DEEP LEARNING MODEL FOR PREDICTION OF PROGRESSIVE MILD COGNITIVE IMPAIRMENT TO **ALZHEIMER'S DISEASE USING STRUCTURAL MRI**

LIM BING YAN

FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

DEEP LEARNING MODEL FOR PREDICTION OF PROGRESSIVE MILD COGNITIVE IMPAIRMENT TO ALZHEIMER'S DISEASE USING STRUCTURAL MRI

LIM BING YAN

RESEARCH PROJECT SUBMITTED TO THE FACULTY OF ENGINEERING UNIVERSITY OF MALAYA, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF BIOMEDICAL ENGINEERING

FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Lim Bing Yan

Matric No: S2011146

Name of Degree: Master of Biomedical Engineering

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

Deep Learning Model for Prediction of Progressive Mild Cognitive Impairment To Alzheimer's Disease Using Structural MRI

Field of Study: Artificial Intelligence

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date: 24/9/2021

Subscribed and solemnly declared before,

Witness's Signature Date: 24/9/2021

Name:

Designation:

DEEP LEARNING MODEL FOR PREDICTION OF PROGRESSIVE MILD COGNITIVE IMPAIRMENT TO ALZHEIMER'S DISEASE USING STRUCTURAL MRI

ABSTRACT

Alzheimer's disease (AD), an irreversible neurodegenerative disorder that has caused the majority cases of dementia, wherein patients suffer progressive memory loss and cognitive function decline. Despite having no drugs for curing, early detection of AD allows the provision of preventive treatment to control the disease progression. The objective of this project is to develop a computer-aided system based on deep learning model to identify AD from cognitively normal and its early stage, mild cognitive impairment (MCI), using only structural MRI (sMRI). In this project, multiclass classification was applied. The dataset consisted of 3D T1-weighted brain sMRI images from the ADNI database containing 819 participants. A series of pre-processing methods were applied to the dataset; For example, skull stripping, bias field correction, pixel values normalisation, and data augmentation. HMRF tissue classifier was used to segment the brain MRI into 3 separate regions of grey matter, white matter, and cerebrospinal fluid. Axial brain images were extracted from the 3D MRI and being fed as input to the convolutional neural network (CNN) to perform multiclass classification of AD-CN-MCI. 3 different models were being experimented namely a CNN from scratch, VGG-16, and ResNet-50. The convolutional base of VGG-16 and ResNet-50 trained on ImageNet dataset were used as a feature extractor. Additionally, a new densely connected classifier was added on top of the convolutional base for performing classification. Using the 20% held out testing data, the performance of each model was reported and discussed. Among the 3 models, VGG-16 achieved the best testing performance with accuracy of 78.57%, precision of 73.94%, recall of 81.37%, and F1-score of 77.48%. Transfer learning technique allowed VGG-16 to achieve better performance despite a small number of data

was being used. However, the best-performed VGG-16 has performed below average in comparison to previous works. Hence, limitations and possible solutions were outlined for future improvement.

Keywords: Alzheimer's disease, deep learning, convolutional neural network, prediction, magnetic resonance imaging

MODEL PEMBELAJARAN DALAM UNTUK PREDIKASI KEMEROSOTAN KOGNITIF SEDERHANA KEPADA PENYAKIT ALZHEIMER MENGGUNAKAN MRI STRUKTUR

ABSTRAK

Penyakit Alzheimer (PA) ialah salah satu penyakit neurodegeneratif yang tidak dapat dipulihkan. Ia telah menyebabkan sebahagian besar kes demensia. Biasanya, pesakit PA mengalami kehilangan ingatan yang progresif dan penurunan fungsi kognitif. Walaupun tiada ubat untuk mengubati PA pada masa ini, pengesanan PA semasa peringkat awal boleh membenarkan rawatan pencegahan untuk mengawal perkembangan penyakit tersebut. Objektif projek ini adalah untuk mencipta sistem bantuan komputer berdasarkan model pembelajaran dalam untuk membezakan PA dengan kognitif normal (KN) dan peringkat awal PA iaitu gangguan kognitif ringan (GKR) hanya menggunakan MRI struktur (sMRI). Dalam projek ini, klasifikasi multikelas telah digunakan. Dataset yang digunakan terdiri daripada gambar sMRI otak T1-weighted dalam 3 dimensi yang dimuat turun dari database ADNI yang mengandungi 819 peserta. Selepas itu, semua data telah diproses dengan pelbagai cara pemprosesan imej. Contohnya dalam istilah teknikal Bahasa Inggeris seperti skull stripping, bias field correction, pixel values normalisation dan data augmentation. Selain itu, HMRF tissue classifier digunakan untuk membahagikan sMRI otak kepada 3 segmen iaitu bahan kelabu, bahan putih dan cecair serebrospinal. Imej otak pandangan 'axial' diekstrak daripada MRI 3D dan digunakan sebagai input untuk convolutional neural network (CNN) untuk melakukan klasifikasi multikelas PA-KN-GKR. 3 model yang berbeza telah dieksperimen iaitu CNN, VGG-16, dan ResNet-50. Convolutional base VGG-16 dan ResNet-50 yang dilatih pada dataset ImageNet telah diaplikasikan sebagai feature extractor. Tambahan pula, densely connected classifier yang baru ditambahkan di atas convolutional base untuk melakasanakan klasifikasi. Prestasi setiap model berdasarkan keputusan klasifikasi

menggunakan data ujian telah dilaporkan. Dalam kalangan 3 model yang digunakan, VGG-16 mencapai prestasi yang terbaik (accuracy 78.57%, precision 73.94%, recall 81.37%, dan skor F1 77.48%). Teknik pembelajaran transfer membolehkan VGG-16 mencapai prestasi yang lebih baik walaupun sebilangan kecil data digunakan. Jika dibandingkan dengan karya sebelum ini, VGG-16 yang dicadangkan mempunyai prestasi di bawah purata. Oleh itu, kekurangan model VGG-16 yang dicadangkan dan cara-cara penyelesaian telah dijelaskan untuk penambahbaikan pada masa hadapan.

Keywords: penyakit Alzheimer, pembelajaran dalam, convolutional neural network, ramalan, MRI

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Ir. Dr. Lai Khin Wee for his invaluable advice and guidance throughout the whole research project. My grateful thanks also go to my parents and younger brother for their unconditional support throughout my master's degree study. Furthermore, I would like to thank my girlfriend who had given moral support and encouragement, which contributed to the successful completion of this project. Last but not least, I extend my thanks to all of my friends and colleagues for their unlimited support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

1D	:	One-dimensional
2D	:	Two-dimensional
3D	:	Three-dimensional
3DMPRAGE	:	Three-dimensional magnetisation-prepared rapid gradient-echo
AD	:	Alzheimer's disease
ADNI	:	Alzheimer's Disease Neuroimaging Initiative
AI	:	Artificial intelligence
ApoE	:	Apolipoprotein
CSF	:	Cerebrospinal fluid
CN	:	Cognitive normal
CNN	:	Convolutional neural network
CT	:	Computed tomography
DL	:	Deep learning
DTI	:	Diffusion tensor imaging
EEG	:	Electroencephalogram
FC	:	Fully connected
FOV	:	Field of view
fMRI	:	Functional magnetic resonance imaging
GPU	:	Graphics processing unit
GM	:	Grey matter
HMRF	:	Hidden Markov random field
MCI	:	Mild cognitive impairment
ML	:	Machine learning
MMSE	:	Mini-Mental State Examination

MRI	:	Magnetic resonance imaging
NIfTI	:	Neuroimaging Informatics Technology Initiative
PET	:	Positron emission tomography
pMCI	:	Progressive mild cognitive impairment
ReLU	:	Rectified Linear Unit
sMCI	:	Stable mild cognitive impairment
sMRI	:	Structural magnetic resonance imaging
SVM	:	Support vector machine
TE	:	Echo time
TR	:	Repetition time
WM	:	White matter

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

1.1 Overview

Alzheimer's disease (AD) is a progressive illness that leads to neuronal loss and commonly causes dementia among the elderly. AD patients usually start to suffer from progressive memory loss and eventually with cognitive decline. Usually, it attributes to a loss of independence for AD subjects. By the year 2050, it is predicted that 1 out of 85 people in the world will suffer from AD (Brookmeyer et al., 2007). At the moment, there are approximately 90 million people identified as AD patients and the number of diseased patients is estimated to reach 300 million by 2050 (Zhu et al., 2015).

There are medications to provide temporary moderate symptomatic relief or to decrease the rate of progression of AD, wherein these treatments are found to help AD subjects by maximising their cognitive function and maintaining independence for a time. However, at present, there is no effective and safe drugs or therapies for curing AD or altering the disease process in the brain (Tatiparti et al., 2020). The search for effective strategies to treat or prevent AD remains one of the most challenging endeavours in the medical field. Hence, it is of utmost importance to identify AD at its early or prodromal stage so that patients can get treatment prior to disease progression. To date, the mainstream non-invasive clinical approach to perform prognostic prediction for AD is by way of manual inspection through analysing structural neuroimaging such as magnetic resonance imaging (MRI) or computed tomography (CT). Presently, computer-aided systems based on artificial intelligence (AI) algorithms have been implemented to perform AD diagnoses (Wen et al., 2020).

In conjunction with the swift development of AI, researchers have been using AI algorithms such as deep learning to solve sophisticated problems in various fields, especially in the medical field. Researchers have extended the usage of various deep learning models to diagnose different stages of AD. Current neuroimaging studies which

utilised computer-aided system studies have made significant progress in classifying AD and cognitively normal (CN) subjects. Even though the binary classification of AD and CN subjects has achieved outstanding performance, it is not as beneficial as predicting the early-stage conversion, which is mild cognitive impairment (MCI) to AD. The majority of the studies stopped at a binary classification without predicting whether a patient has MCI and the possibility to convert to AD.

1.2 Problem Statement

Similar to other diseases, detecting AD in its prodromal stage or anticipation of its possibilities is crucial for its treatment. The treatments are effective if the AD patients are able to receive treatment as early as possible after being suspected of manifesting AD biomarkers or symptoms. With proper treatment, a delay of 1 year of AD progression can reduce the number of diseased patients by 10% (McKhann et al., 2011). The statistic shows that AD diagnosis during its early stage is crucial in decreasing the number of patients globally.

