

**EARLY DIAGNOSIS OF  
ALZHEIMERS DISEASE THROUGH  
MRI (MAGNETIC RESONANCE  
IMAGING) IMAGE PROCESSING  
AND CONVOLUTIONAL NEURAL  
NETWORKS**

A DISSERTATION SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY  
IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF ENGINEERING

Brinnel Dsouza

School of Engineering, Computer and Mathematical Sciences

October 2024

# **Attestation of Authorship**

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

A handwritten signature in black ink that reads "Brinjal". The signature is written diagonally from top-left to bottom-right, with a small dot at the end.

---

Signature of candidate

# Acknowledgements

I would like to express my deepest gratitude to all those who have supported and guided me throughout the course of this project. First and foremost, I want to express my gratitude to my project supervisor at Auckland University of Technology, Prof. Dr. Xuejun Li, for his patience, advice, and unwavering support during this research endeavour. His thorough understanding and perceptive criticism were invaluable in getting this work finished. I would also like to extend my heartfelt thanks to Mr. Bumjun Kim, Senior Technician for his technical assistance in helping me gain remote desktop access and for promptly resolving any issues with the remote computer whenever needed. His help was crucial in enabling me to continue my work smoothly. I extend my thanks to my family and friends for their unwavering support and encouragement, which have been vital to my success. Their patience and understanding have motivated me to stay focused and strive for excellence. Thank you all for your immense support in making this project a reality.

Brinnel DSouza

October 2024

# Abstract

Alzheimer's disease (AD) is the most prevalent form of dementia, leading to progressive cognitive decline, memory loss, and impaired daily functioning. Early diagnosis is essential. This thesis explores two distinct methodologies for detecting Alzheimer's disease (AD) from MRI brain scans: image processing techniques and Convolutional Neural Networks (CNNs). The primary focus is to assess the accuracy of both approaches. The main research findings of this study are as follows:

Image processing methods, including Gaussian blur, edge detection and Otsu thresholding are applied to enhance MRI brain scans. These techniques are used to calculate the ratio of black and white pixels in the brain, aiding in the identification of visual signs of brain atrophy associated with Alzheimer's disease. The accuracy of this approach in detecting AD is evaluated based on its ability to highlight key structural changes in the brain.

A CNN model is implemented to automatically analyze MRI scans. The CNN is trained to recognize patterns in the brain that are indicative of Alzheimer's disease, offering a higher level of precision compared to traditional methods. The model's performance is measured in terms of its accuracy in identifying Alzheimer's-related changes, even in the early stages of the disease.

This study provides a comparative analysis of the accuracy of image processing techniques and CNN-based deep learning models for Alzheimer's disease detection, contributing valuable insights into the strengths and limitations of each methodology.

# Contents

<b>Attestation of Authorship</b>	ii
<b>Acknowledgements</b>	iii
<b>Abstract</b>	iv
<b>List of Tables</b>	viii
<b>List of Figures</b>	ix
<b>1 Introduction</b>	1
1.1 Background . . . . .	1
1.2 Rationale of the Research . . . . .	3
1.3 Objective . . . . .	3
1.4 Research Questions . . . . .	4
1.5 Scope and Limitations . . . . .	5
1.6 Organisation of the thesis . . . . .	6
<b>2 Literature Review</b>	8
2.1 Traditional Method . . . . .	9
2.1.1 Mini-Mental State Examination (MMSE) . . . . .	9
2.1.2 The Alzheimer's Disease Assessment Scale-Cognitive Subscale (ADAS-Cog) . . . . .	10
2.1.3 Rey Auditory Verbal Learning Test (RAVLT) . . . . .	11
2.2 Role of Neuroimaging . . . . .	11
2.3 Image Processing Techniques . . . . .	12
2.3.1 Preprocessing Techniques . . . . .	13
2.3.2 Interpolation technique . . . . .	15
2.3.3 Thresholding Techniques . . . . .	18
2.3.4 Edge detection techniques . . . . .	21
2.4 Convolution neural network techniques . . . . .	24
2.4.1 VGG16 . . . . .	25
2.4.2 AlexNet . . . . .	26
2.4.3 MobileNet V2 . . . . .	27
2.5 Research Gaps and Implications . . . . .	29

2.6	Summary . . . . .	30
<b>3</b>	<b>Methodology</b>	<b>31</b>
3.1	Image Processing Method . . . . .	31
3.2	Convolution Neural Network Method . . . . .	34
3.3	Summary . . . . .	36
<b>4</b>	<b>Design and Implementation</b>	<b>37</b>
4.1	Image Processing Technique . . . . .	38
4.1.1	Input Data . . . . .	38
4.1.2	Image Resizing and Interpolation . . . . .	39
4.1.3	Grayscale conversion . . . . .	40
4.1.4	Gaussian Blur . . . . .	41
4.1.5	Edge Detection . . . . .	42
4.1.6	Otsu Thresholding . . . . .	43
4.1.7	Morphological Operations . . . . .	44
4.1.8	Contour Detection . . . . .	46
4.1.9	Red border around the brain . . . . .	47
4.1.10	Pixel Counting . . . . .	48
4.1.11	Excel Reporting . . . . .	49
4.2	Convolution Neural Network Technique . . . . .	51
4.2.1	Data Collection and Labeling for Alzheimer's Disease Image Dataset . . . . .	51
4.2.2	Model Training with Validation for Alzheimer's Disease Classification . . . . .	56
4.2.3	Model Evaluation on Training Data . . . . .	57
4.2.4	Model Training for One Epoch with Validation . . . . .	58
4.2.5	Model Evaluation on Test Data . . . . .	59
4.2.6	Model Prediction and Class Label Mapping . . . . .	59
4.2.7	Plotting Training and Validation Accuracy Over Epochs . . . . .	60
4.2.8	Plotting Training and Validation Loss Over Epochs . . . . .	62
4.2.9	Model Performance Evaluation Using Classification Report and Accuracy . . . . .	64
4.3	Summary . . . . .	65
<b>5</b>	<b>Results and Discussions</b>	<b>67</b>
5.1	Results . . . . .	67
5.1.1	Accuracy using image processing technique . . . . .	67
5.1.2	Accuracy using Convolution Neural Network . . . . .	70
5.2	Discussion . . . . .	73
5.3	Summary . . . . .	75
<b>6</b>	<b>Conclusion and Recommendation for Future Work</b>	<b>76</b>
6.1	Conclusion . . . . .	76

6.2 Recommendation for Future Work . . . . .	77
<b>7 Appendix</b>	<b>79</b>

# List of Tables

2.1	Subtests and Cognitive Domains in ADAS-Cog) . . . . .	10
5.1	Confusion Matrix using image processing technique for AD detetion .	68
5.2	Classification Report . . . . .	71
5.3	Confusion Matrix using CNN for AD detetion . . . . .	72

# List of Figures

1.1	Normal vs. Alzheimer's brain showing shrinkage in AD [5] . . . . .	2
1.2	MRI scans showing brain changes across four stages [8] . . . . .	4
2.1	Scoring the WORLD Task in the MMSE Using the Line Method [10]	10
2.2	MRI scans showing increasing gray matter loss from normal to AD [21] . . . . .	12
2.3	Effect of Gaussian Blur: Left shows noise, right shows noise reduction after Gaussian blur [24] . . . . .	14
2.4	Brain stripping on brain [26] . . . . .	15
2.5	Image Upsampling: $2 \times 2$ to $4 \times 4$ Matrix Using Nearest Neighbor Interpolation [28] . . . . .	16
2.6	Interpolation: Before, After, and No Interpolation Effects [30] . .	17
2.7	$2 \times 2$ to $4 \times 4$ Upscaling Using Bicubic Interpolation [32] . . . .	18
2.8	Otsu thresholding on the image [36] . . . . .	20
2.9	Comparison of Thresholding Techniques [38] . . . . .	21
2.10	Original Image with Sobel X and Sobel Y Gradients [38] . . . . .	23
2.11	Comparison of Gray Image and Canny Edge Detection [38] . . . .	24
2.12	The dimensions of each block are reduced post-pooling to condense information [44] . . . . .	26
2.13	AlexNet architecture [45] . . . . .	27
2.14	MobileNet V2 architecture [48] . . . . .	29
3.1	Methodology for Image Processing . . . . .	33
3.2	Methodology for Image Processing . . . . .	35
4.1	Demented MRI image . . . . .	39
4.2	Non-Demented MRI image . . . . .	39
4.3	Dimensions of MRI image before bicubic interpolation . . . . .	40
4.4	Dimensions of MRI image after bicubic interpolation . . . . .	40
4.5	Gray conversion on Demented MRI image . . . . .	41
4.6	Gray conversion on Non-Demented MRI image . . . . .	41
4.7	Gaussian Blur on Demented MRI image . . . . .	42
4.8	Gaussian Blur on Non-Demented MRI image . . . . .	42
4.9	Canny edge detection on Demented MRI image . . . . .	43
4.10	Canny edge detection on Non-Demented MRI image . . . . .	43
4.11	Otsu threshold on Demented MRI image . . . . .	44

4.12	Otsu threshold on Non-Demented MRI image . . . . .	44
4.13	Morphological Operations on Demented MRI image . . . . .	45
4.14	Morphological Operations on Non-Demented MRI image . . . . .	45
4.15	Contour detection on Demented MRI image . . . . .	47
4.16	Contour detection on Non-Demented MRI image . . . . .	47
4.17	Red border around the on Demented MRI image . . . . .	48
4.18	Red border around the Non-Demented MRI image . . . . .	48
4.19	Excel report of the MRI image . . . . .	50
4.20	Class Distribution and Dataset Overview . . . . .	52
4.21	Dataset Split Overview: Training, Testing, and Validation Set Sizes	53
4.22	Data Generator Initialization: Number of Images in Each Sets . .	54
4.23	Xception CNN Architecture and Parameters . . . . .	55
4.24	Epoch-wise Training and Validation Accuracy and Loss . . . . .	57
4.25	Training Data Evaluation: Accuracy and Loss Results . . . . .	58
4.26	Epoch-wise Training and Validation Accuracy and Loss . . . . .	59
4.27	Test Data Evaluation: Loss and Accuracy Results . . . . .	59
4.28	Prediction Process: Test Set Evaluation Progress . . . . .	60
4.29	Training vs Validation Accuracy Over Epochs . . . . .	62
4.30	Training vs Validation Loss Over Epochs . . . . .	64
4.31	Classification Report and Accuracy of AD Detection Model . . .	65

# **Chapter 1**

## **Introduction**

### **1.1 Background**

Alzheimer's disease (AD) represents a significant challenge in healthcare as a progressive brain disorder that severely impacts thinking abilities, memory, and cognitive processes. Symptoms such as memory loss, impaired language abilities, and irrational problem-solving skills usually emerge after the age of 60. Alzheimer's disease is the primary cause of dementia, accounting for 50 to 80 percent of cases [1]. Worldwide, over 55 million people are impacted by dementia, with almost 10 million new cases diagnosed each year [2]. Early diagnosis is essential for prompt interventions, which can limit the disease's progression more effectively than later-stage treatments [3].

Early detection of AD is greatly aided by neuroimaging, especially magnetic resonance imaging (MRI). MRI enables detailed observation of changes in the brain's anatomy, providing crucial information about the structural integrity of brain tissues. It plays a vital role in detecting atrophy in critical regions like the hippocampus, which is severely affected in the early stages of Alzheimer's disease [4].

Comparison of a normal brain (left) and a brain affected by Alzheimer's Disease (AD) (right). The AD brain shows significant shrinkage and loss of brain tissue, particularly

in the cerebral cortex and hippocampus, which are crucial for memory and cognitive function. The bottom row of Figure 1.1 displays coronal views, highlighting the enlargement of the ventricles in the AD brain, a result of brain atrophy.

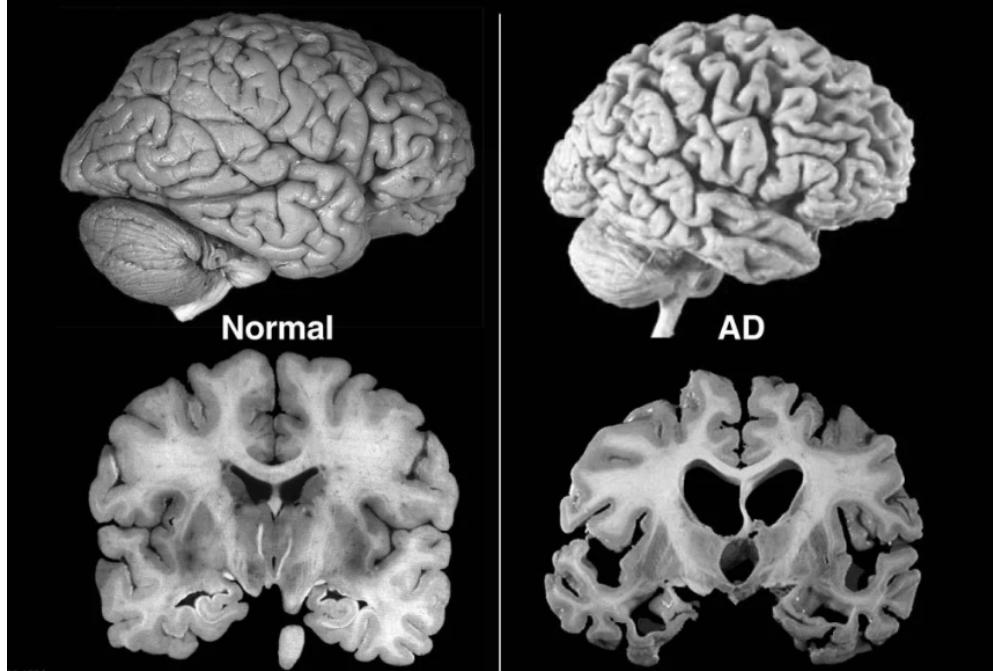


Figure 1.1: Normal vs. Alzheimer's brain showing shrinkage in AD [5]

The amyloid hypothesis and neurofibrillary tangles are fundamental to understanding AD's pathogenesis. Abnormal processing of amyloid precursor protein leads to amyloid-beta peptides forming plaques, causing neuroinflammation and oxidative stress. Similarly, tau protein becomes hyperphosphorylated, forming neurofibrillary tangles that disrupt neuronal structures, leading to cell death. These mechanisms create a destructive feedback loop exacerbating Alzheimer's progression [6].

Incorporating Convolutional Neural Networks (CNNs) in AD diagnosis enhances detection and facilitates the classification of the disease into stages such as Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented. This capability enables the development of tailored therapeutic strategies suited to the severity of the disease.

A critical aspect of this process is the preprocessing of MRI data, which involves normalization, feature enhancement, and segmentation of critical brain regions for focused analysis.

This research explores the use of MRI image processing and CNNs to detect early signs of Alzheimer's, aiming to improve the robustness and applicability of diagnostic tools across diverse populations.

## 1.2 Rationale of the Research

Alzheimer's disease (AD) is a growing global health concern, affecting millions and burdening caregivers and healthcare systems. Early diagnosis is crucial to slow symptom progression and improve outcomes, but current methods often detect the disease too late for effective intervention. Magnetic Resonance Imaging (MRI) reveals structural brain changes linked to AD [7]. This highlights the need for advanced, automated tools that can more accurately and reliably identify early signs of neurodegeneration.

This project seeks to address these limitations by integrating image processing techniques and Convolutional Neural Networks (CNNs) with MRI data. CNNs can automatically detect complex patterns in brain images, offering improved accuracy in early diagnosis. By enhancing the detection of Alzheimer's disease, this research aims to contribute to more timely and targeted interventions, ultimately improving patient care and reducing the burden on healthcare systems.

## 1.3 Objective

The objective of this project is to develop a robust diagnostic tool that combines Magnetic Resonance Imaging (MRI) data with two methodologies—Image Processing and Convolutional Neural Networks (CNNs)—to enhance the detection and classification of

Alzheimer's disease (AD). The Image Processing technique will focus on classifying brain scans into two categories: Alzheimer's (AD) and Non-Alzheimer's (Non-AD) by analyzing black and white pixels ratio present in the data. The second methodology, using CNNs, will further classify brain scans into four categories: Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented. By detecting more subtle and complex patterns in MRI data, the CNN model will enable a more detailed classification of the disease's progression. Together, these methodologies aim to create a comprehensive tool for both early detection and stage-wise classification of Alzheimer's disease, leading to more personalized and timely interventions.

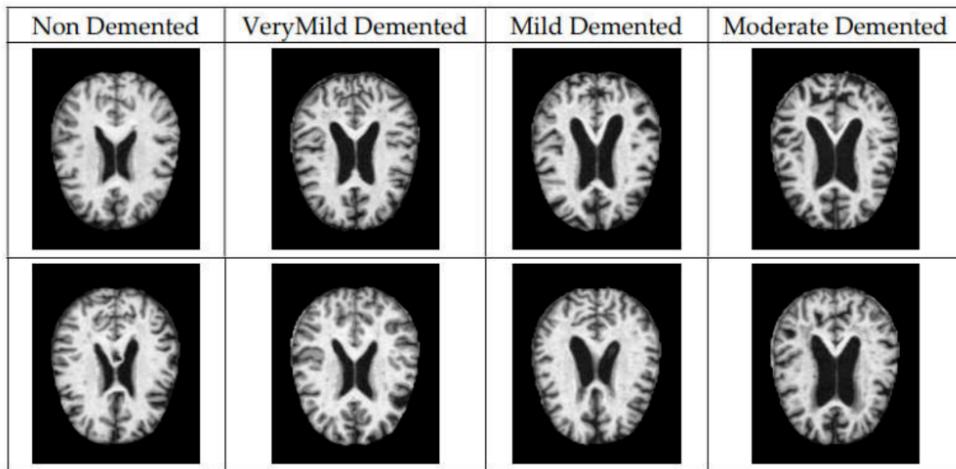


Figure 1.2: MRI scans showing brain changes across four stages [8]

## 1.4 Research Questions

The following three primary research questions have been formulated to direct the research:

RQ1: How can image processing techniques be used to accurately differentiate between Alzheimer's (AD) and non-Alzheimer's (Non-AD) brain MRI scans?

RQ2: What is the effectiveness of Convolutional Neural Networks (CNNs) in classifying MRI data into different stages of Alzheimer's disease: Non-Demented, Very Mild

Demented, Mild Demented, and Moderate Demented?

RQ3: How do image processing techniques and CNNs compare in terms of accuracy, reliability, and scalability for early detection and classification of Alzheimer's disease, particularly when applied to large datasets of MRI scans?

## 1.5 Scope and Limitations

The scope of this project focuses on utilizing Magnetic Resonance Imaging (MRI) data to enhance the detection and classification of Alzheimer's disease (AD) through two key methodologies: image processing techniques and Convolutional Neural Networks (CNNs). Image processing will classify MRI scans into Alzheimer's and Non-Alzheimer's categories by taking the black and white pixel ratio from the data, while CNNs will further classify these scans into four stages: Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented. The project emphasizes early diagnosis by identifying subtle brain changes and evaluates how both techniques handle large-scale data processing, particularly in dealing with large MRI datasets.

However, this project has some limitations. Its success is heavily dependent on the availability and quality of MRI datasets, and the computational resources required for CNN training could be a bottleneck. Additionally, the generalizability of the models may be limited by the diversity of the datasets, as results might vary across populations. Furthermore, the project focuses solely on MRI data, excluding other diagnostic tools like PET scans or blood tests, which could provide a more comprehensive approach to Alzheimer's diagnosis. A significant limitation of the image processing technique is that it can only classify images into two categories: Alzheimer's (AD) and Non-Alzheimer's (Non-AD), lacking the ability to differentiate between the stages of Alzheimer's disease. In contrast, the CNN methodology is capable of classifying MRI scans into four distinct stages: Non-Demented, Very Mild Demented, Mild Demented, and Moderate

Demented. Despite these limitations, the project aims to significantly contribute to early AD detection and improve patient outcomes through more accurate diagnostic tools.

## 1.6 Organisation of the thesis

This thesis is organized into seven chapters, each addressing different aspects of the research conducted on the early diagnosis of Alzheimer's disease through MRI image processing and Convolutional Neural Networks (CNNs).

The first chapter of the thesis introduces the research problem, focusing on the global impact of Alzheimer's disease and the critical importance of early detection. It explains the motivation for using MRI and advanced machine learning techniques, particularly Convolutional Neural Networks (CNNs), to improve the accuracy of Alzheimer's diagnosis. The research objectives are outlined, which aim to develop a robust diagnostic tool combining image processing and deep learning methods.

The second chapter provides a comprehensive literature review, discussing traditional diagnostic approaches such as cognitive assessments and modern neuroimaging techniques. It highlights the role of MRI in visualizing brain changes and explores how image processing and CNNs have been applied to detect Alzheimer's. This chapter identifies the research gaps and limitations in current methods, such as the challenges of handling large datasets and improving generalization across populations.

The third chapter details the methodology, focusing on the design of the image processing pipeline and CNN model. It describes the data collection process, MRI pre-processing techniques, and the experimental setup for training and evaluating the deep learning model. This chapter explains the approach taken to achieve accurate classification of Alzheimer's disease stages.

