[49] J. A. Ramírez-Quintana, M. I. Chacon-Murguia, and J. F. Chacon-Hinojos, "Artificial neural image processing applications: A survey.," *Engineering Letters*, vol. 20, no. 1, 2012.

[50] A. Miller, B. Blott, *et al.*, "Review of neural network applications in medical imaging and signal processing," *Medical and Biological Engineering and Computing*, vol. 30, no. 5, pp. 449–464, 1992.

[51] P. K. Chahal, S. Pandey, and S. Goel, "A survey on brain tumor detection techniques for mr images.," *Multimedia Tools & Applications*, vol. 79, no. 19, 2020.

[52] M. Negnevitsky and A. Intelligence, "A guide to intelligent systems," *Artificial Intelligence, 2nd edition, pearson Education*, vol. 207, no. 71, 2005.

[53] S.-C. Wang, "Artificial neural network," in *Interdisciplinary computing in java programming*, vol. 43, pp. 81–100, Springer, 2003.

[54] L. A. Kurgan, K. J. Cios, R. Tadeusiewicz, M. Ogiela, and L. S. Goodenday, "Knowledge discovery approach to automated cardiac spect diagnosis," *Artificial intelligence in medicine*, vol. 23, no. 2, pp. 149–169, 2001.

[55] J. Jiang, P. Trundle, and J. Ren, "Medical image analysis with artificial neural networks," *Computerized Medical Imaging and Graphics*, vol. 34, no. 8, pp. 617–631, 2010.

[56] A. Papadopoulos, D. I. Fotiadis, and A. Likas, "An automatic microcalcification detection system based on a hybrid neural network classifier," *Artificial intelligence in Medicine*, vol. 25, no. 2, pp. 149–167, 2002.

[57] X. Zhang, M. Kanematsu, H. Fujita, X. Zhou, T. Hara, R. Yokoyama, and H. Hoshi, "Application of an artificial neural network to the computer-aided

differentiation of focal liver disease in mr imaging," *Radiological Physics and Technology*, vol. 2, no. 2, pp. 175–182, 2009.

[58] C. Kondo, T. Kondo, and J. Ueno, "Three-dimensional medical image analysis of the heart by the revised gmdh-type neural network self-selecting optimum neural network architecture," *Artificial life and robotics*, vol. 14, no. 2, pp. 123–128, 2009.

[59] H. Azami, M. Malekzadeh, and S. Sanei, "A new neural network approach for face recognition based on conjugate gradient algorithms and principal component analysis," *Journal of mathematics and computer Science*, vol. 6, no. 3, pp. 166–175, 2013.

[60] M. H. Beale, M. T. Hagan, and H. B. Demuth, "Neural network toolbox user's guide," *The MathWorks Inc*, vol. 103, no. 34, 1992.

[61] A. R. Abd Ellah, M. H. Essai, and A. Yahya, "Comparison of different back-propagation training algorithms using robust m-estimators performance functions," in *2015 Tenth International Conference on Computer Engineering & Systems (ICCES)*, vol. 10, pp. 384–388, IEEE, 2015.

[62] S. Khirirat, H. R. Feyzmahdavian, and M. Johansson, "Mini-batch gradient descent: Faster convergence under data sparsity," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, vol. 13, pp. 2880–2887, IEEE, 2017.

[63] D. Utomo, "Stock price prediction using back propagation neural network based on gradient descent with momentum and adaptive learning rate," *Journal of Internet Banking and Commerce*, vol. 22, no. 3, pp. 1–16, 2017.

[64] I. Gitman, H. Lang, P. Zhang, and L. Xiao, "Understanding the role of momentum in stochastic gradient methods," *arXiv preprint arXiv:1910.13962*, vol. 17, no. 12, 2019.

[65] A. Wanto, S. R. Andani, P. Poningsih, R. Dewi, M. R. Lubis, W. Saputra, and I. O. Kirana, "Analysis of standard gradient descent with gd momentum and adaptive lr for spr prediction," vol. 65, no. 54, 2018.

[66] N. Baykal and A. M. Erkmen, "Resilient backpropagation for rbf networks," in *KES'2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings (Cat. No. 00TH8516)*, vol. 2, pp. 624–627, IEEE, 2000.

[67] R. S. Naoum, N. A. Abid, and Z. N. Al-Sultani, "An enhanced resilient backpropagation artificial neural network for intrusion detection system," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 12, no. 3, p. 11, 2012.

[68] N. Andrei, "Scaled conjugate gradient algorithms for unconstrained optimization," *Computational Optimization and Applications*, vol. 38, no. 3, pp. 401–416, 2007.

[69] M. Wahyudi, M. Safii, M. Zarlis, *et al.*, "Backpropagation network optimization using one step secant (oss) algorithm," in *IOP Conference Series: Materials Science and Engineering*, vol. 769, p. 012037, IOP Publishing, 2020.

[70] N. Ginantra, G. W. Bhawika, G. A. Daengs, P. D. Panjaitan, M. A. Arifin, A. Wanto, M. Amin, H. Okprana, A. Syafii, and U. Anwar, "Performance one-step secant training method for forecasting cases," in *Journal of Physics: Conference Series*, vol. 1933, p. 012032, IOP Publishing, 2021.

[71] R. Shang, B. Ara, I. Zada, S. Nazir, Z. Ullah, and S. U. Khan, "Analysis of simple k-mean and parallel k-mean clustering for software products and organizational performance using education sector dataset," *Scientific Programming*, vol. 2021, no. 54, 2021.

[72] R. Ahuja, A. Solanki, and A. Nayyar, "Movie recommender system using k-means clustering and k-nearest neighbor," in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, vol. 103, pp. 263–268, IEEE, 2019.

[73] Y. Jeong, J. Lee, J. Moon, J. H. Shin, and W. D. Lu, "K-means data clustering with memristor networks," *Nano letters*, vol. 18, no. 7, pp. 4447–4453, 2018.

[74] T. G. Debelee, F. Schwenker, S. Rahimeto, and D. Yohannes, "Evaluation of modified adaptive k-means segmentation algorithm," *Computational Visual Media*, vol. 5, no. 4, pp. 347–361, 2019.

[75] M. N. Akhtar, W. Ahmed, M. R. Kakar, E. A. Bakar, A. Othman, and M. Bueno, "Implementation of parallel k-means algorithm to estimate adhesion failure in warm mix asphalt," *Advances in Civil Engineering*, vol. 2020, no. 23, 2020.

[76] S. Vyas and A. Prasad, "Analysis of algorithms k-means and apriori for data mining," in *Rising Threats in Expert Applications and Solutions*, vol. 11, pp. 187–191, Springer, 2021.

[77] A. Shukla, S. Kumar, and H. Singh, "Ann based execution time prediction model and assessment of input parameters through ism.," *Int. Arab J. Inf. Technol.*, vol. 17, no. 5, pp. 683–691, 2020.

[78] J. Lian, Z. Yang, J. Liu, W. Sun, L. Zheng, X. Du, Z. Yi, B. Shi, and Y. Ma, "An overview of image segmentation based on pulse-coupled neural network," *Archives of Computational Methods in Engineering*, vol. 28, no. 2, pp. 387–403, 2021.