During diagnosis of AD, neurologists have to manually analyse brain images and perform cognitive assessments in order to give an exact judgement on the symptoms and progression of AD. Owing to the fact that brain anatomy changes with subtleties can be observed years ahead before distinct biomarkers can be visualised by humans, it is realised that the human visual system is incapable of identifying subtle changes in underlying brain structure with the possibility of containing vital information about the disease state of a patient despite analysis is being done by experienced neurologists. Thus, an AI-based computer-aided system is helpful to aid neurologists in diagnosing complex brain diseases and reduce the possibility of misdiagnosis. Moreover, it is expected to decrease the workload on medical professionals and reduce the number of patient visitation and waiting times.

1.3 Aim and Objectives

This project aims to design and develop a computer-aided system based on deep learning algorithm to determine the pathological brain structural change of MRI data for predicting the early stage of AD prior to its advanced stages. To reach this aim, the following specific objectives need to be achieved.

(1) To perform novel pre-processing procedures on brain structural MRI used for training and testing the convolutional neural network.

(2) To implement convolutional neural network algorithm to perform multi-class classification (3-way) to classify cognitively normal, MCI, and AD subjects with performance evaluated by metrics such as accuracy, precision, recall, and F1-score.

(3) To compare the classification performance of best-performed model with previous literature.

1.4 Project Approach

The state-of-the-art methods for AD classifications and their limitations were reviewed, and a literature review of peer-reviewed journals and articles was conducted to obtain information and knowledge pertaining to the AD continuum. The neuroimaging dataset used in this project was acquired from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database. In specific, raw structural MRI volumes of the brain from CN, MCI, and AD subjects at baseline, 6-month, 12-month, and 24-month follow-ups were retrieved for pre-processing. The pre-processing steps include skull stripping, bias field correction, tissue segmentation, extraction of two-dimensional (2D) slices as images, pixel values normalisation and augmentation. With regard to tissue segmentation, Otsu threshold and hidden Markov random field (HMRF) tissue classifier were used to extract different regions of the brain such as white matter (WM), grey matter (GM), and cerebrospinal fluid (CSF). The dataset was utilised by 3 different convolutional neural networks (CNN) architectures for training and testing, namely a CNN from scratch,

VGG-16, and ResNet-50. At the same time, the models were improved by hyperparameters tuning to find the best combination of hyperparameters. Classifications were performed to distinguish AD subjects from CN and MCI subjects.

1.5 Project Outcome

The output of this research project is a deep learning model to aid neurologists in AD diagnosis. The automated deep learning framework accepts brain image as input and performs multi-class or 3-way classification to classify an image that comes from a subject of AD, CN, or MCI. The prediction performance was recorded using evaluation metrics, for example, accuracy, precision, recall, and F1-score.

1.6 Report Outline

The report is organised as follows. The first chapter entails the overview of AD and its diagnosis approaches, problem statement, project aim and objectives, and approach and outcome of the project. Chapter 2 consists of background information of AD, theory of deep learning, examples of AD diagnosis using AI algorithms such as machine learning and deep learning, and theory of CNN. The methodology of the project work, including data collection, step-by-step pre-processing, and proposed convolutional neural network architectures are summarised in Chapter 3. In Chapter 4, the results of prediction performance are presented in detail for comparison with in-depth discussions. Lastly, Chapter 5 includes the conclusion for this project and a brief discussion of possible future improvements.

CHAPTER 2: LITERATURE REVIEW

2.1 Alzheimer's Disease

AD is a common dementia-causing neurodegenerative disease that leads to irreversible, progressive memory loss and decline in cognitive functions. It accounts for up to 80% of all cases of dementia. In 1906, Dr Alois Alzheimer discovered AD while he was performing an autopsy on a patient who died from a distinct disease of the cerebral cortex (Hippius & Neundörfer, 2003). He observed some alterations in brain tissues that were extraordinary. Study reported that AD does affect people in the middle-aged group apart from elderly people of age above 65 years old (Grundman et al., 2004). Brookmeyer et al. (2007) projected that 1.18% of the world population would suffer from AD by 2050. At the moment, there are approximately 90 million people identified as AD patients and the number of diseased patients is estimated to hit 300 million by 2050 (Zhu et al., 2015).

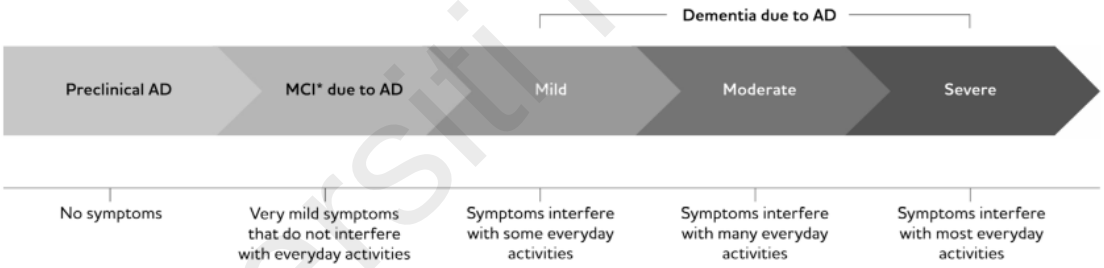

Figure 2.1: Illustration of AD continuum adopted from (Alzheimer's Association, 2020)

The most common symptoms associated with AD are progressive memory loss, language decline, and cognitive deterioration (Alzheimer's Association, 2020). The first symptom of AD varies from person to person. The majority of AD subjects begin with a decline in non-memory aspects of cognition; for instance, impaired reasoning, word-finding, and vision issues may signal early stages of AD. In the late stages of this disease, diseased subject normally suffers from complications from severe loss of brain function, for example, malnutrition, infection or dehydration, and results in death. As of now, there is no effective and safe drugs or therapies for curing AD or halting the disease from

causing neuronal damage (Tatiparti et al., 2020). However, there are existing medications to offer temporary moderate symptomatic relief or to decrease the rate of progression of AD. Study also reported that AD drugs are rudimentary symptomatic drugs. In other words, medications can only offer moderate symptomatic relief instead of halting the disease (Iyaswamy et al., 2020). These treatments are found to help AD subjects by maximising their function and maintaining independence for a time. The most challenging task along the AD continuum is to search for effective approaches to treat or prevent the disease. Thus, early detection of AD especially its prodromal stage is critical to allow delivery of appropriate treatment to delay its development before the disease progresses.

A transitional stage between normal ageing and AD, known as mild cognitive impairment (MCI), presents a higher risk of progression to AD. People who suffered from MCI has mild cognitive problems more than people of their age. However, symptoms of MCI usually do not interfere with the independence and ability of a person to carry out daily activities (National Institute of Health, 2017). Study presented that there is one-fifth of people of age over 65 years old have MCI (Alzheimer's Association, 2020). The conversion rate of MCI to AD is about 33%, and the conversion period is normally within 5 years (Ward et al., 2013). Depending on each individual, the transition is within 6 to 36 months, but the average conversion time is 18 months. Elderly MCI subjects have a higher risk for developing AD, yet there is a likelihood to revert to normal cognition.

2.1.1 Pathogenesis

AD as a neurodegenerative disorder can be characterised by neuronal loss and overall atrophy of the brain. However, the actual mechanisms and causes of the disease progression are not completely discovered. Despite the uncertainty of the underlying process of AD onset, there is a number of accepted biomarkers.

Pathologically, the progression of AD happens years before clinical manifestations. The build-up of 2 key types of protein inside the brain such as beta-amyloid (Aβ) and hyper-phosphorylated tau are the most studied biomarkers of AD. Aβ peptides are generated inherently by the human body and is the most prominent compound known to be associated to AD (Taipa, 2018). Under normal circumstances, Aβ presented in the brain is degraded and cleared. When Aβ piles up abnormally in the brain, it causes senile plaques formation originated from the amyloid-precursor protein. Owing to the toxicity of senile plaques, it deteriorates brain neurons and disrupts communication between cells. The Apolipoprotein (ApoE) genotype was found to help the breakdown of Aβ. However, the APOE ε4 allele is less efficient compared to the other allele for Aβ clearance, leading to a higher possibility of the formation of amyloid plaque (Emrani et al., 2020). Hence, the presence of ApoE ε4 is a strong risk factor for developing AD.

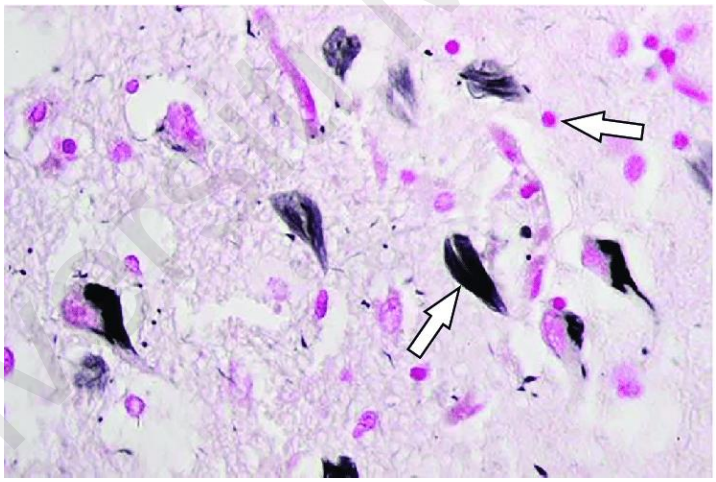

Figure 2.2: Histology of amyloid plaques (pink) and neurofibrillary tangles (black). Source: https://www.alzheimers.org.uk/

Tau protein is responsible for neuronal architecture and stability, in which it ensures the internal support and transport system of a neuron to channel nutrients and other essential materials. When an abnormal accumulation of tau proteins occurs in the neurons, tau proteins will reshape and reorganise themselves to form neurofibrillary tangles that disrupt the structure and function of neurons. The nutrients supply to the neurons will be interrupted, leading to degeneration and eventually death of neurons that can be

characterised by neuronal loss. The presence of neurofibrillary tangles was discovered during an autopsy of an AD patient (Grundke-Iqbal et al., 1986). As of now, it is accepted that the presence of the two cerebral lesions is necessary to develop AD since one does not come without the other. However, the question of which lesion comes first is still a debatable topic in the AD community.