The fourth chapter covers the technical design and implementation of the system. It discusses the application of image enhancement, segmentation, and classification

## Chapter 1. Introduction

---

techniques to MRI data, along with the architecture of the CNN model. The chapter explains how these methods were used to classify Alzheimer's disease into different stages and optimize model performance.

In the fifth chapter, the results of the experiments are presented and discussed. The chapter evaluates the performance of the proposed methods, comparing the accuracy and reliability of image processing techniques and CNN models. Key findings are analyzed in relation to the research questions, providing insights into the strengths and limitations of each approach for detecting Alzheimer's disease.

The sixth chapter concludes the thesis by summarizing the main contributions and implications of the study. It provides recommendations for future research, such as integrating multimodal data and improving model efficiency for clinical applications. The chapter emphasizes how the findings contribute to advancing Alzheimer's diagnosis. The final chapter, the appendix, includes supplementary materials such as additional figures, tables, or code relevant to the research but not essential to the main text.

# **Chapter 2**

## **Literature Review**

Researchers and healthcare professionals are increasingly focusing on early detection and intervention strategies for Alzheimer's disease (AD) due to its significant impact on patients, caregivers, and healthcare systems globally. As AD cases continue to rise with the aging population, early diagnosis is crucial for slowing disease progression and improving outcomes. This literature review explores current research, methodologies, and challenges in AD detection, particularly emphasizing the role of Magnetic Resonance Imaging (MRI) in visualizing early structural brain changes, such as hippocampal atrophy. MRI, combined with advanced algorithms like image processing techniques and Convolutional Neural Networks (CNNs), has become a key tool for more accurate and efficient diagnosis. Additionally, it compares the effectiveness of traditional methods with modern machine learning techniques in classifying AD and its stages. A synopsis of literature related to the use of image processing and Convolutional Neural Networks (CNNs) for Alzheimer's disease detection is provided, identifying key research gaps to inform the approach adopted for this study.

## 2.1 Traditional Method

Traditional methods for detecting Alzheimer's disease (AD) mainly focused on clinical evaluations of cognitive decline using various neuropsychological assessments. These included tools such as the Mini-Mental State Examination (MMSE), the Alzheimer's Disease Assessment Scale-Cognitive Subscale (ADAS-Cog), and tests of memory and executive function like the Rey Auditory Verbal Learning Test (RAVLT). Physicians also depended on patient history, observations from caregivers, and assessments of functional abilities to identify difficulties with daily tasks. In the past, diagnoses were often made in the later stages of the disease, when symptoms like severe memory loss, confusion, and challenges with routine activities became more evident [9].

### 2.1.1 Mini-Mental State Examination (MMSE)

The study focuses on the Mini-Mental State Examination (MMSE) as a tool for detecting Alzheimer's disease [10]. It evaluates mental function by assessing tasks related to memory, orientation, language, and spatial skills [11]. For example, one task asks patients to spell the word "WORLD" backwards, and the scoring is based on how accurately the letters are recalled in reverse order. Using the line method for scoring, points are given depending on the number of correct sequences. This task helps in assessing a patient's attention and cognitive flexibility, both crucial for diagnosing Alzheimer's disease. It has limitations, including reduced sensitivity to early-stage Alzheimer's disease, a ceiling effect that can lead to false negatives in higher-functioning individuals, and biases related to education, cultural background, or language, making it insufficient for complex cognitive assessments [12].

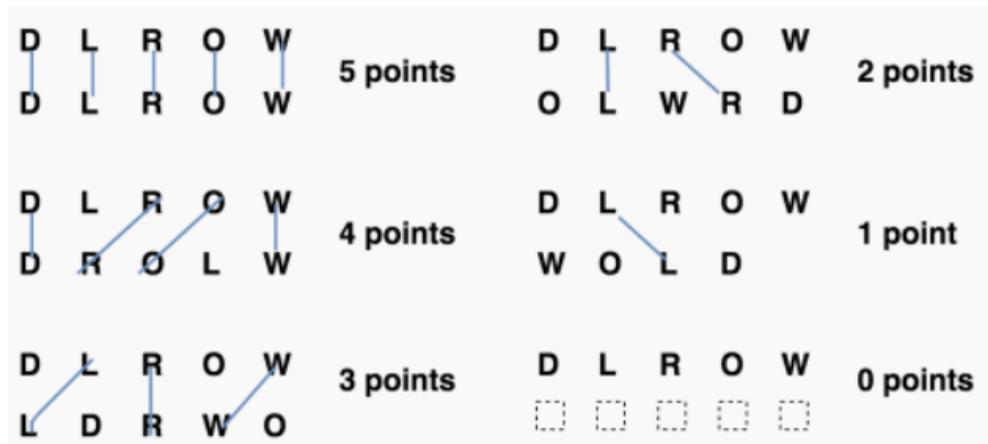


Figure 2.1: Scoring the WORLD Task in the MMSE Using the Line Method [10]

### 2.1.2 The Alzheimer's Disease Assessment Scale-Cognitive Subscale (ADAS-Cog)

The Alzheimer's Disease Assessment Scale-Cognitive Subscale (ADAS-Cog) is a widely used cognitive tool that assesses memory (50%), language (28%), praxis (14%), and command understanding (8%) [13] across 11 tasks [14]. While effective, it has limitations, including low sensitivity for detecting mild cognitive impairment and early-stage Alzheimer's [15]. Additionally, its assessment of memory and executive functions is limited, and the scores can be influenced by socio-educational factors, potentially affecting the reliability of results across diverse patient populations.

Subtest	Score Range	Assessed Cognitive Function
<b>Word Recall</b>	0-10	Memory
<b>Naming Objects and Fingers</b>	0-5	Language
<b>Following Commands</b>	0-5	Praxis
<b>Constructional Praxis</b>	0-5	Spatial Skills
<b>Ideational Praxis</b>	0-5	Motor Planning
<b>Orientation</b>	0-8	Temporal-Spatial Awareness
<b>Word Recognition</b>	0-12	Memory (Recognition)
<b>Spoken Language Ability</b>	0-5	Communication Skills

Table 2.1: Subtests and Cognitive Domains in ADAS-Cog)

### **2.1.3 Rey Auditory Verbal Learning Test (RAVLT)**

The Rey Auditory Verbal Learning Test (RAVLT) is a neuropsychological assessment designed to evaluate cognitive functions such as short-term memory, long-term memory, learning ability, and recognition by having participants recall a list of words over multiple trials. After being presented with the list, participants are asked to recall as many words as possible immediately and again after a delay [16]. However, the test has limitations, including its susceptibility to practice effects, where repeated administration may lead to improved scores due to task familiarity rather than cognitive improvement.

## **2.2 Role of Neuroimaging**

Magnetic Resonance Imaging (MRI) is a critical tool for diagnosing Alzheimer's disease early by offering detailed, non-invasive images of the brain's structural changes [17]. One of the most significant early indicators of Alzheimer's is hippocampal atrophy, where the hippocampus—a key region for memory and learning—begins to shrink. MRI enables precise measurement of this shrinkage, even before severe cognitive symptoms appear. This is crucial for diagnosing Alzheimer's in its early stages, allowing for earlier interventions that can slow the disease's progression. Unlike cognitive tests, which may not capture subtle brain changes, MRI can reveal degeneration in regions like the hippocampus, temporal lobes, and cortex, which are affected in the initial phases of the disease [18]. Furthermore, MRI also helps track the disease's progression and assess the effectiveness of treatments over time, making it a powerful tool for both diagnosis and long-term management of Alzheimer's disease. By providing a clear picture of structural changes, MRI enhances the ability to predict the onset and trajectory of Alzheimer's, contributing to improved patient outcomes and care. Magnetic Resonance Imaging (MRI) not only reveals hippocampal atrophy, but it also highlights changes

in gray and white matter in the brain, which are important in Alzheimer's detection [19]. Gray matter, containing most of the brain's neuronal cell bodies, decreases in volume in key regions such as the hippocampus and cortex as Alzheimer's progresses. This leads to cognitive decline, particularly in memory and decision-making. White matter, responsible for connecting different brain regions, also deteriorates, affecting the brain's communication networks. Tracking these changes with MRI helps in early diagnosis and monitoring disease progression [20].

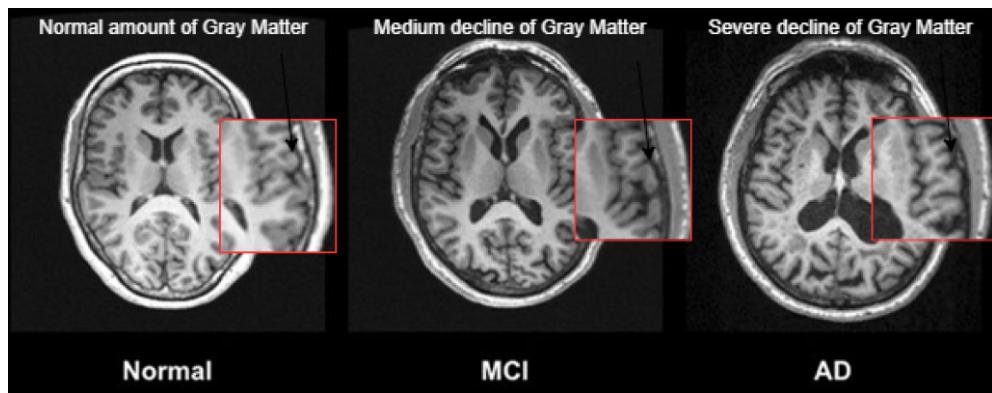


Figure 2.2: MRI scans showing increasing gray matter loss from normal to AD [21]

## 2.3 Image Processing Techniques

In Alzheimer's Disease detection, image processing plays a crucial role in the analysis of brain scans, primarily MRI images, to identify structural and functional abnormalities. AD is characterized by progressive brain atrophy, particularly in regions such as the hippocampus and cerebral cortex [22]. Classical image processing techniques, including image preprocessing, segmentation, and feature extraction, are essential for isolating these brain structures and quantifying changes over time.

Preprocessing techniques like noise reduction are used to enhance the quality of MRI images by removing irrelevant structures and minimizing noise. Segmentation techniques, such as thresholding, are applied to separate brain tissues like gray matter, white

matter, and cerebrospinal fluid. This helps in identifying critical structural changes, such as hippocampal atrophy, which are key indicators of AD progression. Additionally, feature extraction techniques enable researchers to quantify subtle variations in brain tissues that are often linked to the early onset of Alzheimer's.

### **2.3.1 Preprocessing Techniques**

Preprocessing in image analysis involves steps such as noise reduction, intensity normalization, and background removal to prepare medical images for more accurate feature extraction and analysis. Techniques like Gaussian Blur help in smoothing out irrelevant details while preserving critical anatomical structures, which is essential in detecting early signs of Alzheimer's disease.

#### **2.3.1.1 Gaussian Blur**

Gaussian Blur is one of the best image preprocessing techniques in medical imaging, particularly for Alzheimer's disease detection, due to its ability to effectively reduce noise while preserving important structural details. Noise, which can arise from variations in scanning devices or patient movement, can interfere with the accuracy of subsequent image analysis, such as segmentation or edge detection. By using a filter to smooth out fluctuations in pixel intensity, Gaussian Blur solves this problem and produces a cleaner, more uniform image that is ready for analysis [23].

One of the key advantages of Gaussian Blur is its effectiveness in enhancing the performance of edge detection algorithms, like the Canny edge detector, which is critical for isolating brain structures in MRI scans. Only relevant edges like those of brain tissues are kept after the Gaussian kernel smooths the image and lessens the effect of spurious edges brought on by noise. The exact identification of brain regions, such as the hippocampus and cortical areas, is critical for monitoring atrophy and tissue loss



Figure 2.3: Effect of Gaussian Blur: Left shows noise, right shows noise reduction after Gaussian blur [24]

in the context of Alzheimer's disease, making this preprocessing step very important. By selectively smoothing the image without significantly blurring key features, Gaussian Blur maintains the necessary balance between noise reduction and detail preservation, making it an ideal preprocessing technique in medical imaging workflows.

### 2.3.1.2 Skull Stripping

A method of preprocessing medical images that is especially utilised for brain MRI scans and involves removing non-brain tissues from the image, such as the eyes, scalp, and skull [25]. By isolating the brain tissue, this procedure makes it possible to analyse brain structures more precisely, particularly when dealing with neurodegenerative illnesses like Alzheimer's.

Skull stripping, which concentrates solely on the anatomy of the brain, improves later processes like segmentation, feature extraction, and classification by removing the surrounding non-brain features. This is important for studying atrophy and structural alterations in Alzheimer's disease research, which looks at areas like the cerebral cortex and the hippocampus.



Figure 2.4: Brain stripping on brain [26]

### 2.3.2 Interpolation technique

Interpolation techniques are essential for resizing images, particularly in medical imaging, as they estimate pixel values between known data points. These techniques help maintain image integrity during transformations like scaling, rotating, or zooming by predicting new pixel values to fill gaps. There are several commonly used methods: nearest-neighbor interpolation, which assigns the closest pixel value to a new position but may create jagged edges; bilinear interpolation, which smooths images by considering adjacent pixels; and bicubic interpolation, which uses 16 neighboring pixels for a more refined, smoother output. These methods are vital for maintaining high-quality visuals in processes like MRI image resizing and analysis, ensuring accuracy in diagnostic applications.

#### 2.3.2.1 Nearest-neighbor interpolation

Nearest-Neighbor Interpolation is one of the simplest and fastest techniques for resizing images, particularly in medical and real-time applications. It works by selecting the pixel value closest to the target point to fill in missing data during resizing or transformation [27]. This simplicity makes it computationally efficient, making it suitable for scenarios where speed is a priority. However, it can produce blocky or pixelated images, especially when scaling up, as no smoothing or interpolation between pixel values is applied. As a

result, this method can introduce visible artifacts or jagged edges, which are undesirable in tasks requiring high-quality visual output. Despite its limitations, nearest-neighbor interpolation remains useful in applications where speed, rather than precision, is the main requirement. For example, it is often utilized in quick prototyping, binary images, or when resizing categorical data where the precision of pixel values isn't as critical.

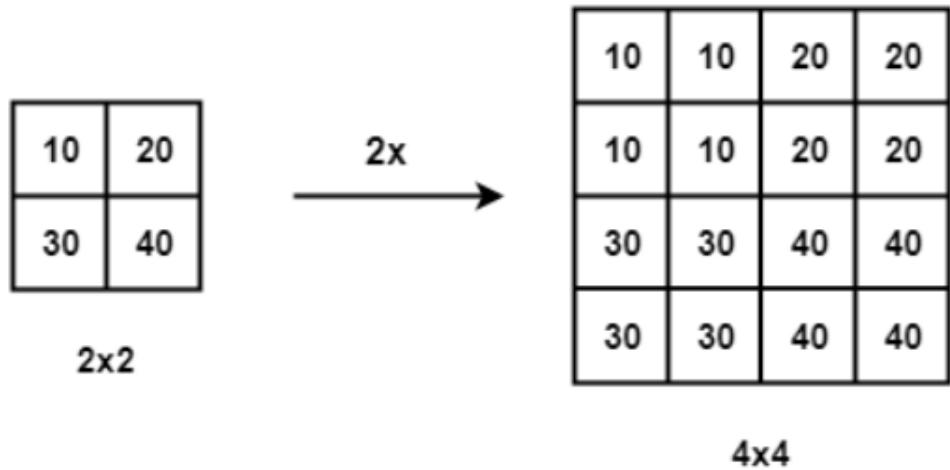


Figure 2.5: Image Upsampling:  $2 \times 2$  to  $4 \times 4$  Matrix Using Nearest Neighbor Interpolation [28]

### 2.3.2.2 Bilinear interpolation

Bilinear interpolation is a resampling technique used in image processing to estimate new pixel values when resizing or transforming an image. It considers the closest two-by-two neighborhood of known pixel values surrounding the new pixel and applies a weighted average to compute the intensity of the unknown pixel. This method results in smoother images compared to nearest-neighbor interpolation, as it accounts for both horizontal and vertical gradients.

The interpolation process involves linearly interpolating first along one axis and then along the other, ensuring smooth transitions. This technique is especially effective for scaling and rotating images, where maintaining smoothness and avoiding jagged edges is essential. However, while bilinear interpolation provides a good balance between

computational efficiency and image quality, it may still blur fine details when scaling images up significantly [29].

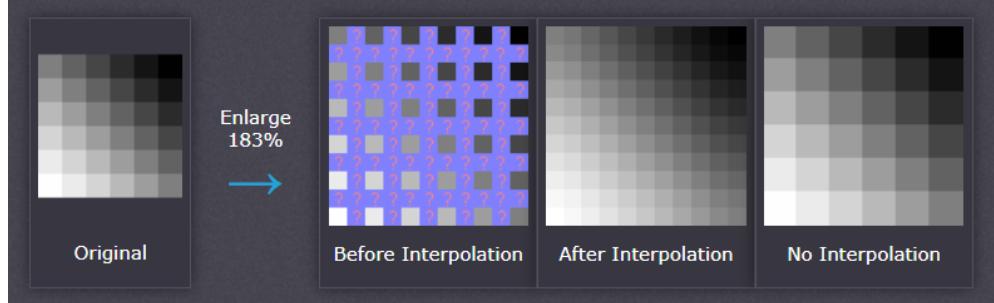


Figure 2.6: Interpolation: Before, After, and No Interpolation Effects [30]

### 2.3.2.3 Bicubic interpolation

Bicubic interpolation is a widely used image scaling technique that improves upon bilinear interpolation by considering the values of 16 surrounding pixels, as opposed to just 4. This method generates smoother, higher-quality images by applying a more complex mathematical function to estimate the value of new pixels based on the distances and intensities of neighboring pixels. As a result, it reduces blockiness and preserves finer details, especially when enlarging images.

Compared to simpler methods like nearest-neighbor and bilinear interpolation, bicubic interpolation offers a more visually appealing result with fewer artifacts. It maintains the continuity of gradients and softens the transition between pixels, making it particularly suitable for medical imaging and other applications requiring very high accuracy [31]. However, it is computationally more intensive, which can be a trade-off when processing large datasets or requiring real-time results.

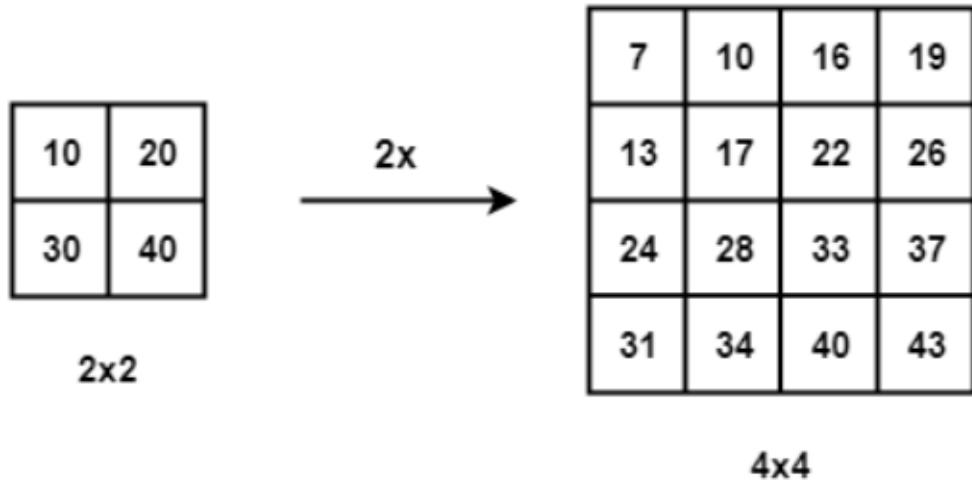


Figure 2.7: 2x2 to 4x4 Upscaling Using Bicubic Interpolation [32]

### 2.3.3 Thresholding Techniques

By transforming a greyscale image into a binary image, the fundamental image processing technique known as thresholding is utilised to divide an image into discrete sections. Pixel intensities are compared to a preset threshold value in order for it to function. Foreground pixels are identified as having intensity values above the threshold, whereas background pixels are classed as having intensity values below the threshold. One value is used for the entire image in simple global thresholding, however this can not work well in photographs with uneven lighting. Bimodal histograms are well-suited to techniques like Otsu's approach, which automatically generates an optimal threshold by reducing intra-class variation between the two classes. Adaptive thresholding, on the other hand, effectively manages photos with varying lighting by calculating distinct thresholds for each part of the image. Medical imaging, object detection, and document image analysis frequently use thresholding techniques to differentiate objects of interest from the background.

### 2.3.3.1 Fixed (Global) Thresholding

Fixed global thresholding is a basic image processing technique used to segment an image into binary format, where each pixel is classified as either foreground or background [33]. The process involves selecting a single threshold value, which is applied globally across the entire image. Any pixel with a value above the threshold is set to one intensity (white), while those below are set to another (black). This method is simple and computationally efficient but can struggle with images where lighting varies or contrast is uneven.

One of the major limitations of fixed global thresholding is its sensitivity to changes in illumination. If an image has regions with varying brightness, a single threshold might not be able to effectively distinguish the foreground from the background in all parts of the image. This technique works best on well-illuminated, high-contrast images but often needs to be paired with more adaptive methods for more complex scenarios like medical imaging, where consistent lighting conditions are not always guaranteed.

The thresholding operation can be expressed as Equation 2.1 [34]:

$$T(x, y) = \begin{cases} 255 & \text{if } I(x, y) > T \\ 0 & \text{if } I(x, y) \leq T \end{cases} \quad (2.1)$$

Where  $T$  is the threshold value, and  $(x, y)$  are the coordinates of a pixel in the image.

### 2.3.3.2 Otsu's Thresholding

Otsu's thresholding is a widely used method in image segmentation, particularly when dealing with images that have a bimodal histogram, meaning two distinct peaks representing the foreground and background pixel intensities. The algorithm, introduced by Nobuyuki Otsu in 1979 [35], calculates the optimal threshold value by minimizing the intra-class variance, or the variance within the foreground and background classes.

In simpler terms, it seeks the threshold that best separates these two regions in terms of pixel intensity, making it highly effective for images where the foreground and background have distinct intensity distributions.

One of the key advantages of Otsu's method is that it is fully automated and does not require manual selection of the threshold value, which is particularly useful for large datasets or in cases where consistent segmentation is needed across multiple images. Otsu's method works best for images with a clear intensity separation between the foreground and background, but it may perform poorly when the image has uneven lighting or when the intensity distributions of the foreground and background overlap significantly. To overcome these limitations in more complex images, Otsu's method is often combined with preprocessing techniques like Gaussian blur or used in conjunction with adaptive thresholding for better results.

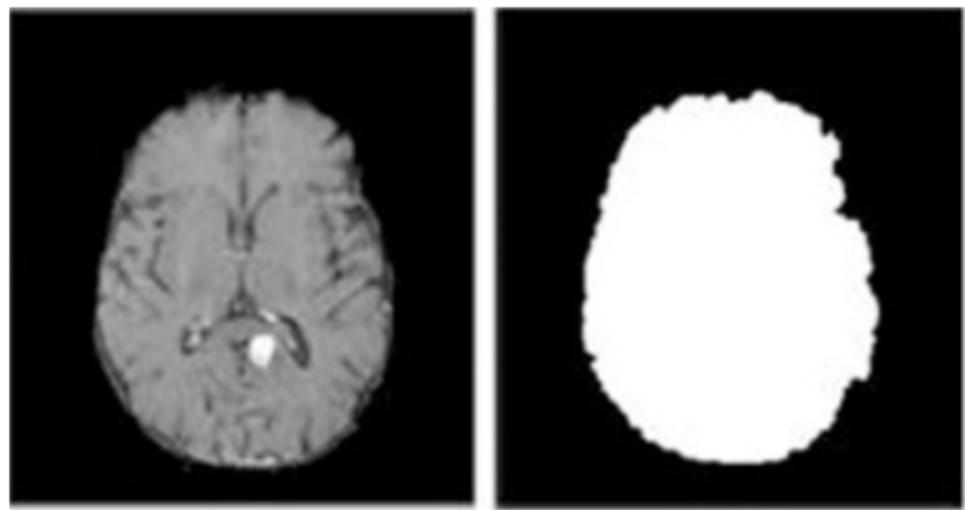


Figure 2.8: Otsu thresholding on the image [36]

#### 2.3.3.3 Adaptive thresholding

Adaptive thresholding is a technique used in image processing to handle images with non-uniform lighting or varying brightness levels. Unlike global thresholding, which applies a single threshold value across the entire image, adaptive thresholding calculates

the threshold for small regions (local neighborhoods) within the image. This makes it particularly useful for images where different parts are lit unevenly, such as medical images or documents with shadows [37].

Adaptive thresholding is especially advantageous when segmenting images where the foreground and background are not easily distinguishable under global thresholding methods due to varying lighting conditions. For instance, in document image processing, where shadows or different background textures are present, adaptive thresholding can effectively isolate text from the background. Similarly, in medical imaging, it helps in extracting features from complex images where brightness inconsistencies may distort analysis using simpler thresholding methods. While adaptive thresholding is computationally more expensive than global thresholding, its ability to handle varying illumination makes it a powerful tool in many real-world applications.

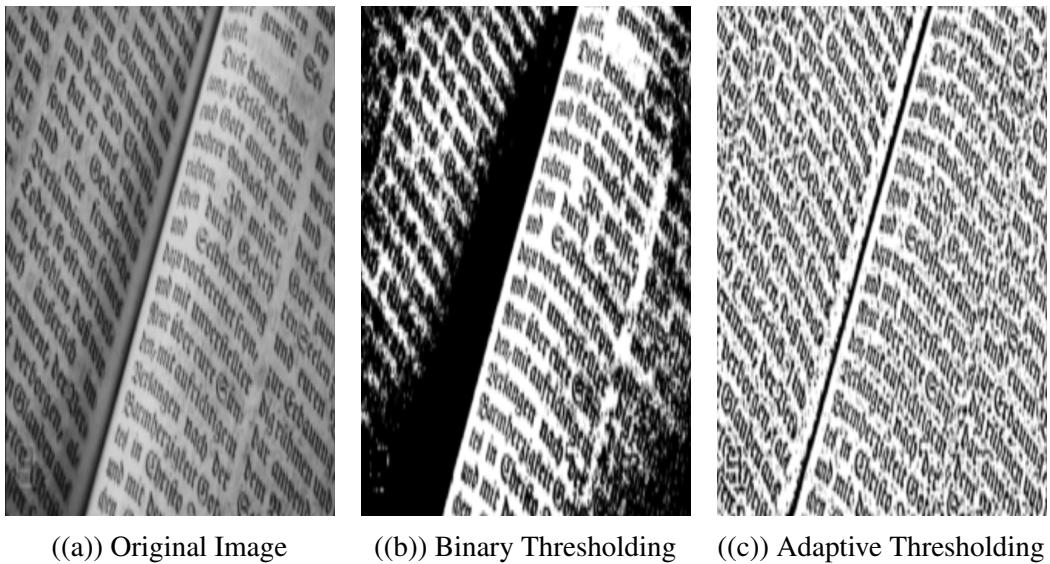


Figure 2.9: Comparison of Threshholding Techniques [38]

### 2.3.4 Edge detection techniques

Edge detection is a crucial image processing technique used to identify and highlight boundaries within an image. It works by detecting areas where the intensity of pixels

changes significantly, typically signifying the edges of objects. This helps to outline the shape of objects, identify features, and improve object recognition in various applications. Edge detection is fundamental in fields like computer vision, medical imaging, and machine learning. Common algorithms include Sobel and Canny each offering different methods for detecting edges in an image.

#### 2.3.4.1 Sobel detector

The Sobel detector is a widely-used edge detection method in image processing that calculates the gradient of image intensity. By applying two filters (horizontal and vertical) [39], the Sobel operator highlights regions of significant intensity changes, often representing the edges of objects. The result is a gradient magnitude image that shows the edges prominently.

The Sobel operator consists of two  $3 \times 3$  convolution kernels: the horizontal kernel, which responds maximally to edges running vertically, and the vertical kernel, which responds to edges running horizontally. When these kernels are convolved with the input image, they produce two gradient images that represent the rate of intensity change in both orientations. The gradients from these two directions are then combined to compute the gradient magnitude, resulting in a single image that emphasizes the edges. This gradient magnitude image is particularly useful as it clearly delineates the boundaries of objects, making it easier for subsequent image analysis tasks. The Sobel detector is favored for its simplicity and effectiveness, making it a fundamental tool in image processing workflows. Its ability to produce a clear representation of edges contributes significantly to the overall understanding and interpretation of the visual information contained within images.

One of the advantages of the Sobel detector is its simplicity and efficiency, making it suitable for real-time applications. It smoothens the image slightly through Gaussian filtering, reducing noise, but can still struggle with finer details and complex edges.

Sobel is frequently used in applications like computer vision, medical imaging, and object recognition, where edge information is essential for feature extraction.

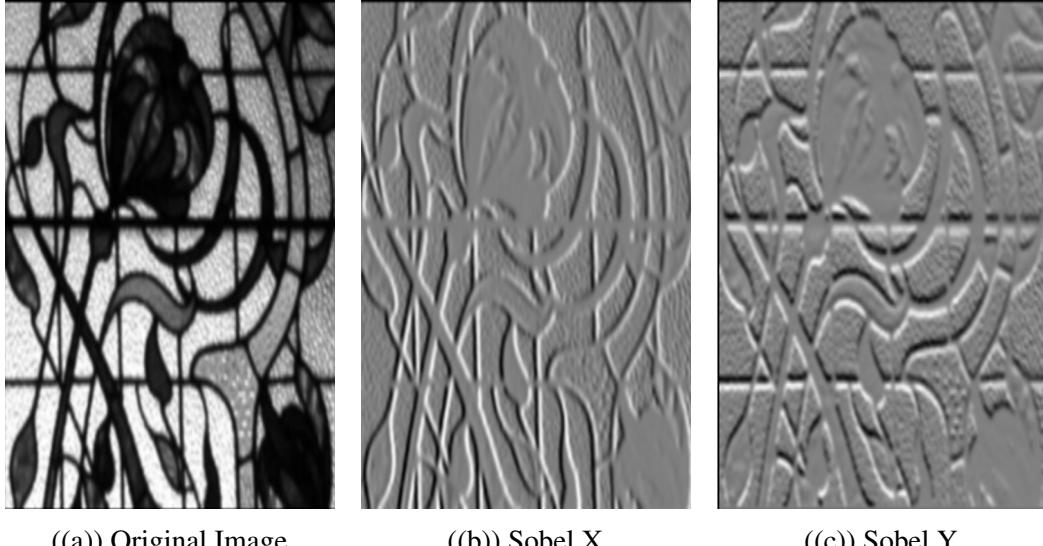


Figure 2.10: Original Image with Sobel X and Sobel Y Gradients [38]

#### 2.3.4.2 Canny edge Detection

Canny edge detection stands out from other edge detection techniques due to its multi-stage process, which balances edge detection accuracy with noise reduction. Unlike simpler techniques like Sobel or Prewitt, which rely solely on first-order derivatives to detect edges, the Canny algorithm starts by applying a Gaussian filter to smooth the image, thereby reducing the impact of noise. This step helps to prevent the detection of false edges caused by random noise or small variations in pixel intensity. Following the smoothing step, the algorithm computes the intensity gradient at each pixel, detecting the strength and direction of edges [40].

Canny edge detection uses non-maximum suppression and double thresholding[41]. Non-maximum suppression ensures that the detected edges are thin and accurate by keeping only the local maxima of the gradient magnitudes. The double thresholding

technique further enhances precision by classifying pixels into strong, weak, or non-edges, based on two threshold values. The final stage, edge tracking by hysteresis, ensures continuity by connecting weak edges that are adjacent to strong ones. This combination of techniques allows Canny edge detection to produce clear, continuous edges with minimal noise interference, making it highly effective in applications such as medical imaging, computer vision, and object recognition.

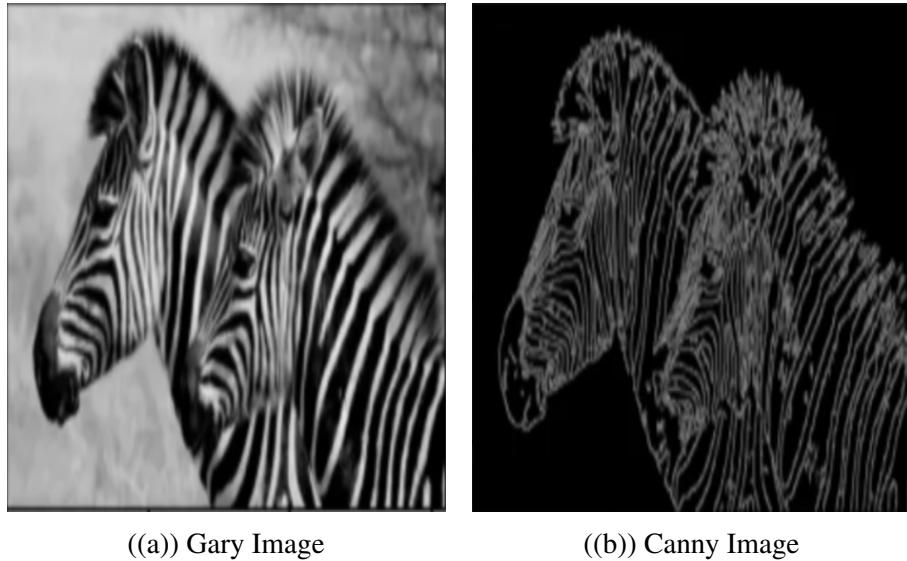


Figure 2.11: Comparison of Gray Image and Canny Edge Detection [38]

## 2.4 Convolution neural network techniques

Convolutional Neural Networks (CNNs) have transformed medical imaging by making it possible to automatically extract features from complicated picture data, which has greatly enhanced the ability to identify and categorise neurological conditions like Alzheimer's. When it comes to MRI scan analysis, CNNs are especially good at spotting minor structural changes like cortical thinning and hippocampus shrinkage, which are early indicators of Alzheimer's disease progression. Because CNNs can learn hierarchical features from raw pixel data, manual feature engineering is no longer

necessary, which improves illness diagnosis speed and accuracy.

### 2.4.1 VGG16

VGG16 is a well-known CNN architecture that consists of 16 layers, primarily using convolutional and fully connected layers [42]. It was one of the first architectures to show effectiveness of using deep networks for image classification, and it has been used in various medical imaging tasks, including Alzheimer’s disease detection.

While VGG16 is a robust model, it is computationally expensive due to the large number of parameters and layers. This makes it less suitable for real-time applications or when working with limited computational resources. MobileNetV2, on the other hand, is designed for efficiency, using depthwise separable convolutions to reduce the number of computations significantly. VGG16 lacks the efficient feature extraction mechanism that MobileNetV2 offers through its bottleneck layers and inverted residuals. This makes it slower to train and more memory-intensive, particularly when handling large datasets like MRI scans in Alzheimer’s disease detection. Additionally, VGG16 is prone to overfitting on smaller datasets, unless extensive regularization is applied [43].

In this strategy, the VGG-16 network was used as an encoder with pretrained weights, which were essential given the dataset’s limitations. VGG-16, trained on common image classification tasks, first learns basic image features like edges and contours in early layers. Later layers identify more complex objects. This transfer learning approach allows the model to apply these learned features to satellite images, speeding up the training process by not starting from scratch. The architecture consists of five blocks/pooling layers, each containing two convolutional layers reducing the feature map dimensions [44].

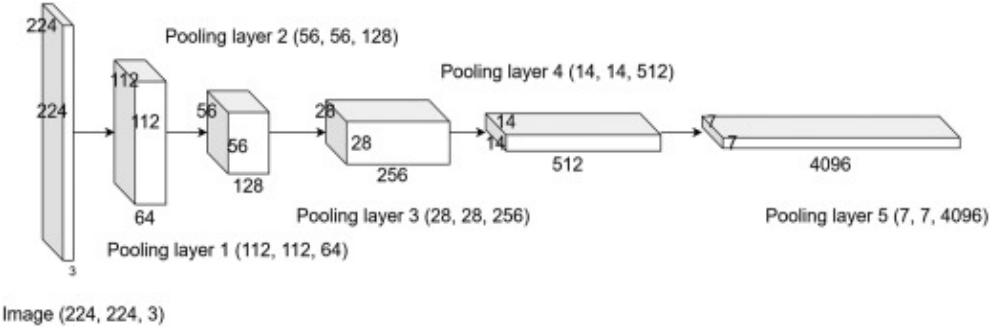


Figure 2.12: The dimensions of each block are reduced post-pooling to condense information [44]

### 2.4.2 AlexNet

AlexNet is another early CNN architecture that consists of five convolutional layers and three fully connected layers [45]. It was revolutionary for its time and demonstrated the potential of deep learning in image classification tasks, but it is now considered less powerful compared to more recent architectures.

AlexNet had limitations due to hardware constraints at the time, leading to the use of two GPUs to enhance its learning ability. Overfitting was a major issue with its depth, although the use of techniques like local response normalization, ReLU activation, and overlapping subsampling helped mitigate this. Despite these solutions, challenges like vanishing gradients and large filter sizes in earlier layers remained concerns, although AlexNet significantly influenced the development of modern CNN architectures [46]

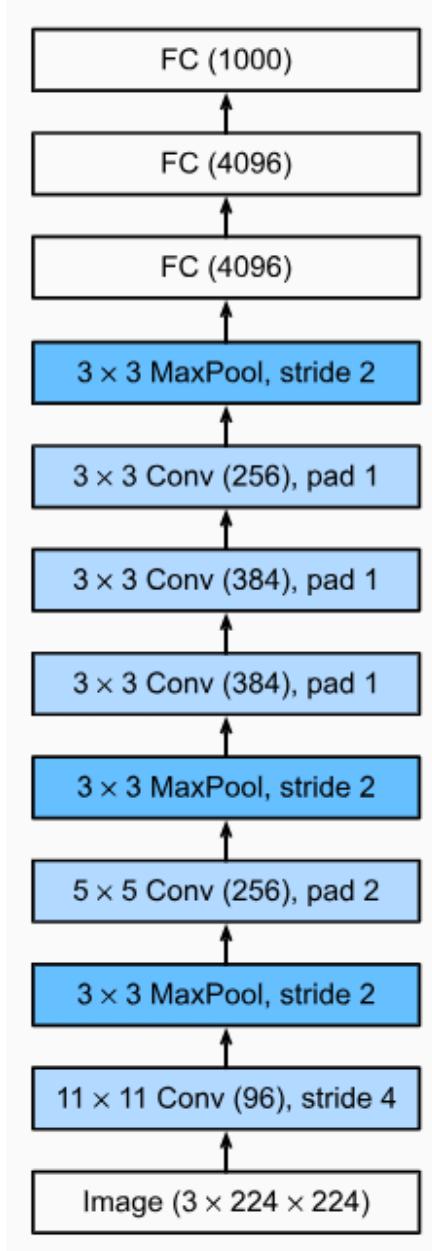


Figure 2.13: AlexNet architecture [45]

### 2.4.3 MobileNet V2

MobileNet V2 is a state-of-the-art convolutional neural network (CNN) specifically designed for mobile and embedded vision applications, achieving an optimal balance between computational efficiency and accuracy. This architecture is an evolution of

MobileNet V1 and incorporates several key innovations that enhance its performance. One of the primary advancements is the introduction of inverted residuals, which help to efficiently pass information through the network while maintaining a compact architecture. These residual connections allow for better gradient flow during training, addressing some of the challenges associated with deep networks [47].

In addition to inverted residuals, MobileNet V2 utilizes linear bottlenecks, which minimize information loss during the downsampling process. This design choice effectively reduces the model's computational complexity without compromising its ability to perform tasks such as image classification or object detection. The architecture starts with a conventional convolution layer that serves as the input, followed by a series of inverted residual blocks, each designed to refine features and enhance learning efficiency. The network culminates in a fully connected layer that performs classification based on the extracted features [47].

MobileNet V2 also offers the flexibility to adjust both the width and resolution of the network, allowing it to be tailored to the specific requirements of various applications. By modifying these parameters, developers can optimize the model for specific hardware constraints or performance targets, making it particularly suitable for deployment on mobile devices with limited computational resources. This adaptability, combined with its efficient architecture, has made MobileNet V2 a popular choice for a wide range of applications in mobile and embedded vision, including real-time image processing, augmented reality, and robotics. Its performance and efficiency make it a vital tool for advancing the capabilities of mobile vision systems, thereby contributing to the growing field of artificial intelligence on portable platforms [47].

With improvements over its predecessor, MobileNet V2 delivers enhanced accuracy with fewer parameters and a lower computational cost, making it highly effective for mobile devices. Its ability to scale and its lightweight design have made it popular in real-time applications like object detection and image classification in constrained

environments, maintaining high performance while conserving power and processing resources [47].