Successive accumulation of amyloid plaques and tau protein can cause neurons death, resulting in changes in brain structure. The most affected brain regions are the hippocampus, amygdala, and entorhinal cortex. In advanced stages, AD affects the parietal, temporal, and frontal regions of the brain. As a consequence of neuron degeneration, large scale changes take place in the brain. Progression of AD stems from the hippocampus which in charge of short-term memory. Hippocampus is one of the first brain regions to reveal abnormalities in AD patients (Eman et al., 2016). It is reported that the hippocampus shrinks atypically in AD patients, wherein the shrinkage of volume is within 2.2% to 5.9% annually. In normal people, the volume of shrinkage ranges from 0.24% to 1.73% yearly. Eventually, the degeneration of neurons spreads to the rest of the brain following a centrifugal movement and reach the entire brain structure for late-stage AD. This comes with inevitable atrophy of brain tissue related to memory loss. The process causes brain atrophy or shrinkage of brain which engenders global dysfunction. For instance, the gyri, ridges of the brain become narrower and the sulci, the grooves get wider. In addition, enlargement of the fluid-filled ventricles or cavities takes place as well. The progression of the brain lesions corresponds with the manifestation of symptoms, which begin with memory problems followed by cognitive impairment.

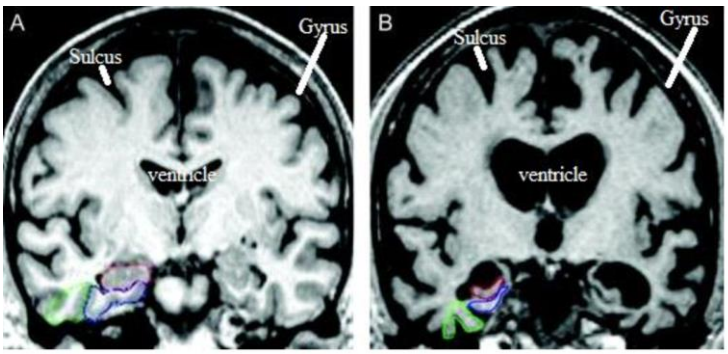

Figure 2.3: Comparison of brain MRI of (A) clinical normal and (B) AD patient adopted from (Saurabh, 2015)

Typically, changes in brain structure caused by AD occur before patients reveal amnestic symptoms (Buckner, 2004). To help AD diagnosis, functional magnetic resonance imaging (fMRI) and structural magnetic resonance imaging (sMRI) have been used for detecting abnormalities in the brain. Studies have also been done on analysing biomarkers to detect early changes in underlying brain structure for exposing early stages of AD (Mobed & Hasanzadeh, 2020; Tan et al., 2014).

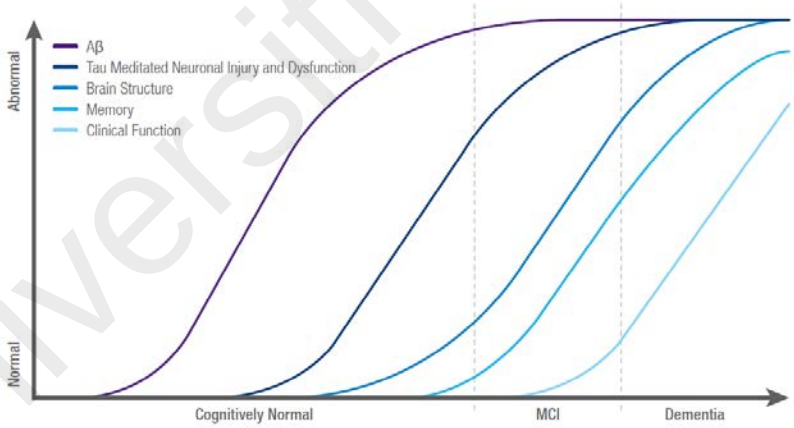

Figure 2.4: Alzheimer's disease progression curve

2.1.2 Diagnosis Techniques

There are limited information neurologists are clear about the underlying mechanisms of progression of AD. Therefore, early detection of AD is key for timely provision of treatment to ensure normalcy of life of patients as well as their caretakers and families. Generally, AD diagnosis techniques can be grouped into two main approaches, which are neuroimaging and neurological examination or cognitive tests. However, the only way

for a ‘ground truth’ AD diagnosis is through clinical autopsy, which is clinically impractical (Ebrahimighahnavieh et al., 2020).

Early-stage AD diagnosis requires the assessment of non-invasive quantitative biomarkers. Clinical examination involves performing non-invasive prognostic prediction for AD through manual inspection by analysing structural neuroimaging such as positron emission tomography (PET), CT, or MRI. Neuroimaging techniques have excellent capability to capture alterations of pathological brain structure or quantitative biomarkers associated with AD in vivo (Johnson et al., 2012). In clinical trials, out of all imaging techniques, the prevailing imaging modality is the MRI to obtain clear images of the brain owing to its advantages of high spatial resolution, non-invasive nature, wide accessibility, and moderate costs. MRI is one of the imaging techniques, which excels in using strong magnetic fields and radio waves to produce three-dimensional (3D) representations of images of internal organs and soft tissues. Clinically, MRI is utilised to visualise the anatomy and physiological processes of the body. For studying AD progression, MRI is widely used to examine changes in anatomical and functional of the brain. It is highly preferred in AD diagnosis due to the fact that the images show details of the brain topology together with the variations in brain morphology. Also, the neuropathological progression of AD patients can be identified many years prior to the onset of clinical symptoms, which has led to interest in the study of AD detection at various stages. In particular, for study using single imaging modality, sMRI is the most extensively used due to its great capability in visualising structure of the brain and identifying anatomic anomalies and lesions and provides a measure of the unavoidable brain atrophy biomarkers in AD (Folego et al., 2020). For instance, sMRI is used to measure two important biomarkers of neurodegeneration which are shrinkage in volume and change in structure of the hippocampus (Aderghal et al., 2020; Gupta et al., 2019). Furthermore, detection of volume loss and intensity changes of cerebrospinal fluid, white

matter, and grey matter are viable through sMRI (Mehmood et al., 2021; Pelkmans et al., 2019).

On the other hand, fMRI with function to indicate the changes in brain activity linked to blood flow is commonly utilised to detect changes in function, metabolism, and connectivity. This imaging technique utilises BOLD (Blood Oxygen Level Dependant) mechanism to show the intensity and pattern of human brain activity (Oghabian et al., 2010). Up to now, it is still a challenging undertaking to detect AD at its early phases owing to the subtle changes in brain structure and low distinguishability patterns that could be quantified through quantitative analysis or conventional neuroimaging.

Apart from various neuroimaging modalities, a spectrum of quantitative assessments is implemented clinically or being used in conjunction with neuroimaging modality for multi-modalities study. In addition to multiple neuroimaging modalities, multi-modalities study may consider other factors such as demographic data (age, gender, educational level, etc.), speech pattern, genes, electroencephalogram (EEG), retinal abnormalities, postural kinematic analysis, CSF biomarkers, neuropsychological measures, Mini-Mental State Examination (MMSE) score, and logical memory test (Cuingnet et al., 2011). The MMSE is usually performed as a short cognitive function screening test to predict AD (Arevalo-Rodriguez et al., 2015). The maximum score for the test is 30 points. If an individual has MMSE score that reduces periodically, the person is affected by AD.

Multiple modalities-based classifier has shown better classification performance in comparison to single modality-based classifier despite the increased complexity in training deep learning model. Studies hypothesised that this is due to the complexity and heterogeneity of predicting AD (Liu et al., 2020; Zhang et al., 2019). However, there is a trade-off between improved prediction performance and computational power as well as the monetary cost in acquiring additional biomarkers. According to study, MRI is considered the ubiquitous mode of neuroimaging for study of AD (Ebrahimighahnavieh

et al., 2020). Despite the fact that numerous studies have argued that MRI is more discriminative as compared to diffusion tensor imaging (DTI) or PET, it is argued that MRI is slightly less discriminative than PET or as discriminative as PET. Also, there are studies that suggested fMRI or DTI provide the most helpful evidence for AD study. The argument of which mode of neuroimaging is the most discriminative remains a controversial topic. Nevertheless, the most effective way is to use a combination of multiple modalities as AD heterogeneous, but this will evidently incur higher cost.

2.2 State-of-the-art

With the recent increasing development of AI, many researchers have gained interest to apply AI technology in the medical field. AI is seen as a new technology that could provide solutions in designing a clinical prognostic or diagnostic tool for AD with high accuracy.

2.2.1 Machine Learning

In the field of neuroimaging, machine learning (ML) algorithms have advanced the development of AD research rapidly (Weiner et al., 2017). Studies have introduced computer-aided systems incorporating ML techniques to decode the disease states from MRI (Gupta et al., 2019; Rallabandi et al., 2020). ML model is trained to analyse high dimensional data by learning the data complex relationships and patterns and recognise certain relationships or patterns from the data being tested. Likewise, imaging biomarkers or features can be captured by ML algorithms to predict AD. Different data types have been implemented as input for ML to predict AD, such as neuroimaging data, biological data, and neuropsychological test results. Typically, a ML framework includes several components, including features extraction, features training, and classification. ML is highly automated in which it classifies input data through examining the similarities or disparities of testing images with labelled images regardless the subtle differences in brain images that are hardly seen by the human visual system (Battista et al., 2020).