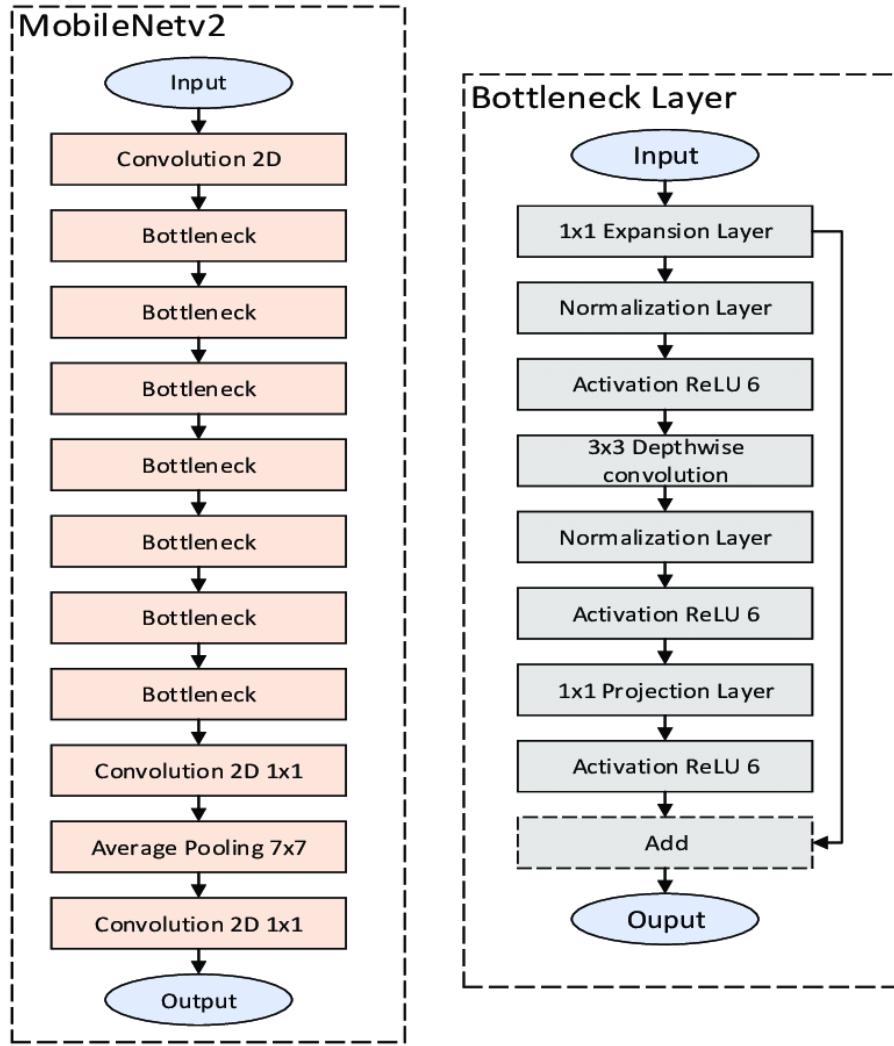


Figure 2.14: MobileNet V2 architecture [48]

## 2.5 Research Gaps and Implications

Current research in Alzheimer's detection is limited by dataset diversity, as many models rely on region-specific data that do not reflect a broad population. Expanding to larger, more diverse datasets is necessary to improve model generalizability. Additionally,

most approaches focus on MRI data alone, neglecting the integration of multimodal data, such as blood tests or PET scans, which could enhance diagnostic accuracy. Furthermore, CNN models, although powerful, are computationally expensive, and there is a significant need to develop lightweight, scalable models for real-time use in clinical settings with limited resources.

## 2.6 Summary

This chapter presents a comprehensive overview of the traditional and modern techniques used for detecting Alzheimer's disease. It highlights how traditional cognitive assessment tools like MMSE, ADAS-Cog, and RAVLT have limitations in detecting early stages of Alzheimer's. The role of MRI in visualizing structural brain changes, such as hippocampal atrophy, is also discussed, emphasizing its importance in early diagnosis. The chapter then explores the impact of advanced image processing techniques and convolutional neural networks (CNNs), including architectures like VGG16, AlexNet, and MobileNet V2, in automating the detection and classification of Alzheimer's from MRI scans. These models demonstrate improved accuracy but also pose challenges, such as high computational cost and the need for large, diverse datasets. Additionally, the research gaps discussed focus on the need for integrating multimodal data sources, such as PET scans and blood tests, and the development of lightweight models suitable for real-time clinical applications. Overall, this chapter underscores the ongoing evolution in Alzheimer's detection methodologies, bridging the gap between traditional approaches and cutting-edge AI-driven techniques, while also pointing out the need for further advancements in data diversity, computational efficiency, and multimodal integration.

# **Chapter 3**

## **Methodology**

### **3.1 Image Processing Method**

The methodology for this project involves an extensive combination of image processing techniques and deep learning models through Convolutional Neural Networks (CNNs). Initially, the MRI scans undergo preprocessing to standardize image dimensions and improve quality. Bicubic interpolation is applied for resizing images, ensuring consistency without compromising fine details. Gaussian Blur is used to reduce noise and improve the clarity of the images. This step is critical for eliminating minor image distortions that could interfere with further analysis.

Next, Canny edge detection is employed to identify the structural boundaries of the brain by detecting areas where the intensity of pixels changes rapidly. This process effectively outlines key anatomical features such as the hippocampus, which is essential for Alzheimer's diagnosis. To further refine the results, contour detection is used to isolate and highlight these significant brain structures. This technique emphasizes critical features that might indicate neurodegeneration.

Otsu's thresholding technique is a powerful and widely-used method in image processing, particularly for segmenting brain tissue from the background in MRI images.

### Chapter 3. Methodology

---

By analyzing the histogram of pixel intensities, Otsu's method automatically calculates an optimal threshold value that minimizes intra-class variance while maximizing inter-class variance. This ensures that pixels representing different tissue types are accurately categorized into foreground (the brain tissue) and background (the non-tissue areas). The ability to automatically determine the threshold value is crucial in medical imaging, where manual thresholding can be subjective and prone to error. This automatic segmentation significantly enhances the quality of the analysis by providing clear delineations between tissue types, thereby allowing for more accurate measurements and assessments in further studies.

In addition to Otsu's thresholding, morphological operations play a vital role in refining the segmentation results. Dilation and closing operations are used to process the binary images obtained after thresholding, helping to close small gaps in the segmented edges and connect fragmented regions. Dilation expands the boundaries of the foreground objects, making them more cohesive, while closing fills in small holes within those objects. This process results in smoother boundaries, which are essential for accurately defining brain structures in the images. By applying these morphological operations, the quality of segmentation is further enhanced, allowing for more reliable feature extraction and analysis in subsequent steps of the imaging process. Overall, the combination of Otsu's thresholding and morphological operations creates a robust framework for effective image segmentation in medical imaging applications.

For the final stages, pixel counting is applied to the binary images. This allows for a quantitative analysis of the affected brain regions, aiding in the comparison of healthy versus diseased tissues. Throughout the image processing pipeline, the results are documented using Excel reports, which record key data points like pixel distributions, ensuring a structured analysis.

## Chapter 3. Methodology

---

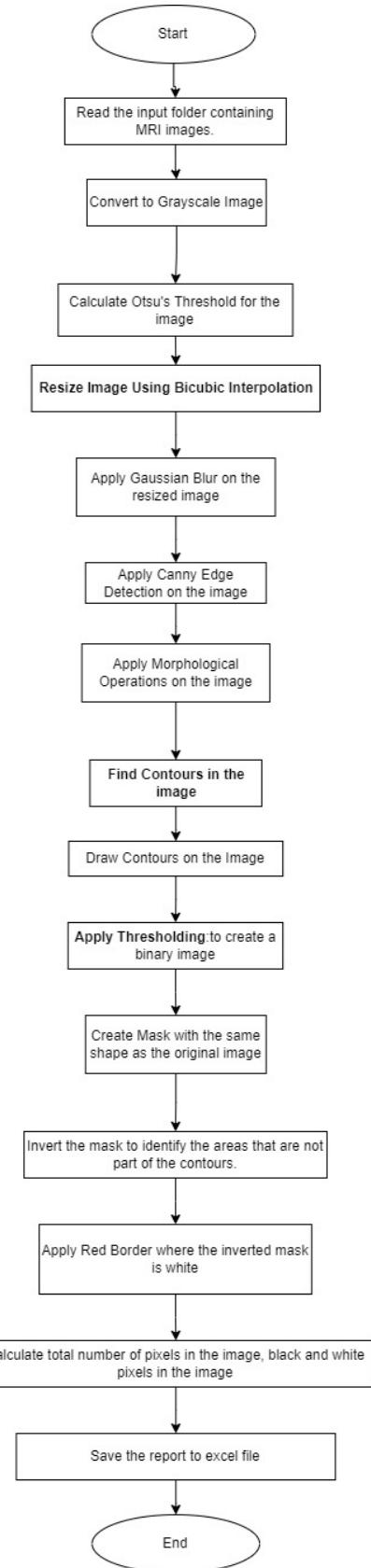


Figure 3.1: Methodology for Image Processing

## 3.2 Convolution Neural Network Method

The second phase utilizes convolutional neural networks (CNNs), specifically MobileNetV2, to perform deep learning on preprocessed MRI images. MobileNetV2, optimized for mobile and embedded applications, balances accuracy and computational efficiency. The model is trained to classify MRI images into four categories of Alzheimer's disease: Mild, Moderate, Very Mild, and Non-Demented. This classification enables early detection and diagnosis, supporting timely intervention and care.

To optimize image handling during training, preprocessing functions from MobileNetV2 are utilized to standardize the input images, allowing the model to learn effectively. To enhance dataset variability and improve model robustness, various data augmentation techniques are employed, including rotation, zooming, and horizontal flipping. These transformations help the model generalize better across different scenarios, increasing the diversity of the training dataset. This approach makes the model more resilient to variations, ensuring better performance on unseen data. Consequently, it helps mitigate overfitting and enhances the overall effectiveness of the CNN, leading to more reliable predictions in real-world applications of Alzheimer's disease diagnosis.

The integration of image processing techniques and CNNs improved the detection and classification of Alzheimer's disease in this project. While the image processing approach categorized brain scans into "Demented" and "Non-Demented," it lacked the precision for further differentiation. In contrast, CNNs enhanced classification by identifying stages of Alzheimer's, such as Mild, Very Mild, Moderate and Non-Demented. This classification aids in understanding disease progression and enables tailored interventions. Furthermore, CNNs demonstrated greater accuracy and reliability in detecting subtle brain changes, making them valuable tools for early diagnosis and disease management.

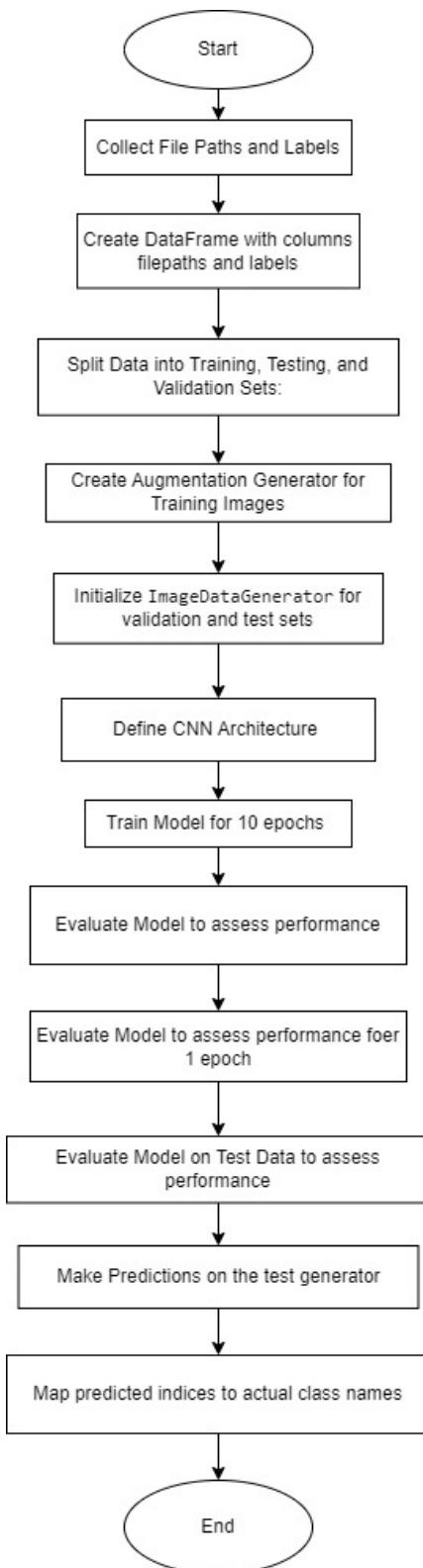


Figure 3.2: Methodology for Image Processing

### 3.3 Summary

This project employs a combination of advanced image processing techniques and deep learning models to classify MRI images for the early detection of Alzheimer's disease. The first phase involves preprocessing the MRI scans using methods like bicubic interpolation, Gaussian Blur, and Canny edge detection to enhance image quality and emphasize critical brain structures. Techniques such as Otsu's thresholding and morphological operations are applied to accurately segment brain tissue, improving the analysis of neurodegenerative features. Quantitative analysis through pixel counting further aids in comparing healthy and diseased brain regions.

In the second phase, convolutional neural networks (CNNs), particularly MobileNetV2, are used to classify the preprocessed MRI images into four categories of Alzheimer's: Mild Demented, Moderate Demented, Very Mild Demented, and Non Demented. The convolutional neural networks benefit from data augmentation techniques to increase the variability and robustness of the model, enhancing its generalization and performance. While traditional image processing methods could differentiate between "Demented" and "Non-Demented," convolutional neural networks provide more granular classification, enabling a better understanding of disease progression and facilitating more targeted interventions for Alzheimer's patients.

# **Chapter 4**

## **Design and Implementation**

In designing the experiment for this project, the primary objective was to create a robust framework for classifying Alzheimer's disease stages using MRI data. The experiment involves two key components: image processing and deep learning. Initially, the MRI images were preprocessed using techniques such as resizing, bicubic interpolation, and noise reduction through Gaussian Blur. This ensured uniformity in image size and quality across the dataset. Next, image segmentation was carried out using Canny edge detection and Otsu's thresholding to isolate key brain structures relevant to Alzheimer's disease.

After image preprocessing, the study shifted focus to Convolutional Neural Networks (CNNs) to enhance classification accuracy. A pre-trained CNN model was fine-tuned to classify the MRI images into four categories: Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented. The deep learning model was able to provide a more refined classification, which was crucial for understanding the various stages of Alzheimer's progression. In comparison, traditional image processing could only differentiate between Demented and Non-Demented categories.

For the dataset, the MRI images were obtained from an established Alzheimer's disease dataset, which contains images categorized into four stages. These were divided into

training, validation, and test sets to ensure the model's performance was evaluated on unseen data [49].

## 4.1 Image Processing Technique

In this project, image processing techniques were applied to MRI scans, allowing the classification of images into two categories: Demented and Non-Demented. This classification was based on the ratio of black and white pixels after applying techniques like Otsu's thresholding and edge detection. If the number of black pixels was greater than white, the image was categorized as Demented, indicating significant structural loss in the brain, a common symptom of Alzheimer's. On the other hand, when the white pixels exceeded or equaled black, the image was classified as Non-Demented, reflecting a healthier brain structure. These pixel ratios highlight key differences in brain tissue density, where demented brains typically show more atrophy, resulting in greater black pixel areas post-thresholding. This simple yet effective approach helped to identify major brain structural changes linked to Alzheimer's disease.

### 4.1.1 Input Data

The input data used for the image processing classification involved MRI scans of Alzheimer's patients. These images were processed to distinguish between Demented and Non-Demented categories. MRI scans provide high-resolution, structural views of the brain, allowing the identification of atrophy in regions like the hippocampus and cortex. By applying techniques such as edge detection and thresholding, the scans were analyzed for major structural differences. This approach enabled the classification of the images based on the visible impact of Alzheimer's on brain tissue [49]

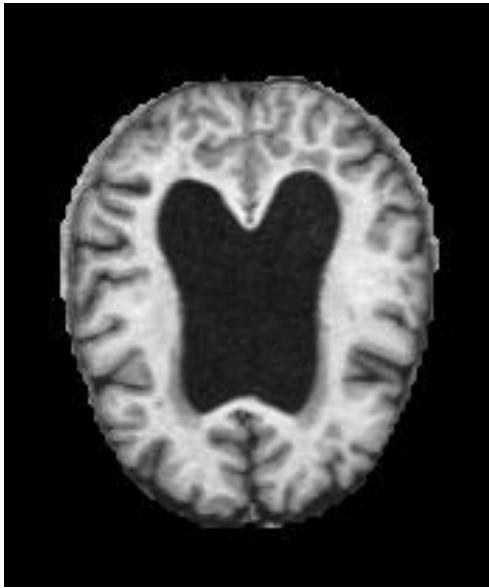


Figure 4.1: Demented MRI image

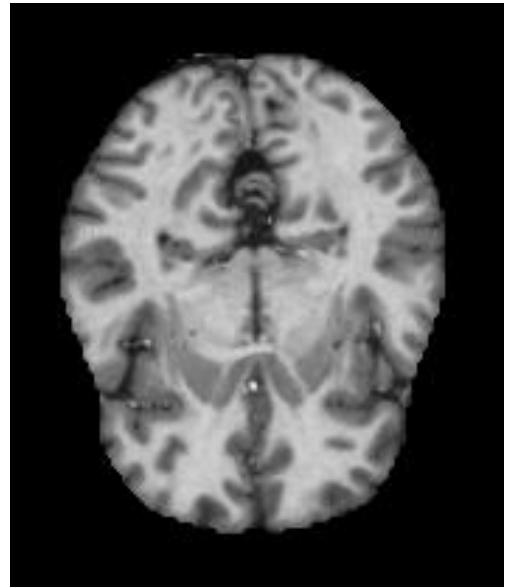


Figure 4.2: Non-Demented MRI image

### 4.1.2 Image Resizing and Interpolation

Bicubic interpolation is a more sophisticated method for resizing images. It takes into account the values of 16 surrounding pixels to calculate the new pixel value, resulting in smoother transitions and higher-quality images [31]. This method is especially useful in medical imaging where fine details, such as small brain structures, need to be preserved during resizing. In the context of MRI images, bicubic interpolation helps standardize the size of images to ensure that all images used in the analysis have consistent dimensions. This standardization is crucial because it allows the model to focus on structural patterns rather than distortions introduced by resizing.

In medical image processing, maintaining the original quality of images is critical to avoid the loss of vital anatomical details. By applying bicubic interpolation, even after scaling up or down, the image preserves smooth gradients, reducing visual artifacts that could otherwise impair diagnosis or image analysis. This technique effectively avoids pixelation and blocky images, which would interfere with further image analysis steps such as edge detection or contour detection. It also allows for flexible resizing

without losing crucial image fidelity, making it essential for preparing MRI scans for deep learning models.

```
bicubic_img = resized_image.resize((new_width, new_height),  
Image.BICUBIC)
```

image_demented Properties	
General Security Details Previous Versions	
Property	Value
Description	
Title	
Subject	
Rating	☆ ☆ ☆ ☆ ☆
Tags	
Comments	
Origin	
Authors	
Date taken	
Program name	
Date acquired	
Copyright	
Image	
Image ID	
Dimensions	176 x 208
Width	176 pixels
Height	208 pixels
Horizontal resolution	96 dpi
Vertical resolution	96 dpi

Figure 4.3: Dimensions of MRI image before bicubic interpolation

image_demented_outlined_binary Properties	
General Security Details Previous Versions	
Property	Value
Origin	
Date taken	
Image	
Dimensions	2048 x 2048
Width	2048 pixels
Height	2048 pixels
Bit depth	8
File	
Name	image_demented_outlined_binary.png
Item type	PNG File
File location	C:\DESK\Desktop\report\bicubic
Date created	15-10-2024 17:57
Date modified	15-10-2024 17:57
Size	226 KB
Attributes	A
Availability	
Offline status	
Shared with	
Owner	RRINNEI\dsouz

Figure 4.4: Dimensions of MRI image after bicubic interpolation

### 4.1.3 Grayscale conversion

The operation in the code utilizes the `convert('L')` function to transform the image into grayscale, reducing its color complexity by converting each pixel into a corresponding luminance (brightness) value. This process helps convert the three-channel RGB image (red, green, blue) into a single channel representing light intensity. In image analysis, especially for tasks like edge detection, grayscale conversion simplifies the data, allowing for more efficient computation and easier detection of sharp intensity changes. As edge detection relies on differences in pixel intensity, grayscale simplifies

the process by focusing solely on brightness rather than color variations. This makes it easier to highlight and detect crucial features like boundaries and contours in the image. Furthermore, grayscale conversion reduces computational load and memory usage, enabling faster processing of images while retaining essential structural information.

```
resized_img = np.array(resized_image_bicubic.convert('L'))
```



Figure 4.5: Gray conversion on Demented MRI image

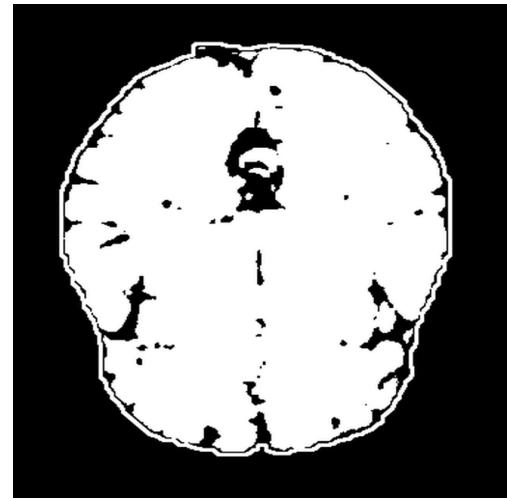


Figure 4.6: Gray conversion on Non-Demented MRI image

#### 4.1.4 Gaussian Blur

Gaussian Blur smoothens the image by reducing noise and fine details, which can interfere with edge detection processes. This technique applies a Gaussian function to average pixel values within a surrounding region, effectively softening transitions between pixels and reducing random variations or noise. By blurring the image, only the significant features remain sharp, making the detection of prominent edges more accurate. The Gaussian function assigns more weight to the pixels near the center of the window, ensuring that the blurring effect is smoother and less abrupt, thus preserving essential image structures while minimizing false edges.

```
blurred_img = cv2.GaussianBlur(resized_img, (5,5), 0)
```



Figure 4.7: Gaussian Blur on Demented MRI image

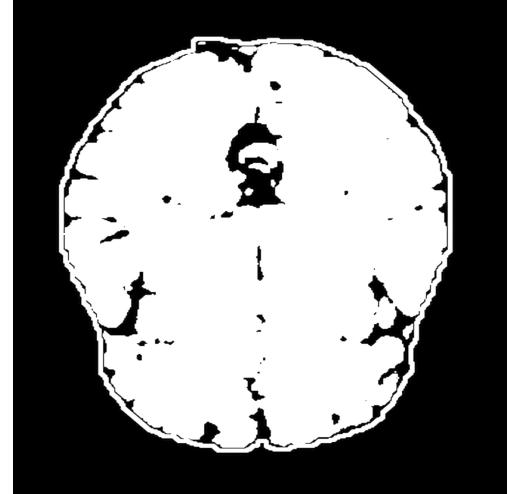


Figure 4.8: Gaussian Blur on Non-Demented MRI image

#### 4.1.5 Edge Detection

The Canny edge detection algorithm is a multi-step process that detects areas in the image where pixel intensity changes abruptly, marking the boundaries of objects in the image. This technique begins by applying a Gaussian blur to smooth the image and reduce noise, which is essential because noise can create false edges [40]. The next step involves calculating the intensity gradient for each pixel, determining where the changes in pixel intensity are the steepest. Through non-maximum suppression, the algorithm filters out weak edges, ensuring that only the strongest edges remain. This makes the Canny algorithm particularly robust, allowing it to detect clear and well-defined edges in MRI scans of the brain.