Within the recent 10 years, a plenitude of AD diagnosis studies dependent on ML technique shared the same goal to achieve classification accuracy as high as possible in the task of binary classification of AD and CN subjects. Researchers have addressed this problem using different strategies reliant on ML technique. For instance, supervised and semi-supervised learning studies with MRI data as input are the two main approaches. Of the numerous methods being experimented, Support Vector Machine (SVM) is the renowned method and offered excellent classification result. Yet, SVM is being condemned for performing poorly on unprocessed data and hand-crafted features is highly required in SVM-based model (Oh et al., 2019). Despite ML models that classify AD and CN classes have achieved promising accuracy within 80% and 95%, the following challenge persists in having an automated framework for classifying different prodromal forms of AD. ML approach still suffers in terms of classification performance in classifying different prodromal AD phenotypes, such as MCI. The performance of ML model depends upon good definition and extraction of features from the brain images, which requires heavy labour and is susceptible to inter- and intra-rater variability. Along with the increase in neuroimaging data and the bottleneck of ML techniques, researchers are encouraged to perform diagnosis and prognosis of AD with deep learning (DL) algorithms.

2.2.2 Deep Learning

DL as a type of ML method has its working mechanism inspiration from the way human brain neurons process information. The most basic element of the DL networks are small nodes known as artificial neurons, which are usually arranged in layers wherein each neuron has connections to every neuron in the subsequent layer via weighted connections. Recently, the rise of DL technique has encouraged different areas of study to solve complex problem or enhance performances of existing study using the new technology. Example of application of DL includes machine translation, speech

recognition, sentiment analysis, image recognition, face recognition, signal processing, etc. (Amodei et al., 2016; He et al., 2016; Jagannath et al., 2020; Parkhi et al., 2015; Zhang et al., 2018).

The current trend of applying DL technique in medical applications has reaped great success with the potential of DL technology to perform faster analysis with higher accuracy when compared with human practitioners. To give an example, a notable study by Google on the diagnostic classification of diabetic retinopathy has shown remarkable performance that exceeds the capabilities of domain experts (Gulshan et al., 2016). Besides that, the improvement in parallel processing power of Graphics Processing Units (GPUs) allows development of more sophisticated DL model with large amount of parameters. This in turn enabled the development of cutting-edge DL algorithms and serves as motivation to apply DL techniques in neuroimaging area of study. Using neuroimaging data, DL models have shown great capacity in detecting invisible representations, discovering relationships between different sections of images, and identifying patterns that relate to a certain disease. Medical imaging modalities such as sMRI, fMRI, DTI, and PET can be successfully applied by DL techniques. Consequently, there are getting more researchers to extend the usage of DL technologies to solve diagnosis and prognosis problems in the realm of AD.

There is a spectrum of deep learning architectures used in the study of AD that can be separated into two main categories, which are supervised DL and unsupervised DL. Deep Neural Network, Deep Polynomial Network, Recurrent Neural Network, and 2D/3D Convolutional Neural Network are examples of supervised DL, whereas unsupervised DL comprises Autoencoder and Restricted Boltzmann Machine. In addition, the application of transfer learning techniques can be seen in several studies. As opposed to training a model from scratch, transfer learning method allowed the use of weights trained previously on a specific task to be reused as the starting point for a model on another task.

It comes with the benefits of shorter training time and is possible to deliver better results. CNN models such as AlexNet, VGG, ResNet, and LeNet are the most dominant models applied in transfer learning approach. Generally, there are two ways of applying transfer learning technique; First, pre-trained model with weights trained on ImageNet dataset as can be used as feature extractor; Second, fine-tuning of pre-trained model on a new problem. Study utilised transfer learning technology with AlexNet 2012 neural network to speed up the training of model for distinguishing CN from MCI subjects (Kumar et al., 2021). Even though transfer learning has the benefits of reducing training time and improving accuracy on small datasets, but the reproducibility of most studies is particularly low, and their modification on pre-trained models are not specified. Besides that, studies were noted to develop DL model from scratch with various novel techniques to pre-processed brain sMRI data (Basheera & Sai Ram, 2020; Y. Zhang et al., 2021).

Among the range of supervised methods, CNN is the most prevalent, especially in computer vision tasks. CNNs have attained promising classification results in the domain of medical imaging for organ segmentation and disease detection. It has also attracted the attention of researchers after several deep CNNs have achieved remarkable performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015). The design of CNN permits great utilisation of spatial information and extracting features by stacking multiple layers of convolutional layers, be it input of 2D or 3D images. Instead of training a classifier independently from extracted features, the main notion of CNN is to combine feature extraction and classification to avoid poor model learning performance affected by the heterogeneity of features and classifiers. In contrast to ML model, DL architecture permits unprocessed data as input and automatically identify and extract highly discriminative features for a problem in hand. If deep learning techniques are used, pre-processing steps are less essential (Liu et al., 2018). Nevertheless, most of the studies still incorporate pre-processing of neuroimaging

data involving different techniques such as skull stripping, registration, denoising, non-uniform intensity normalisation, segmentation, and motion correction. To date, CNN is the best choice of DL model for image-based application due to its capability to perform classification based on contexture information that outperforms approaches using pixel-based classification (Maggiori et al., 2017). In the study of AD, studies claimed that CNNs could automate early diagnosis of AD on an individual basis (Basaia et al., 2019; Bi et al., 2020).

In general, classification tasks are addressed in 3 main steps, notably feature extraction, feature dimensionality reduction, and classification. With the emergence of deep learning methods, all these steps can be merged into one. Feature extraction is crucial in establishing a quantified set of accurate information like shape, volume, and texture of different parts of the neuroimaging data. Basaia et al. (2019) addressed the problem of variation in MRI acquisition protocols by implementing CNN algorithm to classify AD and MCI without feature pre-processing. The model is reported to have lower generalisation error on new data after being tested on two different independent datasets of different magnetic resonance protocols.

However, owing to the extreme difficulty in handling the neuroimaging modality as a whole, the majority of the studies carried out their work by four different feature extraction techniques, notably region-of-interest (ROI) based, patch-based, voxel-based, and slice-based. ROI-based is one of the renowned methods to extract a specific part of the brain that is linked with the progression of AD. Identifying regions of interest of the brain involves the knowledge and expertise of neurologists to indicate which part of the brain will show deformities associated with AD. To give an instance, there is a study that extracted 83 functional features from MRI and PET scans (Liu et al., 2015). Another study extracted features using the measurement of GM tissue volumes and applied DL technique to choose the regional anomalies (Choi et al., 2018). The benefit of region-

based feature extraction is that it can represent the relative spatial information of the entire brain with lesser features. Although the number of regions of interest identified will decide the dimension of the ROI-based feature, it is far lesser compared to slice or voxel-based methods. According to prodromal stages study of AD, brain region such as hippocampus located in the temporal lobe can be used as features to diagnose AD due to its shrinkage in volume as the disease progresses (Eman et al., 2016). Moreover, in terms of efficiency, the highly sensitive ROI-based method and patch-based method to minor changes in the brain structure are better than the sliced-based method or voxel-based method.

CNN has an architecture of either 2D or 3D. 3D CNNs use 3D kernels to capture volumetric patch information from 3D neuroimaging data. It has better performance than 2D CNNs, but this comes with extra cost in terms of computing power and memory size. However, this has a downside of increased requirement for computing power and memory size. Study reported better accuracy in classifying AD-CN with multi-model 3D CNN (Li et al., 2017). However, the better performed 3D CNNs incurred higher complexity in training when compared to individual CNN. Therefore, 2D CNN is deemed to be easier to train. Nonetheless, study reported that 2D CNN that takes a single slice as input is lack kernel sharing across the third dimension (Liu et al., 2020). This is attributed to the nature of 2D CNN that it is unable to leverage context from adjacent slices, resulting in inefficient encoding of the voxel information of the 3D brain. Hence, some studies used RNNs following a 2D CNN to learn voxel information in adjacent slices (Ebrahimighahnavieh et al., 2020). Also, it is studied that the depth of CNN has an impact on classification performance, in which shallow or very deep network will not promise a good outcome (Wang et al., 2019).

Computer-aided systems developed for the detection of AD and MCI is a relatively new trend of study in the field. The main issue of the problem began with classifying AD

subjects from CN and is now evolving into a much more technically challenging problem — classification of subjects who will likely progress into AD (or progressive mild cognitive impairment (pMCI) subjects) from those who has MCI but less likely to convert into AD (or stable mild cognitive impairment (sMCI) subjects). On top of that, study of multiclass classification of AD-MCI-CN, which is the work of this project, is the least addressed problem along the AD continuum. The difficulties of multiclass classification as well as study of different phenotypes of AD are being reflected on the performance of previous studies as summarised in Table 2.1.

Table 2.1: Literature summary of recent studies that are similar to this project

Study	Database	Classification Algorithm	Validation	Classification	Performance		
					Accuracy	Sensitivity	Specificity
(Abrol et al., 2020)	ADNI	CNN (ResNet)	5-fold stratified cross validation	AD/CN	91.00%	-	-
(Asl et al., 2018)	ADNI	CNN	10-fold cross validation	AD/MCI/CN	94.80%	-	-
(Basaia et al., 2019)	ADNI	CNN	Independent sample	AD/CN pMCI/sMCI	99.00% 75.00%	98.90% 74.80%	99.5% 75.30%
(Basheera & Sai Ram, 2020)	ADNI	CNN	10-fold cross validation	AD/CN AD/MCI CN/MCI AD/MCI/CN	97.00% 97.00% 68.00% 86.70%	97.00% 97.00% 68.00% 74.00%	97.00% 97.00% 70.00% 74.00%
(Bi et al., 2020)	ADNI	CNN	Independent sample	AD/MCI MCI/CN	95.52% 90.56%	- -	- -
(Gupta et al., 2019)	NRCD	SVM	5-fold cross validation	AD/CN pMCI/sMCI	93.06% 86.95%	87.87% 77.77%	95.58% 92.85%
(Jain et al., 2019)	ADNI	CNN (VGG-16)	Independent sample	AD/MCI/CN	95.73%	-	-
(Liu et al., 2020)	ADNI	CNN (multi-model)	5-fold cross validation	AD/CN MCI/CN	88.90% 76.20%	86.60% 79.50%	90.80% 69.80%
(Oh et al., 2019)	ADNI	CNN	5-fold cross validation	AD/CN pMCI/CN	86.60% 77.37%	88.55% 81.03%	84.54% 74.07%
(Payan & Montana, 2015)	ADNI	CNN	Independent sample	AD/MCI/CN	89.47%	-	-
(Rallabandi et al., 2020)	ADNI	SVM	10-fold cross validation	AD/MCI/CN	75.00%	75.00%	77.00%

2.3 Convolutional Neural Network

Convolutional neural network (CNN), also known as ConvNet is a type of deep neural network with extensive usage, particularly in image-based applications. CNNs are the most representative supervised learning model. Hierarchical framework together with multiple levels are characteristics of CNNs. It excels in performing classification of contextual data. Using 2D or 3D images as inputs, CNN has great capability in utilizing spatial information and extracting features by stacking multiple convolutional layers. CNN incorporates feature extraction to identify different features of the data or image for analysis. Fully connected layers at the end of a CNN accept convolution output from previous layers to perform predictions based on features retrieved from previous layers. Instead of training a classifier independently from extracted features, the main concept of CNN is to combine feature extraction and classification to avoid poor learning process caused by the heterogeneity of features and classifiers.