In the context of Alzheimer's disease, edge detection plays a key role in identifying brain structures such as the hippocampus and cortex, which undergo structural changes during the disease's progression. By isolating these boundaries, Canny edge detection allows the model to analyze specific areas that are most affected by Alzheimer's. It

provides a clear distinction between the different tissues and enhances the ability to identify brain atrophy. This step is vital in preprocessing because well-defined edges ensure that the next stages of the pipeline, such as contour detection and segmentation, have accurate boundaries to work with.

```
edges = cv2.Canny(blurred_img, threshold1=30, threshold2=100)
```

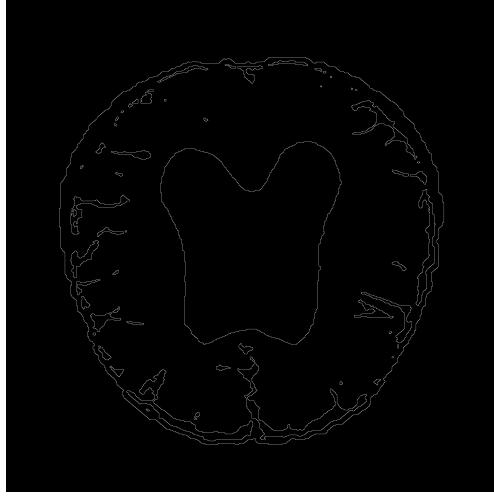


Figure 4.9: Canny edge detection on Demented MRI image

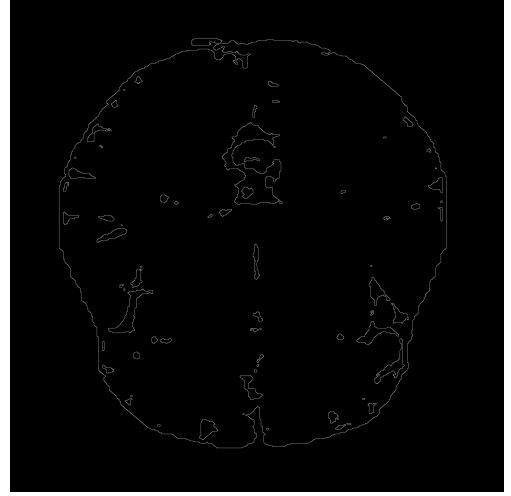


Figure 4.10: Canny edge detection on Non-Demented MRI image

#### 4.1.6 Otsu Thresholding

Otsu's method is an adaptive thresholding technique used to segment an image into foreground (brain tissue) and background (non-brain areas) [35]. By automatically selecting an optimal threshold based on the image's histogram, Otsu's method maximizes the variance between the foreground and background pixels, ensuring that brain tissue is clearly distinguished from the surrounding structures. This is particularly useful in brain MRI scans where accurate segmentation is necessary for identifying regions of interest, such as detecting areas of atrophy linked to Alzheimer's disease. Once the threshold is applied, the image is converted to a binary format (black and white), making it easier to analyze the distribution of tissues. The application of Otsu thresholding in

Alzheimer's disease research allows for more accurate analysis of brain morphology. By converting the grayscale MRI image into a binary form, this method simplifies the task of distinguishing between healthy and diseased tissue. It is especially useful in measuring hippocampal volume and cortical thinning, which are critical markers for the early detection of Alzheimer's. This technique allows for automatic segmentation, removing the need for manual intervention and improving the overall efficiency of image processing workflows

```
threshold_value = filters.threshold_otsu(gray_image)
```



Figure 4.11: Otsu threshold on Demented MRI image

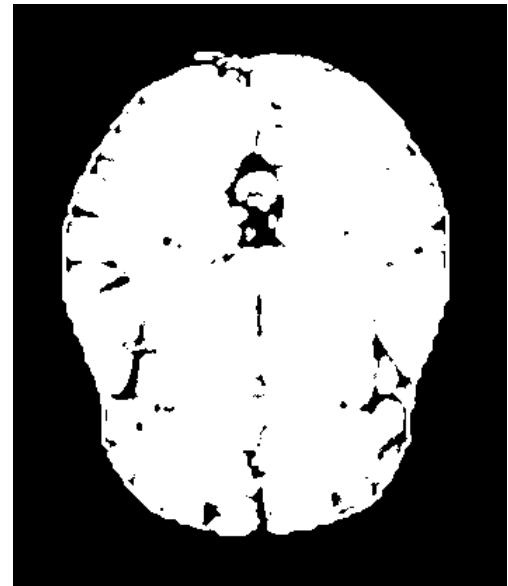


Figure 4.12: Otsu threshold on Non-Demented MRI image

#### 4.1.7 Morphological Operations

Morphological operations are used in image processing to improve the shape and structure of the objects within an image. Dilation expands the boundaries of objects, making them more pronounced, while closing fills in small gaps in the edges of objects. These techniques are applied after edge detection and contour detection to ensure that the

detected boundaries are smooth and continuous. For MRI scans, morphological operations are essential for refining the shapes of brain structures, ensuring that fragmented contours are connected and any small holes within the structures are filled. These operations are particularly useful in medical image analysis, where precise boundaries are crucial for accurate segmentation. For example, in Alzheimer's disease research, morphological operations help refine the contours of key brain regions, ensuring that the structures being analyzed are complete and continuous. Dilation and closing ensure that even small discontinuities in the contours are corrected, leading to more accurate volume calculations and structural assessments of brain regions, which are critical for diagnosing and monitoring the progression of Alzheimer's.

```
kernel = np.ones((5, 5), np.uint8)
edges_dilated = cv2.dilate(edges, kernel, iterations=1)
edges_closed = cv2.morphologyEx(edges_dilated,
                                cv2.MORPH_CLOSE, kernel)
```



Figure 4.13: Morphological Operations on Demented MRI image

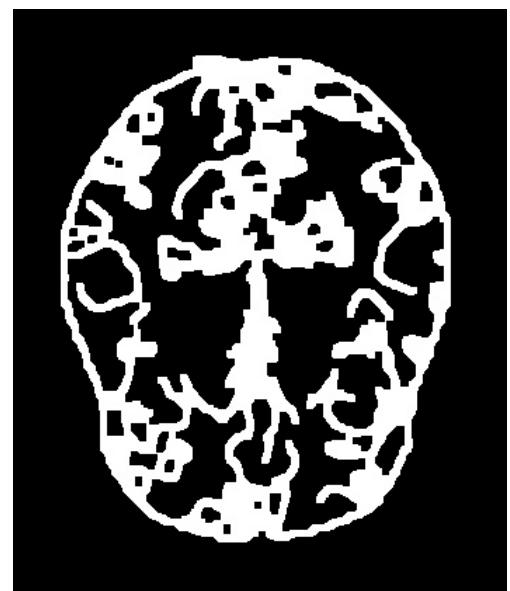


Figure 4.14: Morphological Operations on Non-Demented MRI image

#### 4.1.8 Contour Detection

Contour detection is an important step in image processing where the boundaries of significant objects in an image are traced. For MRI scans, contours highlight anatomical structures such as different regions of the brain that are critical in diagnosing diseases like Alzheimer's. By identifying and outlining these contours, the image processing algorithm can isolate regions of interest, such as brain, which often shows atrophy in Alzheimer's patients. The contour detection algorithm follows the edges detected in the previous stage and draws outlines around them, making it easier to distinguish one structure from another. In medical imaging, contours are often used to define the regions of interest (ROI) for further analysis, such as tissue segmentation or 3D reconstruction. For instance, in Alzheimer's research, contour detection can be used to outline the boundaries of the brain that are critical for memory and cognitive function. By drawing contours around these regions, the algorithm helps in quantifying the degree of atrophy or tissue loss, which can then be correlated with disease progression. This technique is essential for segmenting brain regions in a clear and interpretable manner, aiding in diagnosis

```
contours, _ = cv2.findContours(binary_image,
                               cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

cv2.drawContours(mask, contours, -1,
                 (255, 255, 255), thickness=cv2.FILLED)
```

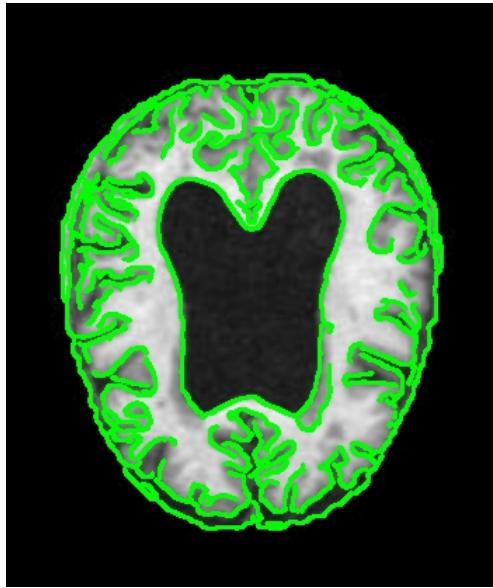


Figure 4.15: Contour detection on Demented MRI image

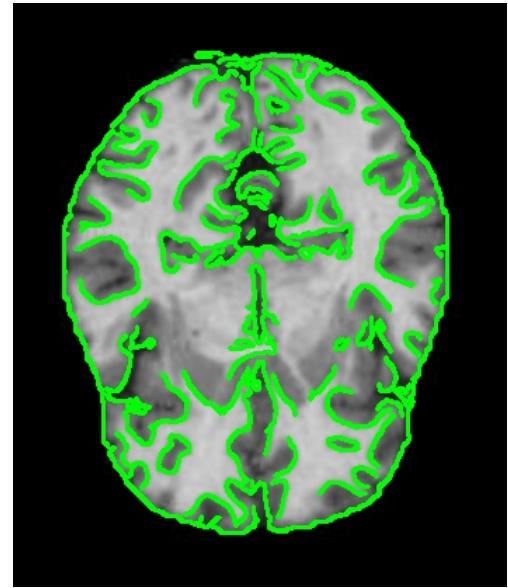


Figure 4.16: Contour detection on Non-Demented MRI image

#### 4.1.9 Red border around the brain

Implemented a feature to add a red border around the brain region after performing morphological operations. This was necessary because the background of the images was black, which complicated the process of counting black and white pixels. By adding a red border, I was able to clearly delineate the brain from the black background, ensuring that only the relevant regions were analyzed for pixel ratio calculations. This step improved the accuracy of classifying the images into categories based on the proportion of black and white pixels. The feature used in the function to make red border around the brain is the contour detection. It detects the contours in the image using `cv2.findContours` after converting the image to binary. Then, the function creates a mask and draws the detected contours using `cv2.drawContours`, effectively highlighting the brain region. By applying a red color ([0, 0, 255]) to the areas outside the detected brain region, this method ensures a clear distinction between the brain and the background for further pixel analysis.

```
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
_, binary_image = cv2.threshold(gray_image, 1,
                                255, cv2.THRESH_BINARY)
contours, _ = cv2.findContours(binary_image,
                               cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
mask = np.zeros_like(image)
cv2.drawContours(mask, contours, -1,
                  (255, 255, 255), thickness=cv2.FILLED)
inverted_mask = cv2.bitwise_not(mask)
image[inverted_mask[:, :, 0] == 255] = [0, 0, 255]
cv2.imwrite(save_path, image)
```

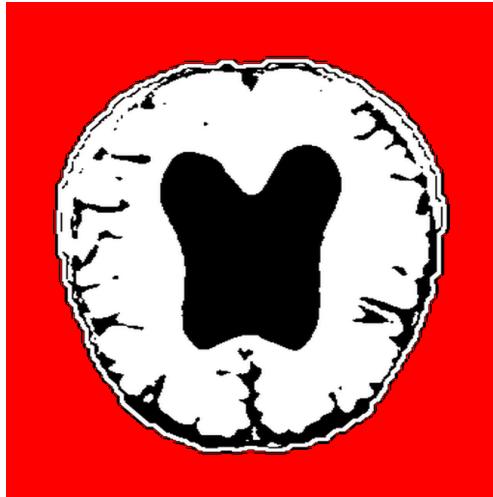


Figure 4.17: Red border around the on Demented MRI image

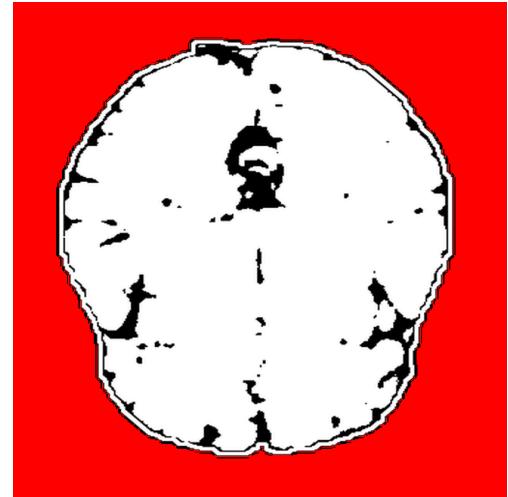


Figure 4.18: Red border around the Non-Demented MRI image

#### 4.1.10 Pixel Counting

Black and White Pixel Counting involves counting the number of black and white pixels in a binary image, providing quantitative data on the distribution of tissues in medical images. This technique is particularly useful in analyzing MRI images where the white

pixels typically represent brain tissue, and the black pixels represent the background or areas with no tissue. By counting the number of white pixels, the algorithm can calculate the proportion of brain tissue present in the scan, which is critical for measuring tissue loss or atrophy, key indicators in Alzheimer's disease. In Alzheimer's research, pixel counting helps quantify the degree of atrophy in specific brain regions such as the hippocampus or cortex. By analyzing the binary image, researchers can determine how much brain tissue has been lost over time, providing a clear metric for disease progression. This quantitative data is valuable for both diagnosis and monitoring, as it provides an objective measure of how the disease is affecting the brain.

```
image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

total_pixels = image.size

black_pixels = np.sum(image == 0)

white_pixels = np.sum(image == 255)

black_percentage = (black_pixels / total_pixels) * 100

white_percentage = (white_pixels / total_pixels) * 100
```

#### 4.1.11 Excel Reporting

The Excel reporting feature allows for the results of the image processing pipeline to be documented and saved for further analysis. After processing the MRI images, the script calculates various metrics, such as the number of black and white pixels in the binary image, and saves this data to an Excel file. This provides a quantitative analysis of the image segmentation process, which is crucial for documenting the degree of brain tissue affected by Alzheimer's. Excel reporting is particularly useful in medical research, where large datasets need to be carefully documented and analyzed. By saving the pixel counts and other relevant metrics, researchers can track changes in brain volume or tissue over time, providing valuable insights into the progression of Alzheimer's disease.

## Chapter 4. Design and Implementation

---

This quantitative data can then be used for statistical analysis, helping to validate the findings and ensure the accuracy of the image processing pipeline.

```
sheet = workbook.active

sheet.append(["Image", "Black Pixels", "White Pixels",
              "Black Pixel Percentage", "White Pixel Percentage"])

for data in report_data:
    sheet.append(data)

workbook.save(excel_file_path)

print(f"Excel report saved at: {excel_file_path}")
```

In the provided Excel report, the formula `=IF(F2>7, "DEMENTED", "NON DEMENTED")` is used to classify whether an MRI image corresponds to a demented or non-demented subject based on pixel percentage values in column D. If the black pixel percentage (in F2) exceeds 7, the formula labels the image as "DEMENTED"; otherwise, it is labeled "NON DEMENTED."

For summarizing the count of categories, the formulas:

```
1 =COUNTIFS(F2:FLast_row, ">0", F2:Last_row, "<7.0")
2 =COUNTIFS(F2:Last_row, ">7.0", F2:Last_row, "<40")
```

Listing 4.1: Excel formulas for counting non-demented and demented images

Listing 4.1 used to count the number of non-demented and demented images, respectively. These formulas check the pixel percentage ranges to differentiate between the two categories.

Image	Black Pixels	White Pixels	Black Pixel Percentage	White Pixel Percentage	Overall Black Pixel Percentage	Demented/Non-Demented
image_demented_outlined_binary.png	27163	75810	10.36186218	28.91921997	26.37875948	DEMENTED
image_nondem_outlined_binary.png	1816	115692	0.692749023	44.13299561	1.545426694	NON DEMENTED
NONDEMENTED	1					
DEMENTED	1					

Figure 4.19: Excel report of the MRI image

## 4.2 Convolution Neural Network Technique

Convolutional Neural Networks (CNNs) have become a powerful tool for the detection and classification of Alzheimer's Disease (AD) using medical imaging, such as MRI or CT scans. By automatically learning features from raw images, CNNs can identify patterns that distinguish between various stages of AD, including non-demented, very mild, mild, and moderate stages. The architecture of a CNN, with layers like convolutional, pooling, and fully connected layers, enables it to extract hierarchical features, from simple edges to complex brain tissue structures associated with AD. This method surpasses traditional manual feature extraction techniques by learning subtle differences in brain morphology that may not be immediately evident. By training the model on labeled datasets, CNNs can effectively classify images into different dementia stages, offering an invaluable tool for early diagnosis and disease progression monitoring, which is crucial for timely interventions in Alzheimer's Disease.

### 4.2.1 Data Collection and Labeling for Alzheimer's Disease Image Dataset

the process being performed is the collection and labeling of image file paths for an Alzheimer's disease dataset. The dataset is organized into four categories: 'Mild Demented', 'Moderate Demented', 'Non Demented', and 'Very Mild Demented', representing different stages of dementia. The script uses a dictionary called directories where each key corresponds to a label for the dementia stage, and each value is the path to the directory containing the corresponding images. The code iterates through each directory, collects the full file path of every image, and associates each image with the correct label based on the directory from which it is retrieved. These file paths and labels are stored in two lists, filepaths and labels.

After gathering the data, the file paths and labels are converted into pandas Series

(Fseries for file paths and Lseries for labels), which are then used to construct a DataFrame called `Alzheimer_df`. This DataFrame holds two columns: 'filepaths' (containing the location of each image) and 'labels' (containing the corresponding dementia stage). The `value_counts()` function is used to print out the distribution of labels, showing how many images belong to each category. Finally, the shape of the DataFrame, which indicates the number of images and labels collected, is printed for verification. This step is crucial for further processing, such as training a CNN model for Alzheimer's disease classification.

```
labels
Very Mild Demented    2240
Mild Demented         896
Non Demented          176
Moderate Demented     64
Name: count, dtype: int64
Alzheimer_df:
(3376, 2)
```

Figure 4.20: Class Distribution and Dataset Overview

#### 4.2.1.1 Dataset Splitting for Training, Testing, and Validation

The process being performed is splitting the Alzheimer's disease dataset into training, testing, and validation sets. The `train_test_split()` function from the `sklearn.model_selection` module is used to divide the dataset into subsets, which are crucial for evaluating the performance and generalization of the CNN model. Initially, the dataset (`Alzheimer_df`) is split into two parts: the `train_images` set (75% of the data) and the `test_images` set (25% of the data) using `test_size=0.25`. This division ensures that the model will be trained on the training set and later tested on the test set to measure its accuracy on unseen data. The split is done randomly, but the `random_state=42` argument ensures the results are reproducible by producing

the same split every time the code is run.

Additionally, the training data (`train_images`) is further split into two subsets: `train_set` and `val_set`. The `train_set` (80% of the original dataset) will be used for training the CNN, and the `val_set` (20% of the dataset) will be used for validating the model's performance during training. This validation set helps in monitoring the model's performance, tuning hyperparameters, and preventing overfitting before the final evaluation on the test set. Finally, the shapes of each set are printed to verify the correct sizes of the splits, ensuring that the training, validation, and testing sets contain the right number of samples.

```
Train_set (2700, 2)
Test_set (844, 2)
Train_set (2532, 2)
Validation_set (676, 2)
```

Figure 4.21: Dataset Split Overview: Training, Testing, and Validation Set Sizes

#### 4.2.1.2 Image Data Augmentation and Generator Creation for CNN Training

The process being performed here is image data augmentation and the creation of data generators for training, validation, and testing of a Convolutional Neural Network (CNN) model. Data augmentation is applied to the training images to enhance the model's ability to generalize by artificially expanding the dataset with transformations such as rotation, zoom, shift, and horizontal flip. The `ImageDataGenerator` is used to apply these transformations to the training images, which are passed through MobileNetV2's preprocessing function to scale pixel values appropriately. On the other hand, no augmentation is applied to the validation and test sets; however, they are still preprocessed using the `MobileNetV2` function to maintain consistency in image format across all datasets.

The `create_generator` function simplifies the creation of data generators using the `flow_from_dataframe()` method. This function takes in parameters like the dataframes (for training, validation, and test sets), the image file paths (`x_col`), the labels (`y_col`), and other attributes such as target size, color mode, and batch size. It returns generators that will be fed into the CNN during training. The `train_generator` is set to shuffle the data to ensure randomness during training, while both `test_generator` and `val_generator` do not shuffle, ensuring the model is evaluated consistently on the same data order during validation and testing. These generators load the images in batches during the model's training and evaluation, enabling efficient memory usage and allowing real-time data augmentation.

```
Found 2700 validated image filenames belonging to 4 classes.  
Found 844 validated image filenames belonging to 4 classes.  
Found 676 validated image filenames belonging to 4 classes.
```

Figure 4.22: Data Generator Initialization: Number of Images in Each Sets

#### 4.2.1.3 Visualization and Model Construction for Alzheimer's Disease Classification

The process being performed here is twofold: visualizing image samples from the dataset and constructing a Convolutional Neural Network (CNN) model for Alzheimer's disease classification. The function `show_knee_images` takes a batch of images from the provided data generator (in this case, the `train_generator`), displays 25 images from the dataset, and labels them with their corresponding class names. It adjusts the image pixel values to be between 0 and 1 for better visualization and uses `matplotlib` to plot the images. The function identifies the predicted class of each image by calculating the index of the maximum value in the label array, which corresponds to one of the four classes (e.g., Mild Demented, Moderate Demented, Non Demented, Very Mild Demented). This visualization step helps inspect if the images

are being correctly loaded and augmented.

In the second part of the code, a deep learning model is constructed using the Xception architecture pre-trained on the ImageNet dataset. The Xception model, without its top layers (`include_top=False`), serves as the base model to extract high-level features from the input images. The model architecture is extended with a `Flatten` layer to transform the feature maps into a one-dimensional vector, followed by two fully connected layers with `Dropout` to reduce overfitting. The final layer is a `Dense` layer with 4 units (one for each class) and a `softmax` activation function to perform multi-class classification. The model is compiled using the Adamax optimizer, with categorical cross-entropy as the loss function, which is suitable for multi-class classification tasks. The model summary is printed, and it is built with the specified input image shape (244, 244, 3). This step prepares the CNN for training on the Alzheimer's dataset.

Layer (type)	Output Shape	Param #
xception (Functional)	(None, 2048)	20,861,480
flatten (Flatten)	(None, 2048)	0
dropout (Dropout)	(None, 2048)	0
dense (Dense)	(None, 128)	262,272
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 4)	516

Total params: 21,124,268 (80.58 MB)

Trainable params: 21,069,740 (80.37 MB)

Non-trainable params: 54,528 (213.00 KB)

Figure 4.23: Xception CNN Architecture and Parameters

## 4.2.2 Model Training with Validation for Alzheimer’s Disease Classification

The process being performed here is training the Convolutional Neural Network (CNN) model using the training data and simultaneously evaluating its performance on the validation set. The `model.fit()` function initiates the training process, where the model learns to classify images into one of the four Alzheimer’s disease categories (e.g., Non Demented, Mild Demented, Moderate Demented, Very Mild Demented). The training data is provided via the `train_generator`, which loads batches of augmented images during each epoch. The model goes through 10 epochs, meaning the entire training dataset is used 10 times to adjust the model’s weights. During each epoch, the model optimizes its parameters using the Adamax optimizer and calculates the categorical cross-entropy loss to guide its learning process.

In addition to training, the model’s performance is evaluated after every epoch on the validation set through the `validation_data=val_generator` argument. This allows the model to be tested on unseen data at regular intervals (every epoch, due to `validation_freq=1`) to monitor overfitting and generalization capability. The validation process ensures that the model’s performance is consistent and helps in early stopping if overfitting is detected. The training history, stored in the `history` object, includes loss, accuracy, validation loss, and validation accuracy values, which can be used later to analyze the model’s learning curve and performance.

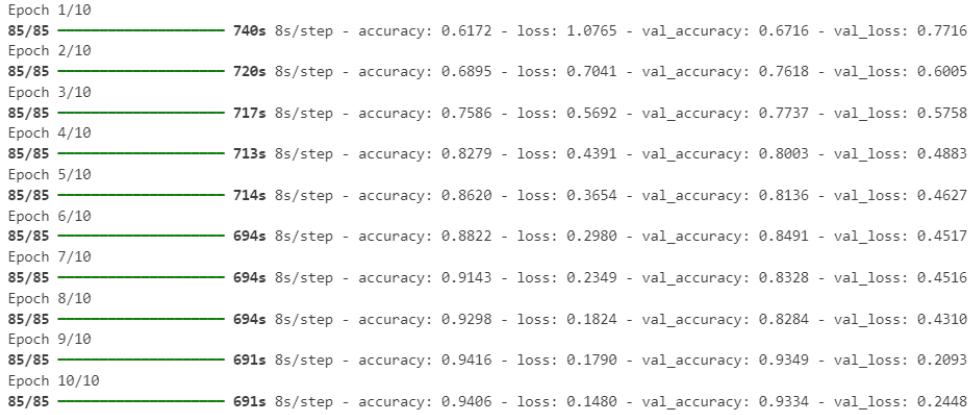


Figure 4.24: Epoch-wise Training and Validation Accuracy and Loss

### 4.2.3 Model Evaluation on Training Data

The process being performed here is evaluating the model's performance on the training data using the `model.evaluate()` function. This function takes the `train_generator` as input, which provides batches of images and labels from the training set. During evaluation, the model makes predictions on the provided data and compares them to the actual labels to calculate the loss and accuracy metrics. The `verbose=1` argument ensures that detailed progress information is displayed during the evaluation, such as the number of steps completed and the final results for loss and accuracy.

The evaluation phase is important as it gives insight into how well the model has learned from the training data. While a high accuracy on the training set can indicate that the model has successfully fit the data, it is crucial to compare these results with those from the validation and test sets. A significant difference between training and validation accuracy may indicate overfitting, where the model performs well on the data it has seen but struggles to generalize to new, unseen data. Therefore, while the training evaluation is useful for understanding how well the model is learning, it's equally important to assess the model on the validation and test datasets to ensure its

generalization capabilities.

```
85/85 ━━━━━━━━ 160s 2s/step - accuracy: 0.9578 - loss: 0.1125  
[0.11091769486665726, 0.9574074149131775]
```

Figure 4.25: Training Data Evaluation: Accuracy and Loss Results

#### 4.2.4 Model Training for One Epoch with Validation

The process being performed here is training the Convolutional Neural Network (CNN) model for a single epoch using the `model.fit()` function. The `train_generator` supplies batches of images from the training dataset to the model during this epoch, which allows the model to update its weights based on the computed loss. The `epochs=1` argument indicates that the entire training dataset will be used once during this training process. The model is designed to learn how to classify Alzheimer's disease stages, and during each epoch, the loss function (in this case, categorical cross-entropy) is minimized using the Adamax optimizer. This step is crucial for adjusting the model's parameters (weights) and learning from the training data.

In addition to training, the model's performance is evaluated on the validation set at the end of the epoch. The `validation_data=val_generator` argument specifies that the model will use the validation data after each epoch to monitor its performance. The `validation_freq=1` ensures that validation occurs after every epoch, providing insights into how well the model generalizes to unseen data. This validation step is critical to identify if the model is overfitting the training data or if it is capable of learning patterns that extend beyond the training set. After this process, the `history1` object will store metrics such as training accuracy, validation accuracy, training loss, and validation loss, which can be used to analyze the model's performance.

```
85/85 ━━━━━━━━ 711s 8s/step - accuracy: 0.9570 - loss: 0.1245 - val_accuracy: 0.9320 - val_loss: 0.1914
```

Figure 4.26: Epoch-wise Training and Validation Accuracy and Loss

#### 4.2.5 Model Evaluation on Test Data

The process being performed here is evaluating the Convolutional Neural Network (CNN) model's performance on the test dataset using the `model.evaluate()` function. The `test_generator` provides batches of images and labels from the test set, which the model uses to make predictions. The predictions are compared against the actual labels to compute evaluation metrics such as loss and accuracy. The argument `verbose=1` ensures that detailed information, including the number of steps and final results for loss and accuracy, is printed during the evaluation process.

This evaluation phase is crucial as it provides insight into how well the model generalizes to unseen data. While training and validation accuracy indicate how well the model performs on the data it has been trained on or validated with, the test accuracy shows how well the model can classify completely new data. A high test accuracy suggests that the model is capable of generalizing well, while significant differences between training and test accuracy might indicate overfitting, where the model performs well on the training data but struggles with new, unseen data.

```
[0.1705905646085739, 0.9383886456489563]
```

Figure 4.27: Test Data Evaluation: Loss and Accuracy Results

#### 4.2.6 Model Prediction and Class Label Mapping

The process being performed here is making predictions on the test dataset using the trained Convolutional Neural Network (CNN). The `model.predict()` function is applied to the `test_generator`, which generates batches of test images. The model

outputs a probability distribution across the different classes for each test image. The line `pred = np.argmax(pred, axis=1)` converts these probability distributions into class predictions by selecting the index of the highest probability for each image. This results in a list of predicted class indices for all the test images.

After generating the class predictions, the code maps the predicted indices to their corresponding class labels. First, the class labels and their respective indices are retrieved from the `train_generator.class_indices` dictionary, which provides the mapping of class names to numerical labels. The dictionary is then inverted, so the keys become the class indices and the values become the class names. Finally, the list comprehension `pred2 = [labels[k] for k in pred]` is used to convert the predicted class indices back into the actual class names, resulting in a list of class names corresponding to the model's predictions on the test dataset. This allows for easier interpretation of the predictions by converting numeric indices into human-readable labels.

27/27 ————— 49s 2s/step

Figure 4.28: Prediction Process: Test Set Evaluation Progress

#### 4.2.7 Plotting Training and Validation Accuracy Over Epochs

The process being performed here is visualizing the model's accuracy during training and validation over multiple epochs. The code uses `plt.plot()` from the `matplotlib` library to create a line graph showing how the model's accuracy changes as it trains on the dataset.

---

##### Algorithm 1 Plotting Accuracy History

---

```
1: plt.plot(history.history['accuracy'] +  
           history1.history['accuracy']))
```

---

Combines the accuracy metrics from two training sessions (`history` and `history1`) to display the training accuracy across all epochs.

---

**Algorithm 2** Plotting Validation Accuracy History

---

```
1: history.history['val_accuracy'] +  
   history1.history['val_accuracy']
```

---

For validation accuracy, how well the model performs on unseen validation data. This helps track the model's progress and provides insight into how well it is learning over time.

The plot includes titles, axis labels, and a legend to improve readability. The title `model accuracy` indicates what the plot represents, while the x-axis (epoch) shows the number of training iterations, and the y-axis (accuracy) shows the accuracy metric. The `plt.legend()` function adds labels to distinguish between training and validation accuracy in the plot. By visualizing the training and validation accuracy, this process helps identify whether the model is learning effectively and whether it is overfitting (when training accuracy is high, but validation accuracy lags).

The plot shows the model's training accuracy (blue line) and validation accuracy (orange line) over 10 epochs. Both accuracies increase steadily during the initial epochs, indicating that the model is learning from the data. However, after around epoch 8, the validation accuracy plateaus slightly, while the training accuracy continues to improve. This could suggest the beginning of overfitting, where the model performs better on the training data but shows less improvement on the unseen validation data.

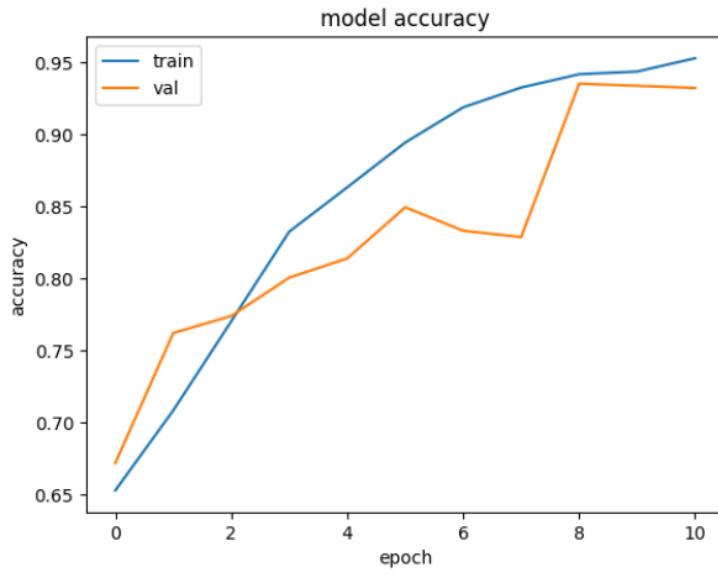


Figure 4.29: Training vs Validation Accuracy Over Epochs

#### 4.2.8 Plotting Training and Validation Loss Over Epochs

The process being performed here is visualizing the model's loss during training and validation across multiple epochs. The `plt.plot()` function from `matplotlib` is used to create a line graph that shows how the loss decreases (or increases) over time.

---

##### Algorithm 3 Plotting Loss History

---

```
1: plt.plot(history.history['loss'] +
           history1.history['loss'])
```

---

Combines the loss values from two training sessions (`history` and `history1`) to display the model's loss during training for each epoch.

---

##### Algorithm 4 Plotting Validation Loss History

---

```
1: plt.plot(history.history['val_loss'] +
           history1.history['val_loss'])
```

---

Does the same for the validation loss, showing how well the model performs on unseen

validation data. Loss is an important metric because it indicates how well the model is minimizing errors during training.

The graph includes titles, axis labels, and a legend to distinguish between the training and validation loss curves. The x-axis (epoch) shows the number of training iterations, while the y-axis (loss) represents the calculated loss value for each epoch. The legend, added with `plt.legend()`, helps identify which line corresponds to training and validation loss. By plotting both losses, this visualization allows you to monitor the model's performance and observe whether overfitting occurs—typically seen when training loss continues to decrease, but validation loss increases or plateaus.

Graph shows the training and validation loss over 10 epochs for a Convolutional Neural Network (CNN) model. The blue line represents the training loss, while the orange line represents the validation loss. Both losses decrease significantly in the initial epochs, indicating that the model is learning and minimizing errors effectively. However, after around 7 epochs, the validation loss plateaus and even increases slightly, while the training loss continues to decrease. This could indicate the beginning of overfitting, where the model performs well on the training data but struggles to generalize to unseen validation data. Overall, the consistent decrease in loss during the early stages demonstrates the model's improvement, but the later divergence between training and validation loss suggests the need to monitor for overfitting.

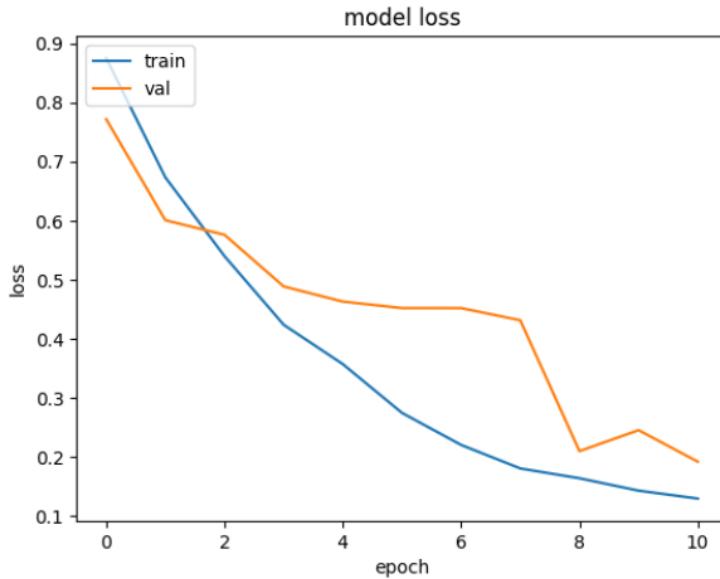


Figure 4.30: Training vs Validation Loss Over Epochs

#### 4.2.9 Model Performance Evaluation Using Classification Report and Accuracy

The process being performed here is evaluating the model's performance on the test dataset using key metrics such as the classification report and accuracy score. First, the `classification_report()` function from `sklearn.metrics` is used to generate a detailed report on the model's performance, including precision, recall, F1-score, and support for each class. These metrics provide insights into how well the model distinguishes between different classes. Precision indicates the proportion of true positive predictions among all positive predictions, recall measures the model's ability to find all relevant cases, and the F1-score provides a balance between precision and recall. By printing this report, you gain a comprehensive understanding of the model's strengths and weaknesses across all test categories.

In addition to the classification report, the model's overall accuracy is calculated using the `accuracy_score()` function, which compares the predicted labels (`pred2`)

against the actual labels (`y_test`). The accuracy score represents the proportion of correct predictions made by the model. The result is formatted to display as a percentage, providing a clear indication of how well the model has generalized to the test data. This overall accuracy metric is a useful summary of the model's performance, but it is essential to consider it in conjunction with the other metrics from the classification report to gain a complete picture, particularly in cases of imbalanced data.

This table shows the classification report for the Alzheimer's disease detection model, which includes precision, recall, F1-score, and support for each class: Mild Demented, Moderate Demented, Non Demented, and Very Mild Demented. Precision measures the accuracy of positive predictions, recall indicates how well the model identifies all true positives, and the F1-score provides a balance between precision and recall. The "support" column shows the number of actual instances in each class. Overall, the model has an accuracy of 93.8

	precision	recall	f1-score	support
Mild Demented	0.98	0.87	0.92	253
Moderate Demented	1.00	0.93	0.97	15
Non Demented	0.91	0.70	0.79	46
Very Mild Demented	0.92	0.99	0.96	530
accuracy			0.94	844
macro avg	0.95	0.87	0.91	844
weighted avg	0.94	0.94	0.94	844

Accuracy of the Model: 93.8%

Figure 4.31: Classification Report and Accuracy of AD Detection Model

## 4.3 Summary

In designing Image Processing Technique, the primary focus was on classifying Alzheimer's disease stages using MRI data through a combination of image processing and deep learning techniques. Initially, MRI images were preprocessed using resizing,

Gaussian Blur, and bicubic interpolation to enhance uniformity and quality. Techniques such as Canny edge detection and Otsu's thresholding were employed to isolate key brain structures. Following this, a Convolutional Neural Network (CNN) was fine-tuned to classify MRI images into four categories—Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented. This experiment highlights how advanced image processing and deep learning techniques can more effectively distinguish various Alzheimer's disease stages compared to traditional methods.

Convolutional Neural Networks (CNNs) have proven to be highly effective in detecting and classifying Alzheimer's Disease (AD) using medical imaging, particularly MRI and CT scans. By automatically learning features from raw images, CNNs are able to identify critical patterns that help distinguish between different stages of AD, such as non-demented, very mild dementia, mild dementia, and moderate dementia. Through its hierarchical architecture of convolutional, pooling, and fully connected layers, a CNN can capture and analyze detailed brain structures associated with Alzheimer's progression, making it an essential tool for early diagnosis and monitoring of the disease.

# **Chapter 5**

## **Results and Discussions**

### **5.1 Results**

#### **5.1.1 Accuracy using image processing technique**

The Convolutional Neural Network (CNN) model developed for Alzheimer's Disease detection demonstrates significant promise in accurately classifying brain scan images into relevant categories, including Mild Demented, Moderate Demented, Non Demented, and Very Mild Demented. Leveraging deep learning techniques, the model effectively learns intricate patterns and features within the imaging data, leading to improved diagnostic capabilities. The results showcase the model's high accuracy, precision, and recall metrics, highlighting its potential as a valuable tool for clinicians in the early detection and monitoring of Alzheimer's Disease. By automating the analysis of brain scans, this CNN not only enhances diagnostic efficiency but also aids in personalized treatment planning, ultimately contributing to better patient outcomes in the field of neurology.