2.3.1 Architecture

All CNN models are based on the same general architecture as described in Figure 2.5: CNN general architecture adopted from (Hidaka & Kurita, 2017)Figure 2.5. CNN architecture comprises four main components as follows: (1) convolutional layer, (2) pooling layer, (3) activation function, and (4) fully connected (FC) layer. The functionality of these components will be illustrated in the subsequent sections. Stacking all these components together forms a CNN architecture. The number of layers of each type of layer varies is dependent on the problem to be solved. On top of that, dropout layer is also one of the important parameters in the CNN.

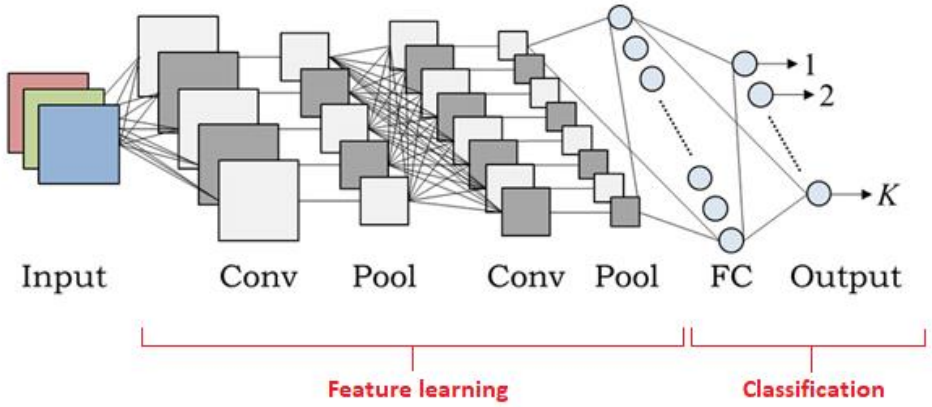

Figure 2.5: CNN general architecture adopted from (Hidaka & Kurita, 2017)

2.3.1.1 Convolutional Layer

Input image represents by an array of pixel values is fed into the model to perform a series of convolution and pooling mathematical operations for feature learning followed by classification carried out in the FC layer. Convolution mathematical operation is performed in the convolutional layer, in which two functions are combined to form a new function. It involves sliding a filter or kernel, also known as the weight vector, horizontally and vertically over the input vector and a feature map is generated with information about the image such as edges, lines, and corners. The dimension of the filter is also called the window size or kernel size of a convolution. While the filter of a size of M×M is sliding over the input vector, the dot product between the filter and the input vector is calculated with respect to the filter size. Based on Figure 2.6, a dot product is calculated between a 3×3 matrix filter and a 3×3 sized area of the input vector. Multiple filters may be used for one input vector and the feature maps generated are combined as one output of one convolutional layer.

In neural networks, each neuron has connections to several neurons in the previous layer. The receptive field is the local correlation that maps a single neuron in a layer to some neurons in the following layer (Indolia et al., 2018). The local features of an input image are extracted using receptive fields. As shown in Figure 2.6, the area of the

receptive field is 3x3 as determined by the filter. Whereas, for FC layer, the receptive field is the whole previous layer.

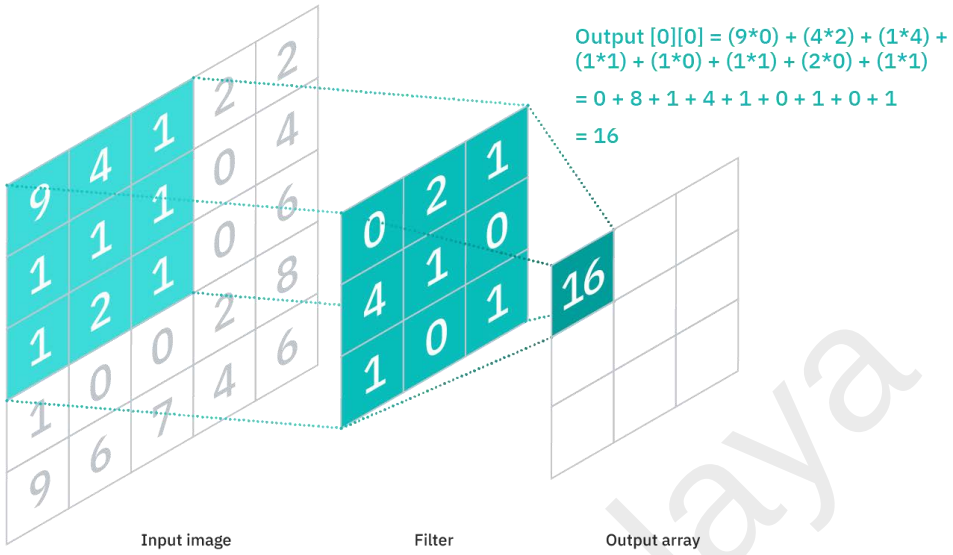

Figure 2.6: Convolution operation with stride 1

In addition, there are three other important parameters that affect the spatial of the output feature map, which are (1) stride, (2) padding, and (3) depth. Stride refers to the number of steps or pixels the filter slides over the image vector for each time. For instance, a stride of 2 implies the movement of filter by two pixels at a time. If the number of strides is increased to 3, which is rare in real-world practice, the filter will move 3 pixels at a time, thereby reducing the feature map output spatially. Padding is usually used when the filter does not fit the input matrix. For instance, the filter goes over the limit of the image boundary while sliding across the input matrix. One of the most common paddings is the zero padding also known as “same” padding. Zero padding can be used to manipulate the spatial size of feature map output. Depth implies the number of filters available for a convolutional layer. It is common for a filter to have depth as the input. For coloured images, there are multiple channels of red, green, and blue. Hence, the depth = 3 as the input has 3 channels.

2.3.1.2 Pooling Layer

Generally, pooling refers to gathering something in a small portion. Pooling or sub-sampling layers help to decrease the computational time for feature extraction by reducing the dimensionality of feature maps in height and width while preserving the depth. Two main types of pooling layer such as max pooling and average pooling are commonly used. Between them, max pooling is the most commonly used method for CNNs.

In max-pooling operation, the maximum value of an image region of size $h \times w$ is extracted, specified by stride, s and kernel size, k, which yields the formula of $\frac{(h-k)}{s+1} \times \frac{(w-k)}{s-1}$. It is extensively used than other pooling methods due to the fact that max-pooling significantly reduces map size and preserves the maximum value information of a pixel instead of the average value of pixels in the window (Lee et al., 2017). By inserting a max-pooling layer in between the successive convolutional layers can progressively decrease the size of spatial representation while preserving the spatial information. Computational cost of the CNN is reduced as the number of trainable parameters are lesser while reducing the possibility of a model to overfit.

Values of k and s of 2 are commonly seen, which allows down sampling of h and w by a factor of 2. Figure 2.7 illustrates max-pooling operation with a filter of size (2, 2) and stride of 2. Extraction of the largest value from the window happens when the filter of size of 2×2 slides over the feature map. Max pooling down samples the feature map by half from a 4×4 matrix to a 2×2 matrix.

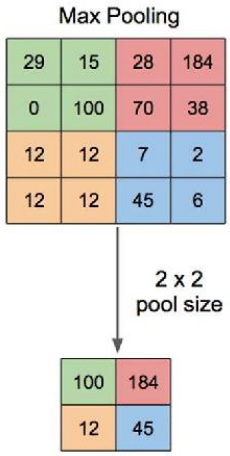

Figure 2.7: Illustration of max pooling

2.3.1.3 Activation Function

Activation functions are responsible for learning the relationship between variables of neural networks. They ensure non-linearity in neural networks and decide which neurons will be fired in the forward direction in a particular layer. Generally, activation functions are placed in subsequent to convolutional layers. In ML algorithms, sigmoid and tanh activations were widely used in a significant amount of literature. However, they have certain limitations which encouraged researchers to introduce another activation function which is Rectified Linear Unit (ReLU) (Nair & Hinton, 2010). Among various activation functions, ReLU has shown to be better than the former due to the easy calculation of partial derivatives (Indolia et al., 2018). ReLU function, as shown in Equation (2.1), returns 0 for input with a negative number else it returns the identical input value. Other examples of activation functions include softmax.

$$f(x) = \max(0, x) \tag{2.1}$$

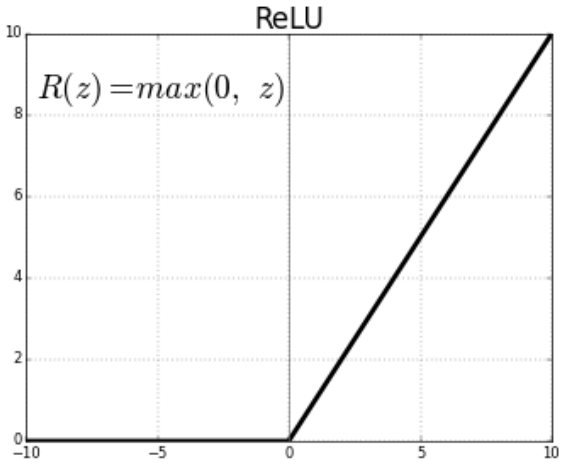

Figure 2.8: Graph of ReLU activation function

Another activation function is the softmax which is commonly used for the output of multiclass classification problems. Similar to sigmoid function, it also outputs a vector of decimal values between 0 and 1 that adds up to 1. In general, the softmax activation function for more than 2 number of classes can be described in the formula as follows:

$$f(\vec{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}} \tag{2.2}$$

where $\vec{x}$ represents the input vector and K signifies the number of classes of the multiclass classifier.