In this image processing technique, Alzheimer's Disease (AD) detection is performed by analyzing the black and white pixel ratio in brain scan images. The approach focuses

on identifying specific patterns and textures in the brain that are indicative of AD progression. In this method, the images are first preprocessed this may involve steps like converting the images to grayscale and applying Gaussian blur to reduce noise and highlight important brain structures. After preprocessing, the number of black and white pixels in the image is counted. These pixels are typically categorized based on intensity thresholds, with white pixels representing brain matter and black pixels potentially representing atrophied regions or areas with reduced brain activity, which are common in demented brains.

The ratio of black to white pixels serves as a key metric in differentiating between demented and non-demented subjects. In non-demented brains, the white pixel count is usually higher, as the brain tissue is more intact and active. In contrast, demented brains, affected by Alzheimer's disease, tend to show a higher proportion of black pixels, indicating tissue loss or degradation. By calculating this ratio, the system can classify an image as either "Non-Demented" or "Demented." This method offers a simple yet effective approach to Alzheimer's detection, allowing for a more straightforward analysis of brain deterioration patterns without the need for complex machine learning algorithms.

ACTUAL	PREDICTION	
	POSITIVE	NEGATIVE
POSITIVE	TRUE POSITIVE = 2355	FALSE NEGATIVE = 1157
NEGATIVE	FALSE POSITIVE = 845	TRUE NEGATIVE = 2043

Table 5.1: Confusion Matrix using image processing technique for AD detection

The table presents a confusion matrix for a classification model evaluating its performance on predicting Alzheimer's Disease (AD). In this matrix, the "ACTUAL" labels represent the true classifications of the data, while the "PREDICTION" labels indicate the model's predictions. The table outlines four key metrics: True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN). The total data comprises

4,402 samples, which are categorized based on the model's predictions against the actual outcomes.

True Positives (TP) are instances where the model correctly predicts that a patient has Alzheimer's (positive) with a count of 2,355. False Negatives (FN), which total 1,157, are cases where the model incorrectly predicts that a patient does not have Alzheimer's when they actually do. On the other hand, False Positives (FP), with a count of 845, represent instances where the model incorrectly predicts Alzheimer's presence in patients who do not have the disease. Lastly, True Negatives (TN), totaling 2,043, are cases where the model correctly identifies that a patient does not have Alzheimer's. The data captured in this confusion matrix is crucial for understanding the model's accuracy and its capacity to differentiate between patients with and without the disease effectively.

The accuracy formula presented above calculates the performance of a classification model by evaluating the ratio of correctly predicted instances to the total number of instances [50]. The formula is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- $TP$  (True Positives) = 2,355: the number of correct positive predictions.
- $TN$  (True Negatives) = 2,043: the number of correct negative predictions.
- $FP$  (False Positives) = 845: the number of incorrect positive predictions.
- $FN$  (False Negatives) = 1,157: the number of incorrect negative predictions.

### **Substituting the Values:**

To calculate the accuracy, we substitute the values into the formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Calculating the numerator:

$$TP + TN = 2,355 + 2,043 = 4,398$$

Calculating the denominator:

$$TP + TN + FP + FN = 2,355 + 2,043 + 845 + 1,157 = 6,400$$

Now substituting back into the accuracy formula:

$$\text{Accuracy} = \frac{4,398}{6,400} \approx 0.68718 \text{ or } 68.72\%$$

This result indicates that the model achieves an accuracy of approximately 68.72%, suggesting its effectiveness in distinguishing between patients with and without Alzheimer's Disease based on the given predictions.

### 5.1.2 Accuracy using Convolution Neural Network

The evaluation metrics displayed—Precision, Recall, F1-Score, and Support—are essential for assessing the performance of classification models, particularly in contexts involving imbalanced datasets. Precision is defined as the ratio of True Positives (TP) to the sum of True Positives and False Positives (FP), indicating the accuracy of positive predictions made by the model; a high precision score reflects few false positive errors. Recall, on the other hand, measures the ratio of True Positives to the sum of True Positives and False Negatives (FN), assessing the model's ability to identify all relevant instances and indicating how many actual positive cases were correctly predicted.

The F1-Score combines both precision and recall into a single metric, providing a

balance that is particularly useful for imbalanced classes, as it accounts for both false positives and false negatives. Finally, Support refers to the actual number of occurrences of each class in the specified dataset, providing context for the precision and recall values. Together, these metrics enable a comprehensive evaluation of model performance, guiding improvements and adjustments to enhance predictive accuracy [51].

**Precision:**

$$\text{Precision} = \frac{TP}{TP + FP}$$

**Recall:**

$$\text{Recall} = \frac{TP}{TP + FN}$$

**F1-Score:**

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

**Support:** Support refers to the number of actual occurrences of the class in the specified dataset.

Table 5.2: Classification Report

Class	Precision	Recall	F1-Score	Support
Mild Demented	0.98	0.87	0.92	253
Moderate Demented	1.00	0.93	0.97	15
Non Demented	0.91	0.70	0.79	46
Very Mild Demented	0.92	0.99	0.96	530
<b>Accuracy</b>		0.94		844
<b>Macro Avg</b>	0.95	0.87	0.91	844
<b>Weighted Avg</b>	0.94	0.94	0.94	844

The table 5.2 presents a comprehensive classification report for the Alzheimer's Disease detection model, detailing its performance across various categories: Mild Demented, Moderate Demented, Non Demented, and Very Mild Demented. Each class is evaluated using four key metrics: Precision, Recall, F1-Score, and Support. Precision measures the accuracy of positive predictions, with the model achieving high precision across

## Chapter 5. Results and Discussions

---

all classes, particularly in the Moderate Demented category, where it reached a perfect score of 1.00. Recall, which assesses the model's ability to identify all relevant cases, also shows strong performance, especially in the Very Mild Demented class with a recall of 0.99, indicating that nearly all actual instances were correctly identified.

The F1-Score, which balances precision and recall, further reinforces the model's effectiveness, particularly in the Mild Demented and Moderate Demented categories, where scores are 0.92 and 0.97, respectively. The Support column provides insight into the number of actual occurrences for each class in the dataset, highlighting that the model has been trained on a diverse range of instances. Overall, the model achieved an accuracy of 0.94, alongside macro and weighted averages of 0.95 and 0.94 for precision, recall, and F1-Score. These results indicate that the model is well-equipped to classify Alzheimer's stages accurately, suggesting its potential utility in clinical settings for enhanced diagnosis and patient management.

Table 5.3: Confusion Matrix using CNN for AD detection

Actual	MildDem	ModerateDem	NonDem	VMildDem
MildDem	220	0	2	31
ModerateDem	1	14	0	0
NonDem	0	0	32	14
VMildDem	3	0	1	526

The table 5.3 presents the confusion matrix for the Alzheimer's Disease detection model, detailing the actual classifications against the predicted classifications across four categories: Mild Demented (MildDem), Moderate Demented (ModerateDem), Non Demented (NonDem), and Very Mild Demented (VMildDem). The rows represent the actual class labels, while the columns indicate the predicted class labels. A notable feature of the matrix is the high number of True Positives (TP) for Mild Demented, where the model correctly identified 220 instances, as well as 526 instances of Very Mild Demented, showing that the model performs well in detecting these categories. However, there are also instances of misclassification; for example, 14 instances of

Moderate Demented were incorrectly classified as Mild Demented, and 32 instances of Non Demented were incorrectly predicted. The confusion matrix thus provides valuable insights into the model's strengths and weaknesses, indicating areas for improvement, particularly in distinguishing between Moderate and Very Mild Demented categories. Overall, the matrix highlights the model's effectiveness while also pointing to specific challenges in classification.

## 5.2 Discussion

The experimental research detailed in the study focuses on the early diagnosis of Alzheimer's disease using MRI image processing techniques and Convolutional Neural Networks (CNNs). The research method, implemented under controlled conditions, aimed to comprehensively test and assess the accuracy of detecting Alzheimer's stages from MRI brain scans. The results indicated that the image processing approach using techniques like Gaussian blur, edge detection, and Otsu thresholding provided a reasonable foundation for detecting structural changes in the brain. However, the CNN model was notably more effective, achieving higher accuracy in identifying subtle patterns associated with early Alzheimer's progression.

The experimental outcomes were consistent with the predictions made based on existing literature and theoretical understanding. Specifically, the CNN approach, which classified MRI images into four stages of Alzheimer's disease, demonstrated significant advantages over traditional image processing methods, particularly in distinguishing early stages. These results corroborate the findings of previous studies that highlight CNNs' ability to enhance diagnostic accuracy in medical imaging applications. The study's application of deep learning models effectively confirmed the practicality of CNNs in neuroimaging analysis, particularly in comparison to traditional, manually driven techniques.

The predictions and hypotheses from the literature review, especially concerning the effectiveness of image processing and CNN models, were validated through the experimental data. The CNN model's capability to classify very mild, mild, and moderate stages of Alzheimer's disease matched the expected outcomes, particularly in comparison to image processing techniques that could only differentiate between Alzheimer's and Non-Alzheimer's images. These findings are in line with other research, such as that conducted by pioneers in the field, who have shown that deep learning models outperform traditional methods in terms of sensitivity and accuracy for early-stage detection [52].

The framework developed for analyzing MRI scans through both image processing and deep learning models offers a valuable foundation for future research and development in Alzheimer's detection. The study's comprehensive approach to testing various parameters, such as pixel distribution and neural network classification, contributes to the potential establishment of clinical standards and best practices for neuroimaging-based diagnosis. Moreover, the research adhered to ethical guidelines, ensuring the responsible use of patient data and compliance with regulations regarding medical research.

In summary, the application of CNNs in the diagnosis of Alzheimer's disease is not only a significant advancement in medical technology but also aligns with emerging trends in artificial intelligence and healthcare. The study highlights the importance of adopting advanced, eco-friendly technologies in medical diagnostics, emphasizing the role of AI in enhancing early diagnosis capabilities. These findings have important implications for industry stakeholders, including healthcare providers and policymakers, as they pave the way for more accurate, efficient, and ethical approaches to Alzheimer's disease detection and patient care.

### 5.3 Summary

The study demonstrates the effectiveness of using both image processing techniques and Convolutional Neural Networks (CNNs) for detecting Alzheimer's Disease (AD) from MRI brain scans. The image processing approach, which involved methods like Gaussian blur and pixel analysis, proved useful in differentiating between demented and non-demented brains. However, it was limited in distinguishing between specific stages of AD. In contrast, the CNN model, trained to classify MRI images into Mild Demented, Moderate Demented, Non-Demented, and Very Mild Demented categories, achieved significantly higher accuracy (93.8%), precision, and recall metrics, making it a more reliable tool for early-stage AD detection.

The experimental results align with previous research, confirming CNNs' superiority over traditional image processing techniques in identifying subtle patterns associated with Alzheimer's progression. The study reinforces the potential of deep learning in medical imaging, emphasizing its ability to enhance diagnostic accuracy and efficiency. By automating and improving the early detection of Alzheimer's Disease, this research lays a foundation for future clinical applications and establishes CNNs as a valuable tool in neuroimaging-based diagnostics, with promising implications for patient care and treatment planning.

# **Chapter 6**

## **Conclusion and Recommendation for Future Work**

### **6.1 Conclusion**

In conclusion, this study demonstrates the effectiveness of using both image processing techniques and Convolutional Neural Networks (CNNs) for the early diagnosis of Alzheimer's disease through MRI brain scans. While traditional image processing methods provided a basic framework for identifying major structural changes in the brain, the CNN model significantly outperformed these methods by offering higher accuracy and the ability to classify the disease into its various stages. This enhanced classification, particularly in detecting very mild and mild stages of Alzheimer's, is crucial for early intervention and treatment, which can slow disease progression and improve patient outcomes.

The findings from this research highlight the potential of CNNs as a robust diagnostic tool in clinical settings, with the ability to automate and refine the analysis of MRI scans. This study not only supports the ongoing shift toward AI-driven healthcare solutions but also emphasizes the need for further research and development to optimize CNN

models and integrate them into medical practice. By improving the early detection of Alzheimer's, this research contributes to the broader goal of advancing diagnostic accuracy, improving patient care, and paving the way for more personalized and timely therapeutic interventions.

## 6.2 Recommendation for Future Work

For future research, it is recommended to explore the integration of multimodal data sources, such as PET scans, blood tests, and genetic markers, alongside MRI scans to further enhance the diagnostic accuracy of Alzheimer's disease. While this study primarily focused on MRI data, incorporating additional diagnostic tools could provide a more comprehensive understanding of Alzheimer's and improve early detection. By combining structural and biochemical data, future work could enable more precise assessments of brain health, capturing a wider spectrum of disease indicators. This multimodal approach would offer CNN models a broader range of features to analyze, which could significantly enhance their diagnostic capabilities.

Another promising direction for future work involves the development of lightweight and scalable CNN architectures that can be applied in real-time clinical settings with limited computational resources. The CNN models used in this study, though highly accurate, demand substantial processing power, which may hinder their implementation in resource-constrained environments. Research focused on creating more efficient, optimized CNN models that retain high accuracy while reducing computational requirements will be essential to ensure the accessibility and practicality of this technology in a wider range of healthcare facilities.

Additionally, future research should aim to expand the diversity of datasets used to train CNN models. The current model's generalizability may be limited by the homogeneity of the dataset, which could affect its applicability across different demographic groups.

## Chapter 6. Conclusion and Recommendation for Future Work

---

By incorporating data from various populations, including different age groups, ethnicities, and geographical locations, researchers can improve the model's robustness and ensure it performs reliably across a broader spectrum of patients. This approach would help mitigate potential biases and enhance the overall accuracy of Alzheimer's detection across various clinical settings.

Lastly, investigating the integration of CNN models with other AI-driven healthcare tools could be highly beneficial. Combining CNNs with machine learning algorithms that predict disease progression, or using natural language processing to analyze patient history, could lead to more comprehensive diagnostic frameworks. This integration would not only enhance early detection but also provide valuable insights into the disease's progression, enabling more personalized treatment plans. These advancements could greatly enhance the management of Alzheimer's disease and lead to more precise and effective patient care.

# Chapter 7

## Appendix

```
1 import cv2
2 import numpy as np
3 from PIL import Image, ImageDraw
4 import os
5 import time
6 from skimage import filters
7 from openpyxl import Workbook
8 from multiprocessing import Pool, cpu_count, freeze_support
9 from openpyxl.utils import get_column_letter
10
11
12 # Function to resize the image, outline it, and apply edge detection
13 def resize_and_outline_image(image_path, outlined_folder):
14     image = cv2.imread(image_path)
15     scale_percent = 200
16     width = int(image.shape[1] * scale_percent / 100)
17     height = int(image.shape[0] * scale_percent / 100)
18     resized_image = cv2.resize(image, (width, height), interpolation
19                               =cv2.INTER_LINEAR)
20     gray_image = cv2.cvtColor(resized_image, cv2.COLOR_BGR2GRAY)
```

## Chapter 7. Appendix

---

```
20     blurred_image = cv2.GaussianBlur(gray_image, (5, 5), 0)
21     edges = cv2.Canny(blurred_image, 50, 150)
22     contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.
23                                     CHAIN_APPROX_SIMPLE)
24     cv2.drawContours(resized_image, contours, -1, (0, 255, 0), 2)
25     outlined_image_filename = os.path.join(outlined_folder, os.path.
26                                             basename(image_path).split('.')[0] + '_outlined.jpg')
27     cv2.imwrite(outlined_image_filename, resized_image)
28
29     return outlined_image_filename
30
31
32 # Function to analyze the MRI image and apply Otsu thresholding
33 def analyze_brain_mri(image_path, binary_folder):
34
35     image = cv2.imread(image_path)
36
37     gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
38
39     threshold_value = filters.threshold_otsu(gray_image)
40
41     #threshold_value = 50 # Override threshold if needed
42
43     binary_image = gray_image > threshold_value
44
45     binary_image_8bit = (binary_image * 255).astype(np.uint8)
46
47     binary_image_filename = os.path.join(binary_folder, os.path.
48                                         basename(image_path).split('.')[0] + '_binary.png')
49
50     cv2.imwrite(binary_image_filename, binary_image_8bit)
51
52     return binary_image_filename
53
54
55 # Function to process images using bicubic interpolation
56 def process_images_bicubic(input_directory, output_directory_base):
57
58     try:
59
60         start_time = time.time()
61
62         output_bicubic_directory = os.path.join(
63             output_directory_base, "bicubic")
64
65         os.makedirs(output_bicubic_directory, exist_ok=True)
66
67         total_images = 0
68
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
```

## Chapter 7. Appendix

---

```
48     for file_name in os.listdir(input_directory):
49         if file_name.endswith('.jpg', '.png', '.jpeg'):
50             input_image_path = os.path.join(input_directory,
51                 file_name)
52
53             output_bicubic_file = os.path.join(
54                 output_bicubic_directory, file_name)
55
56             image = Image.open(input_image_path)
57             resized_image = image.resize((512, 512), Image.
58                 LANCZOS)
59
60             scale_factor = 4
61             new_width = int(resized_image.width * scale_factor)
62             new_height = int(resized_image.height * scale_factor)
63         )
64
65             resized_image_bicubic = resized_image.resize((
66                 new_width, new_height), Image.BICUBIC)
67
68             resized_image_np = np.array(resized_image_bicubic.
69                 convert('L'))
70
71             blurred_image_bicubic = cv2.GaussianBlur(
72                 resized_image_np, (5, 5), 0)
73
74             edges = cv2.Canny(blurred_image_bicubic, threshold1
75 =30, threshold2=100)
76
77             kernel = np.ones((5, 5), np.uint8)
78
79             edges_dilated = cv2.dilate(edges, kernel, iterations
80 =1)
81
82             edges_closed = cv2.morphologyEx(edges_dilated, cv2.
83                 MORPH_CLOSE, kernel)
```

## Chapter 7. Appendix

---

```
69         contours, _ = cv2.findContours(edges_closed, cv2.
70                                         RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
71
72         resized_image_with_border = resized_image_bicubic.
73         convert("RGB")
74
75         draw = ImageDraw.Draw(resized_image_with_border)
76
77         min_contour_area = 300
78         border_thickness = 5
79
80         for contour in contours:
81             if cv2.contourArea(contour) > min_contour_area:
82                 points = [tuple(point[0]) for point in
83                           contour]
84
85                 for i in range(-border_thickness,
86                                 border_thickness + 1):
87                     for j in range(-border_thickness,
88                                     border_thickness + 1):
89                         offset_points = [(x + i, y + j) for
90                               (x, y) in points]
91
92                         draw.polygon(offset_points, outline
93 = (255, 0, 0))
94
95
96                         final_image = resized_image_with_border.resize((512,
97                                         512), Image.BICUBIC)
98
99                         final_image.save(output_bicubic_file)
100
101
102                         total_images += 1
103
104
105                         end_time = time.time()
106
107                         execution_time = end_time - start_time
108
109                         print(f"Completed in {execution_time:.2f} seconds. Total
110 Images: {total_images}")
```

## Chapter 7. Appendix

---

```
92     except Exception as e:  
93         print(f"Failed: {str(e)}")  
94  
95 # Function to add red border to the image based on the processed  
96 # bicubic images  
97 def process_image_with_red_border(image_path, save_path):  
98     image = cv2.imread(image_path)  
99     gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)  
100    _, binary_image = cv2.threshold(gray_image, 1, 255, cv2.  
101        THRESH_BINARY)  
102    contours, _ = cv2.findContours(binary_image, cv2.RETR_EXTERNAL,  
103        cv2.CHAIN_APPROX_SIMPLE)  
104    mask = np.zeros_like(image)  
105    cv2.drawContours(mask, contours, -1, (255, 255, 255), thickness=  
106        cv2.FILLED)  
107    inverted_mask = cv2.bitwise_not(mask)  
108    image[inverted_mask[:, :, 0] == 255] = [0, 0, 255]  
109    cv2.imwrite(save_path, image)  
110  
111 # Function to count black and white pixels in an image  
112 def count_black_and_white_pixels(image_path):  
113     image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)  
114     total_pixels = image.size  
115     black_pixels = np.sum(image == 0)  
116     white_pixels = np.sum(image == 255)  
117     black_percentage = (black_pixels / total_pixels) * 100  
118     white_percentage = (white_pixels / total_pixels) * 100  
119     return black_pixels, white_pixels, black_percentage,  
120         white_percentage  
121  
122 # Function to save report to Excel with additional calculations  
123 def save_to_excel(report_data, excel_file_path):
```

## Chapter 7. Appendix

---

```
119 workbook = Workbook()
120 sheet = workbook.active
121 sheet.append(["Image", "Black Pixels", "White Pixels", "Black
Pixel Percentage", "White Pixel Percentage", "Overall Black Pixel
Percentage", "Demented/Non-Demented"])
122
123 for i, data in enumerate(report_data, start=2):
124     sheet.append(data)
125     # Add the formula for black pixel percentage to total pixels
126     # in the new column (D)
127     sheet[f"F{i}"] = f'=B{i}/(B{i}+C{i})*100"
128     # Add the condition for "Demented" or "NON DEMENTED" in the
129     # new column (E)
130     sheet[f"G{i}"] = f'=IF(F{i}>7.75, "DEMENTED", "NON DEMENTED
")'
131
132     # Add summary rows for NONDEMENTED and DEMENTED based on the new
133     # column (D)
134     total_rows = len(report_data) + 1
135     non_demented_formula = f'=COUNTIFS(F2:F{total_rows}, ">0",
F2:F{total_rows}, "<7.75")'
136     demented_formula = f'=COUNTIFS(F2:F{total_rows}, ">7.75",
F2:F{total_rows}, "<40")'
137
138     sheet[f"A{total_rows+1}"] = "NONDEMENTED"
139     sheet[f"B{total_rows+1}"] = non_demented_formula
140
141     sheet[f"A{total_rows+2}"] = "DEMENTED"
142     sheet[f"B{total_rows+2}"] = demented_formula
143
144     # Save the workbook
145     workbook.save(excel_file_path)
```

## Chapter 7. Appendix

---

```
143     print(f"Excel report saved at: {excel_file_path}")  
144  
145  
146  
147 # Function to process the folder of images  
148 def process_folder(input_folder, outlined_folder, binary_folder):  
149     os.makedirs(outlined_folder, exist_ok=True)  
150     os.makedirs(binary_folder, exist_ok=True)  
151  
152     for filename in os.listdir(input_folder):  
153         if filename.endswith(".jpg") or filename.endswith(".png"):  
154             image_path = os.path.join(input_folder, filename)  
155  
156             # Resize and outline the image  
157             outlined_image_path = resize_and_outline_image(  
158                 image_path, outlined_folder)  
159  
160             # Perform MRI analysis and save the binary image  
161             analyze_brain_mri(outlined_image_path, binary_folder)  
162  
163 def main():  
164     input_folder = "file path"  
165  
166     outlined_folder = "file path"  
167  
168     binary_folder = "file path"  
169  
170  
171     process_folder(input_folder, outlined_folder, binary_folder)  
172  
173     input_directory_bicubic = binary_folder
```

## Chapter 7. Appendix

---

```
174
175     output_directory_base = "file path"
176
177     process_images_bicubic(input_directory_bicubic,
178                             output_directory_base)
179
180     global bicubic_folder
181     bicubic_folder = os.path.join(output_directory_base, "bicubic")
182     red_border_folder = os.path.join(output_directory_base, "
183                                     red_border")
184     os.makedirs(red_border_folder, exist_ok=True)
185
186
187     # Report data for Excel
188     report_data = []
189
190     for file_name in os.listdir(bicubic_folder):
191         if file_name.endswith(('.jpg', '.png', '.jpeg')):
192             image_path = os.path.join(bicubic_folder, file_name)
193             save_path = os.path.join(red_border_folder, file_name)
194             process_image_with_red_border(image_path, save_path)
195
196             black_pixels, white_pixels, black_percentage,
197             white_percentage = count_black_and_white_pixels(save_path)
198             report_data.append([file_name, black_pixels,
199                                 white_pixels, black_percentage, white_percentage])
200
201
202     # Save report to Excel
203     excel_file_path = os.path.join(output_directory_base, "demented.
204                                   xlsx")
205
206     save_to_excel(report_data, excel_file_path)
207
208
209     if __name__ == "__main__":
210
```

## Chapter 7. Appendix

---

```
201     freeze_support() # For Windows compatibility  
202  
main()
```

Listing 7.1: Python Algorithm

# References

- [1] D. S. Knopman, H. Amieva, R. C. Petersen, G. Chételat, D. M. Holtzman, B. T. Hyman, R. A. Nixon, and D. T. Jones, “Alzheimer disease,” *Nat Rev Dis Primers*, vol. 7, no. 1, p. 33, 2021.
- [2] G. N. H. Shetty, H. Surlekar, “Alzheimer’s diseases detection by using convolution neural network conference name: 6th international conference on information systems and computer networks (iscon),” pp. 1–5, 2023.
- [3] J. R. Langerman and Haya, “Alzheimer’s disease – why we need early diagnosis,” *Degenerative Neurological and Neuromuscular Disease*, vol. 9, pp. 123–130, 2019. [Online]. Available: <https://www.tandfonline.com/doi/abs/10.2147/DNND.S228939>
- [4] S. Mohammadi, S. Ghaderi, and F. Fatehi, “Mri biomarkers and neuropsychological assessments of hippocampal and parahippocampal regions affected by als: A systematic review,” *CNS Neuroscience Therapeutics*, vol. 30, no. 2, p. e14578, 2024. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cns.14578>
- [5] J. Rose, “New approach could stop alzheimer’s disease from progressing,” 2015. [Online]. Available: <https://gazettereview.com/2015/04/new-approach-could-stop-alzheimers-disease-from-progressing/>
- [6] N. Wang, Y. Zhang, L. Xu, and S. Jin, “Relationship between alzheimer’s disease and the immune system: A meta-analysis of differentially expressed genes,” *Frontiers in Neuroscience*, vol. Lausanne, 2019, copyright - © 2019. This work is licensed under <http://creativecommons.org/licenses/by/4.0/> (the “License”). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License. SubjectsTermNotLitGenreText - China. [Online]. Available: [http://ezproxy.aut.ac.nz/login?url=https://www.proquest.com/scholarly-journals/relationship-between-alzheimer-s-disease-immune/docview/2306517186/se-2?accountid=8440https://resolver.ebscohost.com/openurl?ctx\\_ver=Z39.88-2004&ctx\\_enc=info:ofi/enc:UTF-8&rfr\\_id=info:sid/ProQ%3Apubliccontent&rft\\_val\\_fmt=info:ofi/fmt:kev:mtx:journal&rft.genre=article&rft.jtitle=Frontiers+in+Neuroscience&rft.atitle=Relationship+Between+Alzheimer%26rsquo%3Bs+Disease+and+the+Immune+](http://ezproxy.aut.ac.nz/login?url=https://www.proquest.com/scholarly-journals/relationship-between-alzheimer-s-disease-immune/docview/2306517186/se-2?accountid=8440https://resolver.ebscohost.com/openurl?ctx_ver=Z39.88-2004&ctx_enc=info:ofi/enc:UTF-8&rfr_id=info:sid/ProQ%3Apubliccontent&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&rft.genre=article&rft.jtitle=Frontiers+in+Neuroscience&rft.atitle=Relationship+Between+Alzheimer%26rsquo%3Bs+Disease+and+the+Immune+)

## REFERENCES

---

- System%3A+A+Meta-Analysis+of+Differentially+Expressed+Genes&rft.au=Wang%2C+Nan%3BZhang%2C+Ying%3BXu%2C+Li%3BJin%2C+Shuilin&rft.aulast=Wang&rft.aufirst=Nan&rft.date=2019-01-17&rft.volume=&rft.issue=&rft.spage=&rft.isbn=&rft.btitle=&rft.title=Frontiers+in+Neuroscience&rft.issn=16624548&rft\_id=info:doi/10.3389%2Ffnins.2018.01026
- [7] A. Chandra, G. Dervenoulas, and M. Politis, "Magnetic resonance imaging in alzheimer's disease and mild cognitive impairment," *J Neurol*, vol. 266, no. 6, pp. 1293–1302, 2019, 1432-1459 Chandra, Avinash Orcid: 0000-0002-3415-0942 Dervenoulas, George Politis, Marios Alzheimer's Disease Neuroimaging Initiative U01 AG024904/AG/NIA NIH HHS/United States Journal Article Systematic Review Germany 2018/08/19 J Neurol. 2019 Jun;266(6):1293-1302. doi: 10.1007/s00415-018-9016-3. Epub 2018 Aug 17.
  - [8] J.-W. Baek and K. Chung, "Dementia prediction support model using regression analysis and image style transfer," *Applied Sciences*, vol. 12, no. 7, p. 3536, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/7/3536>
  - [9] A. Juganavar, A. Joshi, and T. Shegekar, "Navigating early alzheimer's diagnosis: A comprehensive review of diagnostic innovations," *Cureus*, vol. 15, no. 9, p. e44937, 2023, 2168-8184 Juganavar, Anup Joshi, Abhishek Shegekar, Tejas Journal Article Review United States 2023/10/11 Cureus. 2023 Sep 9;15(9):e44937. doi: 10.7759/cureus.44937. eCollection 2023 Sep.
  - [10] psychdb, "Mini-mental status exam (mmse)," June 5 2021 2021. [Online]. Available: <https://www.psychdb.com/cognitive-testing/mmse#resources>
  - [11] H. Australia, "Mini-mental state examination (mmse)," 2024. [Online]. Available: <https://www.healthdirect.gov.au/mini-mental-state-examination-mmse>
  - [12] C. Carnero-Pardo, "Should the mini-mental state examination be retired?" *Neurología (English Edition)*, vol. 29, no. 8, pp. 473–481, 2014. [Online]. Available: <https://www.elsevier.es/en-revista-neurologia-english-edition--495-articulo-should-mini-mental-state-examination-be-S2173580814001217>
  - [13] S. Brazil, "The cognitive subscale of the "alzheimer's disease assessment scale" in a brazilian sample," *Sielo Brazil*, vol. 34, 2021. [Online]. Available: <https://doi.org/10.1590/S0100-879X2001001000009>
  - [14] J. K. Kueper, M. Speechley, and M. Montero-Odasso, "The alzheimer's disease assessment scale-cognitive subscale (adas-cog): Modifications and responsiveness in pre-dementia populations. a narrative review," *J Alzheimers Dis*, vol. 63, no. 2, pp. 423–444, 2018, 1875-8908 Kueper, Jacqueline K Speechley, Mark Montero-Odasso, Manuel CIHR/Canada Journal Article Research Support, Non-U.S. Gov't Review Netherlands 2018/04/18 J Alzheimers Dis. 2018;63(2):423-444. doi: 10.3233/JAD-170991.

## REFERENCES

---

- [15] ——, “The alzheimer’s disease assessment scale-cognitive subscale (adas-cog): Modifications and responsiveness in pre-dementia populations. a narrative review,” *J Alzheimers Dis*, vol. 63, no. 2, pp. 423–444, 2018, 1875-8908 Kueper, Jacqueline K Speechley, Mark Montero-Odasso, Manuel CIHR/Canada Journal Article Research Support, Non-U.S. Gov’t Review Netherlands 2018/04/18 J Alzheimers Dis. 2018;63(2):423-444. doi: 10.3233/JAD-170991.
- [16] nihtoolbox, “Rey auditory verbal learning test,” *nihtoolbox*, 2024. [Online]. Available: <https://nihtoolbox.org/test/rey-auditory-verbal-learning-test/>
- [17] W. Medicine, “Damage early in alzheimer’s disease id’d via novel mri approach,” *WashU Medicine*, 2022. [Online]. Available: <https://medicine.washu.edu/news/damage-early-in-alzheimers-disease-idd-via-novel-mri-approach/>
- [18] L. Chouliaras and J. T. O’Brien, “The use of neuroimaging techniques in the early and differential diagnosis of dementia,” *Molecular Psychiatry*, vol. 28, no. 10, pp. 4084–4097, 2023. [Online]. Available: <https://doi.org/10.1038/s41380-023-02215-8>
- [19] P. M. Thompson, K. M. Hayashi, G. de Zubicaray, A. L. Janke, S. E. Rose, J. Semple, D. Herman, M. S. Hong, S. S. Dittmer, D. M. Doddrell, and A. W. Toga, “Dynamics of gray matter loss in alzheimer’s disease,” *The Journal of Neuroscience*, vol. 23, no. 3, pp. 994–1005, 2003. [Online]. Available: <https://www.jneurosci.org/content/jneuro/23/3/994.full.pdf>
- [20] M. L. F. Balthazar, C. L. Yasuda, F. R. Pereira, T. Pedro, B. P. Damasceno, and F. Cendes, “Differences in grey and white matter atrophy in amnestic mild cognitive impairment and mild alzheimer’s disease,” *European Journal of Neurology*, vol. 16, no. 4, pp. 468–474, 2009. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-1331.2008.02408.x>
- [21] E. H. Alyoubi, K. M. Moria, J. S. Alghamdi, and H. O. Tayeb, “An optimized deep learning model for predicting mild cognitive impairment using structural mri,” *Sensors*, vol. 23, no. 12, p. 5648, 2023. [Online]. Available: <https://www.mdpi.com/1424-8220/23/12/5648>
- [22] A. Khalid, E. M. Senan, K. Al-Wagih, M. M. A. Al-Azzam, and Z. M. Alkhraisha, “Automatic analysis of mri images for early prediction of alzheimer’s disease stages based on hybrid features of cnn and handcrafted features,” *Diagnostics (Basel)*, vol. 13, no. 9, 2023, 2075-4418 Khalid, Ahmed Senan, Ebrahim Mohammed Al-Wagih, Khalil Al-Azzam, Mamoun Mohammad Ali Alkhraisha, Ziad Mohammad (NU/DRP/SERC/12/7)./This research has been funded by the Deanship of Scientific Research at Najran Uni-versity, Kingdom of Saudi Arabia, through a grant code/ Journal Article Switzerland 2023/05/13 Diagnostics (Basel). 2023 May 8;13(9):1654. doi: 10.3390/diagnostics13091654.

## REFERENCES

---

- [23] V. S. Design, “Filtering techniques eliminate gaussian image noise,” *Vision Systems Design*, 2019. [Online]. Available: <https://www.vision-systems.com/home/article/14174546/filtering-techniques-eliminate-gaussian-image-noise>
- [24] . W. R. E. Gonzalez, R. C., “Digital image processing,” vol. 4, pp. 181–183, 2018.
- [25] L. Pei, M. Ak, N. H. M. Tahon, S. Zenkin, S. Alkarawi, A. Kamal, M. Yilmaz, L. Chen, M. Er, N. Ak, and R. Colen, “A general skull stripping of multiparametric brain mrис using 3d convolutional neural network,” *Scientific Reports*, vol. 12, no. 1, p. 10826, 2022. [Online]. Available: <https://doi.org/10.1038/s41598-022-14983-4>
- [26] J. Swiebocka-Wiek, “Skull stripping for mri images using morphological operators,” ser. Computer Information Systems and Industrial Management. Springer International Publishing, Conference Proceedings, pp. 172–182.
- [27] D. Gupta, R. Loane, S. Gayen, and D. Demner-Fushman, “Medical image retrieval via nearest neighbor search on pre-trained image features,” *Knowl Based Syst*, vol. 278, 2023, 1872-7409 Gupta, Deepak Loane, Russell Gayen, Soumya Demner-Fushman, Dina Z99 LM999999/ImNIH/Intramural NIH HHS/United States Journal Article Netherlands 2023/10/02 Knowl Based Syst. 2023 Oct 25;278:110907. doi: 10.1016/j.knosys.2023.110907. Epub 2023 Aug 18.
- [28] TheAIlearner, “Image processing — nearest neighbour interpolation,” 2018. [Online]. Available: <https://theailearner.com/2018/12/29/image-processing-nearest-neighbour-interpolation/>
- [29] Simplilearn, “Bilinear interpolation,” 2023. [Online]. Available: <https://www.simplilearn.com/tutorials/statistics-tutorial/bilinear-interpolation>
- [30] S. McHugh, “Image interpolation,” 2024.
- [31] K. J. M. H. M. A. Z. K. N. M. Donya Khaledyan, Abdolah Amirany, “Low-cost implementation of bilinear and bicubic image interpolation for real-time image super-resolution,” 2020. [Online]. Available: <https://arxiv.org/abs/2009.09622>
- [32] TheAIlearner, “Image processing: Bicubic interpolation,” December 29, 2018. [Online]. Available: <https://theailearner.com/2018/12/29/image-processing-bicubic-interpolation/>
- [33] G. Leedham, S. Varma, A. Patankar, and V. Govindaraju, “Separating text and background in degraded document images - a comparison of global thresholding techniques for multi-stage thresholding,” in *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, Conference Proceedings, pp. 244–249.

## REFERENCES

---

- [34] T. Malche, “What is thresholding in image processing? a guide.” 2024. [Online]. Available: <https://blog.roboflow.com/image-thresholding/#:~:text=Thresholding%20in%20image%20processing%20is%20a%20technique%20used,intensity%20values%20are%20below%20or%20above%20the%20threshold>.
- [35] X. Xu, S. Xu, L. Jin, and E. Song, “Characteristic analysis of otsu threshold and its applications,” *Pattern Recognition Letters*, vol. 32, no. 7, pp. 956–961, 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167865511000365>
- [36] T. Y. Goh, S. N. Basah, H. Yazid, M. J. Aziz Safar, and F. S. Ahmad Saad, “Performance analysis of image thresholding: Otsu technique,” *Measurement*, vol. 114, pp. 298–307, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0263224117306243>
- [37] D. Bradley and G. Roth, “Adaptive thresholding using the integral image,” *Journal of Graphics Tools*, vol. 12, no. 2, pp. 13–21, 2007, doi: 10.1080/2151237X.2007.10129236. [Online]. Available: <https://doi.org/10.1080/2151237X.2007.10129236>
- [38] ProjectPro, “What is adaptive thresholding in opencv,” 2023. [Online]. Available: <https://www.projectpro.io/recipes/what-is-adaptive-thresholding-opencv>
- [39] R. A. A S and S. Gopalan, “Comparative analysis of eight direction sobel edge detection algorithm for brain tumor mri images,” *Procedia Computer Science*, vol. 201, pp. 487–494, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050922004768>
- [40] W. contributors, “Canny edge detector.” [Online]. Available: [https://en.wikipedia.org/wiki/Canny\\_edge\\_detector](https://en.wikipedia.org/wiki/Canny_edge_detector)
- [41] L. Xuan and Z. Hong, “An improved canny edge detection algorithm,” pp. 275–278.
- [42] R. Thakur, “Beginner’s guide to vgg16 implementation in keras,” *BuiltIn*, 2024. [Online]. Available: <https://builtin.com/machine-learning/vgg16>
- [43] M. J. Y. Sutayco and M. V. C. Caya, “A comparative study of mobilenetv2 and vgg-16 convolutional neural network architectures for identification of medicinal mushrooms,” in *2023 IEEE 15th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, Conference Proceedings, pp. 1–6.
- [44] L. R. Indra Mahakalanda, “Vgg-16,” *ScienceDirect*, 2022. [Online]. Available: <https://www.sciencedirect.com/topics/computer-science/vgg-16>
- [45] A. Z. Smola, Z. C. Lipton, M. Li, and A. J., “Alexnet,” 2021. [Online]. Available: [https://d2l.ai/chapter\\_convolutional-modern/alexnet.htm](https://d2l.ai/chapter_convolutional-modern/alexnet.htm)

## REFERENCES

---

- [46] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: concepts, cnn architectures, challenges, applications, future directions,” *Journal of Big Data*, vol. 8, no. 1, p. 53, 2021. [Online]. Available: <https://doi.org/10.1186/s40537-021-00444-8>
- [47] GeeksforGeeks, “MobileNet v2 architecture in computer vision,” 2024. [Online]. Available: <https://www.geeksforgeeks.org/mobilenet-v2-architecture-in-computer-vision/#mobilenet-v2-architecture>
- [48] ResearchGate, “Design space exploration of a sparse mobilenetv2 using high-level synthesis and sparse matrix techniques on fpgas - scientific figure on researchgate,” 2024. [Online]. Available: [https://www.researchgate.net/figure/The-architecture-of-MobileNetV2-DNN\\_fig1\\_361260658](https://www.researchgate.net/figure/The-architecture-of-MobileNetV2-DNN_fig1_361260658)
- [49] D. set, “Alzheimer mri preprocessed dataset,” 2022. [Online]. Available: <https://www.kaggle.com/datasets/sachinkumar413/alzheimer-mri-dataset>
- [50] A. Baratloo, M. Hosseini, A. Negida, and G. El Ashal, “Part 1: Simple definition and calculation of accuracy, sensitivity and specificity,” *Emerg (Tehran)*, vol. 3, no. 2, pp. 48–9, 2015, 2345-4571 Baratloo, Alireza Hosseini, Mostafa Negida, Ahmed El Ashal, Gehad Journal Article Iran 2015/10/27 Emerg (Tehran). 2015 Spring;3(2):48-9.
- [51] N. U. Huda, “Precision and recall curves,” 2024. [Online]. Available: <https://www.blog.trainindata.com/precision-recall-curves/#:~:text=The%20f-1%20score%20is%20the%20geometric%20average%20between,x%20precision%20X%20recall%20%2F%20%28precision%20%2B%20recall%29>
- [52] S. Sajed, A. Sanati, J. E. Garcia, H. Rostami, A. Keshavarz, and A. Teixeira, “The effectiveness of deep learning vs. traditional methods for lung disease diagnosis using chest x-ray images: A systematic review,” *Applied Soft Computing*, vol. 147, p. 110817, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1568494623008359>