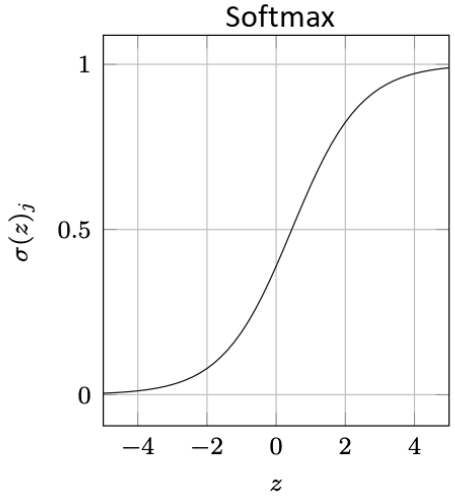

Figure 2.9: Graph of softmax activation function

2.3.1.4 Fully Connected Layer

Fully connected (FC) layer is comparable to the conventional FC neural network. The FC layer carries biases and weights along with the neurons. In general, FC layers are placed prior to the output layer and are sometimes known as the classifier of a CNN architecture. Generally, there will be at least one FC layer. The layers before the FC layer, including repetitive convolutional and pooling layers are flattened into a one-dimensional (1D) vector and fed to the FC layer. In this stage, mathematical operations take place for the classification process. In specific, the dot product of the input vector and weight vector is calculated to obtain the final output.

2.3.1.5 Dropout

Dropout is a technique to randomly drop out both hidden and visible neurons for the purpose of introducing irregularities to prevent overfitting of CNN (Srivastava et al., 2014). Usually, overfitting happens when a specific model is deemed to have memorised the dataset and unable to generalise to unseen data, which negatively impacts model performance. In dropout layer, during training, a number of neurons are dropped from the neural network, thereby reducing the size of the model. Dropout rate or ratio is a parameter that can be tuned in a neural network, where 0.0 indicates no output from the layer and 1.0 means all nodes in a layer are completely dropped out. Another example is that a dropout of 0.2 represents 20% of the nodes are chosen at random and dropped from the neural network. The forward pass and backward pass processes will not include nodes that are dropped out.

2.3.1.6 Batch Normalisation

Batch normalisation is yet another regularisation technique to deal with overfitting often seen in small datasets. It was proposed by Sergey Ioffe and Christian Szegedy in 2015 (Ioffe & Szegedy, 2015). Normalisation is generally done during data pre-processing to ensure the numerical data are in common scale before feeding the data to

the DL model. Now coming back to batch normalisation that takes place in batches, it helps to normalise and standardise the input of layers coming from a previous layer using the mean and variance of the current batch and this process is repeated for every mini-batch. It benefits deep neural networks to train faster and more stable in addition to improving final generalisation error.

Regarding the position to place the batch normalisation layer is still a debatable topic by the researchers and this question is still being discussed up till today. Ioffe and Szegedy (2015) reported that placing batch normalisation layer right before the non-linear function can ensure consistent improvement in model performance as the network will always produce activations with the desired distribution. However, the founder of Keras, Chollet, implied that applying batch normalisation after non-linear function such as the ReLU activation led to better results (Chollet, 2016). Furthermore, it is suggested that dropout could be omitted when batch normalisation is used due to the fact that the statistics used to normalise the activations of the layer before batch normalisation may be noisy caused by the random dropping out of nodes during the dropout procedure (Dario et al., 2016; Ioffe & Szegedy, 2015). This does not indicate that dropout is not viable to be used alongside batch normalisation. Batch normalisation layer should be placed right before dropout layer to avoid the problem of noise.

CHAPTER 3: METHODOLOGY

3.1 Dataset - ADNI

The relevant data were retrieved from the database of Alzheimer's Disease Neuroimaging Initiative (ADNI), which is available publicly upon approval from the ADNI. In 2004, Dr Michael W. Weiner led the ADNI that was launched as a public-private partnership. The ADNI was established to study the progression of AD and diagnosis of its early stages using gathered study data in serial MRI, PET, and other biochemical biomarkers. More information about the ADNI is available on the ADNI official homepage (http://adni.loni.usc.edu).

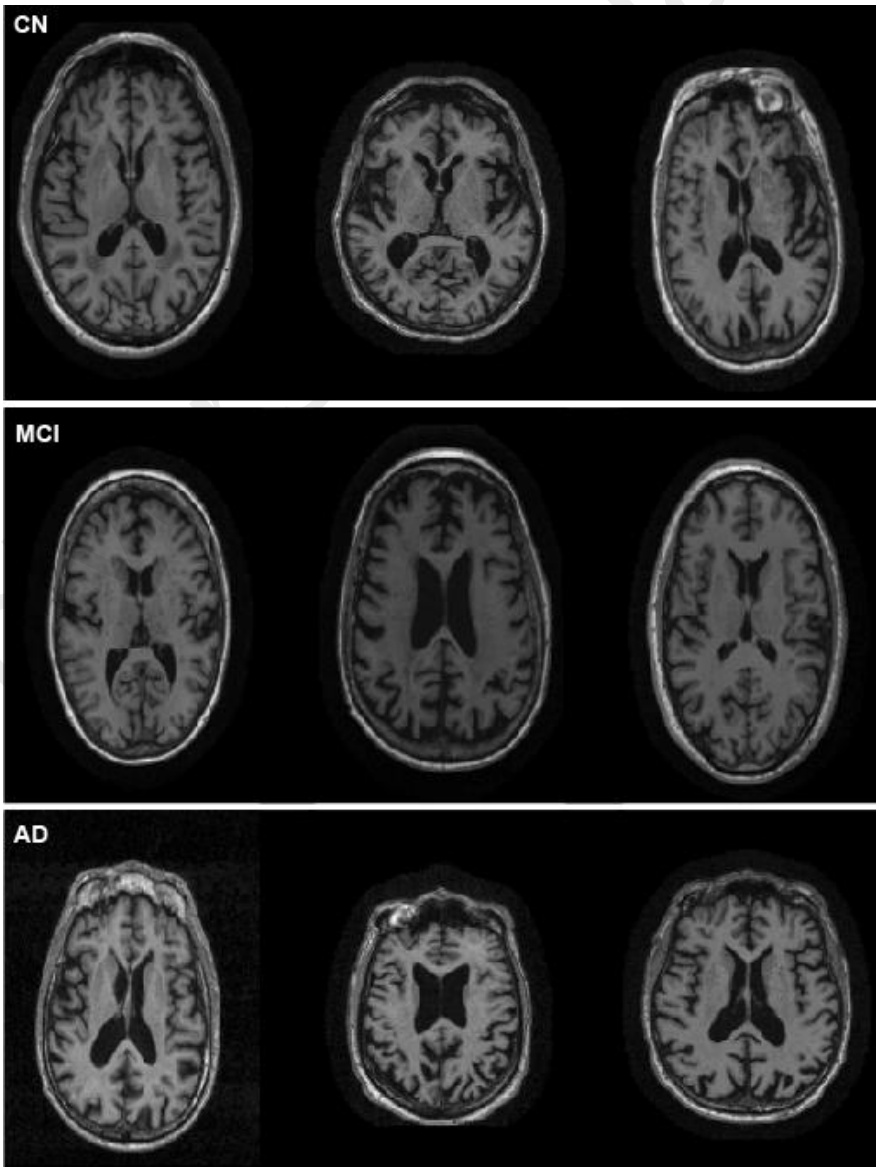

Figure 3.1: Sample of brain sMRI from the ADNI dataset

Based on Figure 3.1, sample brain images from 9 different participants of three different classes (CN, MCI, and AD) are shown. The top row is cognitively normal or healthy control subjects; The middle row is mild cognitive impairment patients; The bottom row is Alzheimer's disease patients.

The ADNI database contains multiple collections of MRI images categorised by phase of the study, for example, ADNI1, ADNI2, ADNI-GO, and ADNI3 (as of August 2021). This project worked with all the sMRI data in the ADNI1 collection. The CN class consisted of healthy aging controls with no conversion within 3 years of follow-up visits from baseline. Subjects diagnosed with mild cognitive problems, but their symptoms do not interfere with their ability to carry out daily activities were retained in the MCI class. The AD class comprised patients identified as AD through diagnosis at baseline and showed no sign of reversion within 2 years of follow-up visits.

All the acquired sMRI were generated from scanners of various manufacturers such as Philips, Siemens, and General Electric. Due to the various acquisition protocols, it was necessary for the dataset to be subjected to pre-processing. All MRI scans had 1.2mm spacing in between and a voxel dimension of 256×256×256. In terms of resolution, there was only a slight difference found across the patients. On each visit, patients have to undergo a series of tests—for example, cognitive tests, PET scans, MRI scans, and other neurological assessments.

The data used were restricted to the standard 1.5T T1-weighted sMRI which were acquired by the volumetric three-dimensional magnetisation-prepared rapid gradient-echo (3DMPRAGE) protocol. Other data acquisition settings include 8-channel coil, TR = 650 ms, TE = min full, flip-angle = 8°, and FOV = 26 cm. Participants may have multiple scans at baseline and follow-up visits (after 1 year, 2 years, and 3 years). It is important to note that not all participants appeared at every planned follow-up visit. Some participants retained in the study without appearing at every follow-up meeting. It was

also noticeable that a substantial drop of follow-up visitation rate after 2 years indicating that fewer data were available over time. Table 3.1 and Figure 3.2 summarise the demographic information for the 819 subjects with age ranges from 55 to 92 years old, including 192 patients with AD, 398 subjects belonging to the MCI, and 192 who are cognitively normal that were included in the study. Based on Table 3.1, it can be seen that the CN group are more educated than the MCI and AD groups with mean education years of 16.0 ± 2.9 years and the MCI group is the youngest among the 3 groups with mean age of 74.7 ± 7.4 years old.

Table 3.1: Demographic of participants with MCI and AD and cognitive normal subjects from the study population

Diagnostic type	Number of participants	Age, Mean ± S.D. (years)	Gender (M/F)	Education, Mean ± S.D. (years)
CN	229	75.8 ± 5.0 (59.9-89.6)	119/110	16.0 ± 2.9 (6-20)
MCI	398	74.7 ± 7.4 (54.4-89.3)	257/141	15.7 ± 3.0 (4-20)
AD	192	75.3 ± 7.5 (55.1-90.9)	101/91	14.7 ± 3.1 (4-20)

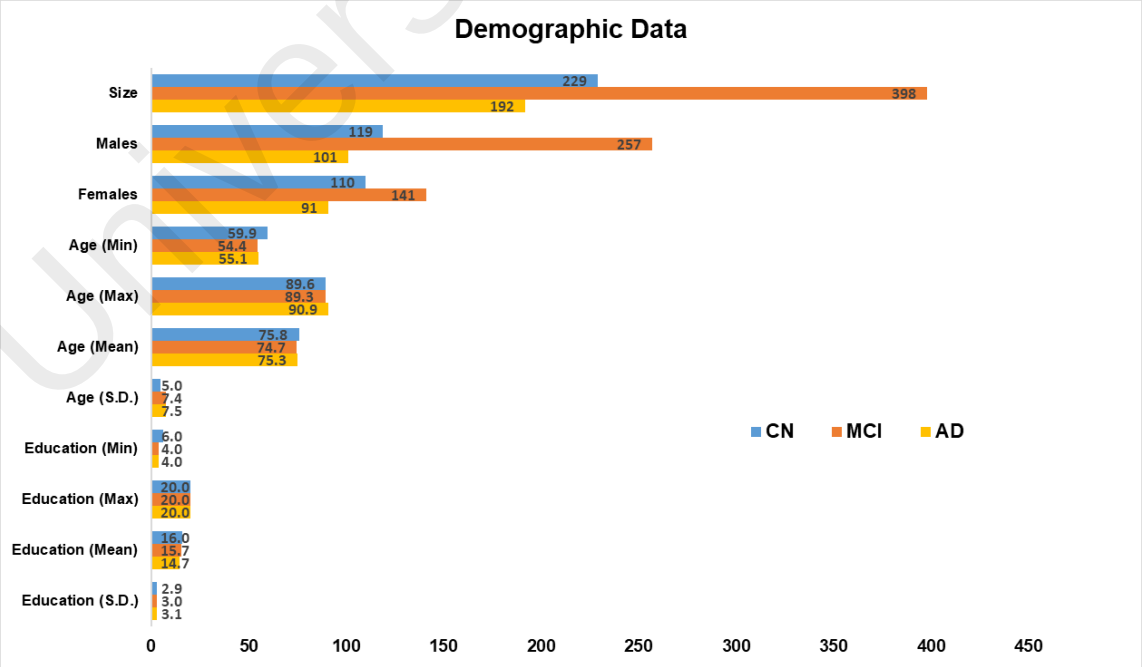

Figure 3.2: A comparison of demographic data for different studied classes

3.2 Project Workflow

The workflow of this project is illustrated in the schema shown in Figure 3.3. It is similar to most pipelines of deep learning model studies. Going along the workflow, first, the retrieved ADNI1 dataset was subjected to a series of pre-processing procedures. Then, the extracted 2D images were split into training, validation, and testing set. Three different CNN models were being experimented, including a CNN trained from scratch, VGG-16, and ResNet-50. The training data was augmented prior to feeding to CNN models for training, while the testing data was used to evaluate the performance of the models with previously unseen data.

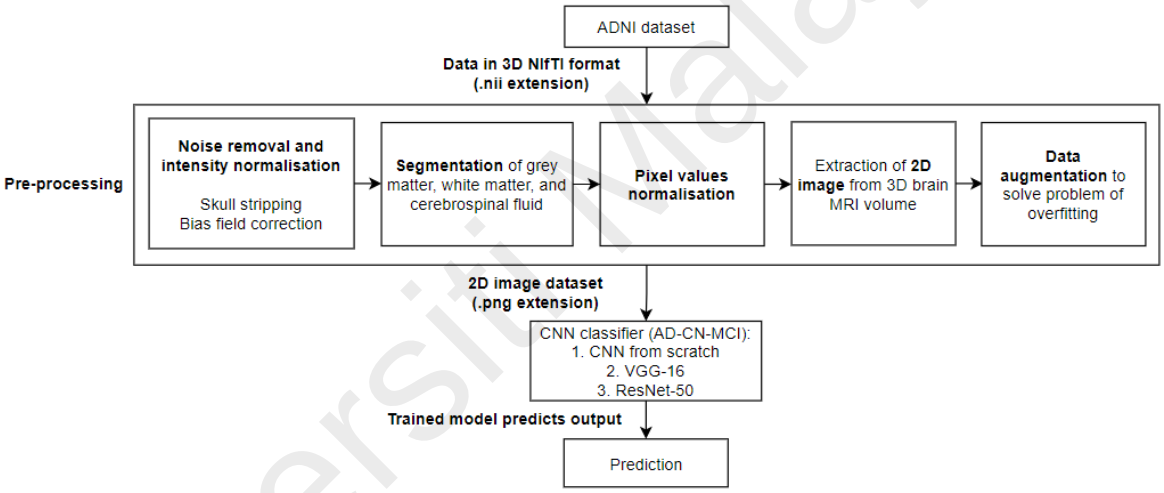

Figure 3.3: Methodology flowchart

3.2.1 Pre-processing

Initially, the dataset was in the Neuroimaging Informatics Technology Initiative (NIfTI) format (.nii extension) with the capability to store information such as the dimension, image array, affine data, etc. NIfTI files are 3D or volumetric images of the brain, where every file has a voxel dimension of 256×256×256. Thus, there were 256 slices or 2D images corresponding to each NIfTI file.

To reduce computational time, 2D images were used as input for the classification task as 3D data is generally huge in data size. In addition, owing to the variation of MRI acquisition protocols, there was heterogeneity in the dataset. Thus, pre-processing was

applied to each brain sMRI to normalise the data into desired form and format. The routine of pre-processing steps can be summarised into 5 different steps: (1) skull-stripping; (2) non-uniform intensity correction; (3) segmentation; (4) extraction of 2D image from 3D MRI volume; (5) pixel values normalisation; and (5) data augmentation. Due to the limitation of the operating system which requires MacOS or Linux, instead of using the popular brain MRI image processing toolboxes such as FreeSurfer by the Harvard University (https://surfer.nmr.mgh.harvard.edu/) and FSL software by the Oxford University (https://fsl.fmrib.ox.ac.uk/), this project opted to perform pre-processing using Python (Rallabandi et al., 2020; J. Zhang et al., 2021).

3.2.1.1 Skull Stripping

Skull stripping is the process of removing the skull from the 3D brain MRI. It is considered the most important preliminary processing step that must be performed prior to other pre-processing steps (Goceri & Songül, 2017). For quantitative morphometric study, the skull is the non-brain tissue that acts as noise which would deteriorate the classification performance of CNN. Besides that, skull stripped brain ensures to get better segmentation results. The skull portion was stripped or removed using the deepbrain library, leaving the brain tissues. The proposed skull stripping algorithm involves a series of steps as presented in Figure 3.4. As shown in Figure 3.5, the raw brain had its skull stripped together with intensity normalised using the deepbrain library.

Step 1: Loads the 3D MRI file which is then later converted to NumPy array
Step 2: Create variable to store affine data of the input file
Step 3: Create instance of class Extractor
Step 4: Calculate and save set of probabilities
Step 5: Identify the region of brain with threshold of 0.5: *mask = prob > 0.5*
Step 6: Make unwanted pixel to 0: *brain[~mask] = 0*
Step 7: Convert pixels back to NIfTI image: *brain = nib.Nifti1Image(brain, affine)*
Step 8: Save the images as 3D using NiBabel library

Figure 3.4: Skull stripping algorithm

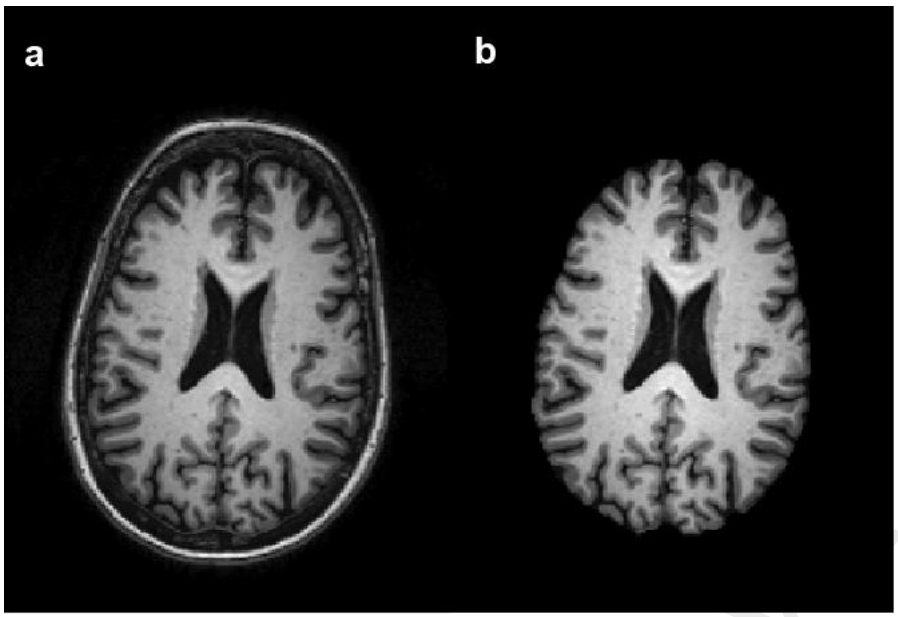

Figure 3.5: Comparison of (a) raw brain MRI and (b) skull stripped brain MRI

3.2.1.2 Bias Field Correction

Strong bias fields are known to cause mislabelling of voxel tissue type. This could compromise algorithm accuracy that is heavily dependent on grey and white matter contrast (Gupta et al., 2019). To control this effect at minimal level, the N4 bias field correction method was performed using the SimpleITK library for correcting low-frequency intensity presented non-uniformly in the brain sMRI (Tustison et al., 2010). Equation (3.1) describes the formation of image model, where v is the given image, u is the uncorrupted images, f is the bias field, and n is the noise.

$$v(x) = u(x)f(x) + n(x) \tag{3.1}$$

Utilising formula $\hat{u} = \log u$ and assuming a noise free scenario, image model in Equation (3.1) becomes,

$$\hat{v}(x) = \hat{u}(x) + \hat{f}(x) \tag{3.2}$$

$$\begin{aligned} \hat{u}^n &= \hat{v} - \hat{f}_e^n \\ &= \hat{v} - S\{\hat{v} - E[\hat{u}|\hat{u}^{n-1}]\} \end{aligned} \tag{3.3}$$

in which Equation (3.3) can be used for calculating the corrected image at specified iteration. $\hat{u}^0 = \hat{v}$, $\hat{f}_e^0$ is typically set to 0 for initial estimation of bias field and the smoothing operator, $S\{.\}$, is a *B*-spline approximator.

The intensity variation of the same brain tissue was eliminated based on the location of the tissue within the image. The bias corrected brain has shown more uniform intensity at the white matter region (Figure 3.7).

Step 1: Load the 3D MRI files as images using the SimpleITK ReadImage class.
Step 2: Type cast the pixels of image as float32 using the SimpleITK sitkFloat32 function.
Step 3: Identify the regions of non-uniform intensity using the SimpleITK BinaryThreshold class.
Step 4: Run the N4 bias field correction algorithm using the nipype N4BiasFieldCorrection class.
Step 5: Save the images as 3D using the SimpleITK WriteImage inbuilt function.

Figure 3.6: N4 bias field correction algorithm

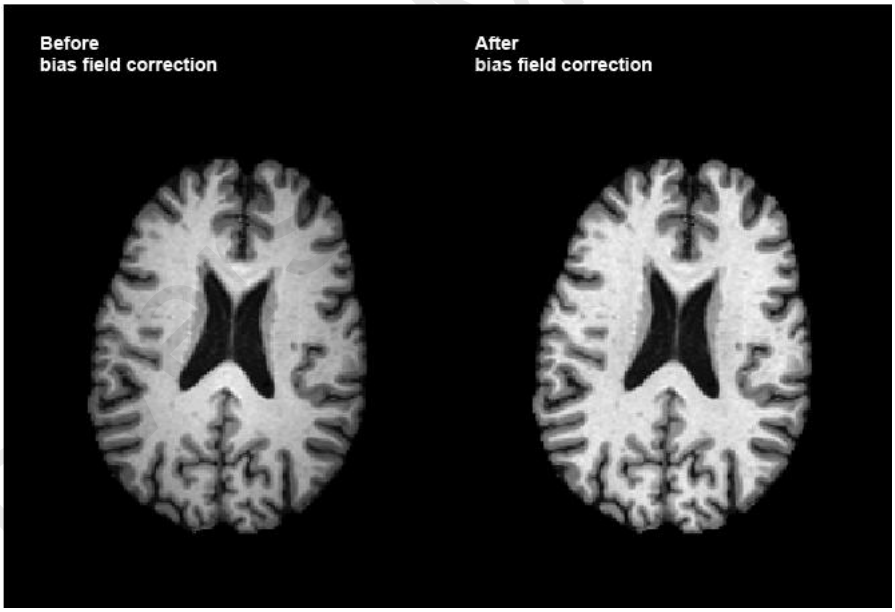

Figure 3.7: Comparison of before and after bias field correction

3.2.1.3 Tissue Segmentation

The T1-weighted sMRI data that had been previously skull-stripped and bias field corrected were segmented using the hidden Markov random field (HMRF) tissue classifier (Zhang et al., 2001). The concept of HMRF was derived from the hidden

Markov models. HMRF has an underlying Markov random field instead of an underlying Markov chain, as seen in hidden Markov.

The brain sMRI volumes were segmented into 3 different regions of GM, WM, and CSF using the HMRF tissue classifier from the dipy libary. These were the 3 main features used to differentiate AD from MCI and CN. Alterations in WM and GM were commonly used for the analysis of AD progression (Klöppel et al., 2008). In ML approach studies, it would be laborious to perform tissue segmentation and feature extraction. Hence, automated segmentation is essential for a dataset with a large number of images.

Step 1: Load the 3D MRI files as image data arrays which had been skull stripped and bias field corrected previously.
Step 2: Define the values of *nclass = 3* since 3 regions will be segmented and beta = 0.1 indicates the smoothness factor of the segmentation.
Step 3: Create an instance of TissueClassifierHMRF class from the dipy library.
Step 4: Call the *classify* method and input the parameters defined in Step 2 to start the segmentation.
Step 5: Convert the pixels back to NIfTI image using the NiftiImage class from the NiBabel library.
Step 6: Save the images as 3D and save them to desired location.

Figure 3.8: Tissue segmentation algorithm

Figure 3.9 shows the plotting of the resulting segmentation with a clear separation between GM, WM, and CSF. Image (a) is the brain image before segmentation. For comparison, image (b) shows each segmented tissue class that is clearly differentiated with different grayscale intensity. For better visualisation, image (c) depicts a brain image with each tissue class with colour coded separately in which WM in yellow, GM in green, and light blue is the CSF.

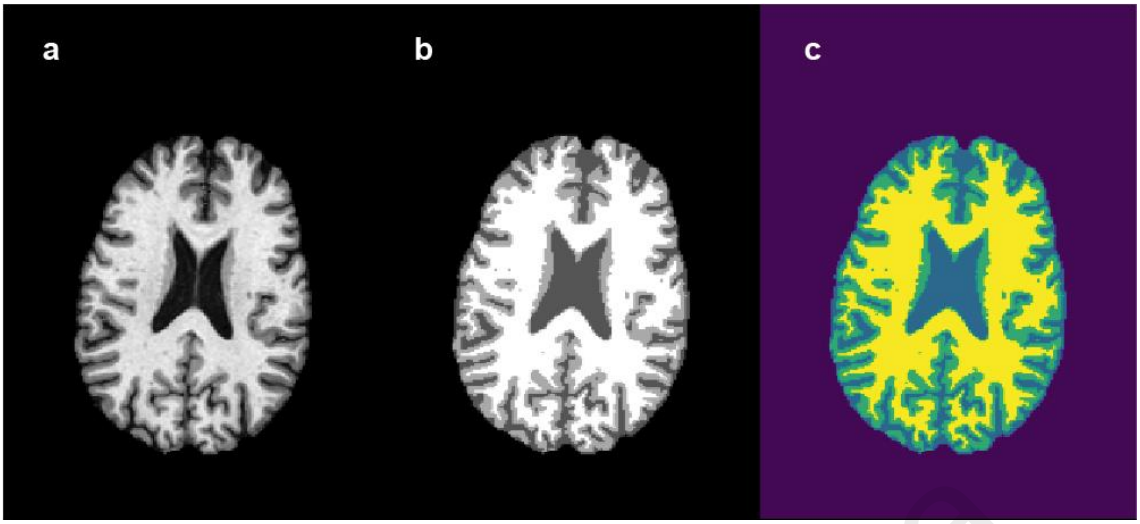

Figure 3.9: Segmentation results using the HMRF tissue classifier

3.2.1.4 Extraction of 2D image

After the segmentation process, matplotlib library was used to extract 2D slices or images from the segmented 3D MRI. More specifically, brain images of the axial view from the slices in the range of 160th slice to 170th slice of the 3D MRI were extracted in PNG format. Slices in this range provided the most information regarding the GM, WM, and CSF. There was a total of 2387 brain scans for the three classes (CN, MCI, and AD). Selecting the best possible slices with relevant morphological information is correlated with good model performance (Fung & Stoeckel, 2007). Given the preferable slice range, every interval of 5 slices (e.g., 160th, 165th, and 170th), 3 brain images were extracted from the MRI volume of the AD and CN subjects in which AD and MCI have 2043 and 2051 images, respectively. One scan was removed from the CN class due to file corruption. In addition, 2 brain images (160th and 165th) were extracted for the MCI class that yielded 2044 images.

A padding private function was implemented to add padding to all final images, so that the output images have a uniform dimension of 271×271 pixels. Here, the images were saved in grayscale format and named according to their classes with a number suffix in an increasing sequence. After pre-processing, the data were all in the form of 2D images. This helped to substantially reduce the dataset size from 37GB to 260MB.

Table 3.2: Number of patients, scans, and images for different diagnostic types

Diagnostic type	Number of patients	Number of scans	Number of images
CN	229	684	2051
MCI	398	1022	2044
AD	192	681	2043

3.2.1.5 Pixel Values Normalisation

As of this stage, every image data was in grayscale and had pixel values which are integer values in the range of 0 to 255 (8-bits). Before allowing image data to be used for model training or evaluation, it was a good practice to normalise every image pixel value with value between 0 and 1. All the pixel values were divided by the greatest pixel value, which is 255. In this case, the normalisation procedure was only performed across one channel as all images were in grayscale. Pixel values normalisation has the advantages of ensuring good computation efficiency as the CNNs use input with small weight values instead of larger floating values that can slow down or disrupt the learning process.

Conventionally, the image data could be normalised prior to model development and stored on a disk in the scaled format. In this project, an alternative approach was implemented. The Keras library provided ImageDataGenerator class to scale or normalise pixel values alongside data augmentation just in time before feeding the image data to the CNNs. Image normalisation would improve image contrast at the same time removing high-frequency and low-frequency noises. As shown in Figure 3.10, the discrepancy between before and after pixel values normalisation is not apparent due to the fact that the segmented image has great contrast, which lessened the visual effect of contrast enhancement. However, the plots of pixels distribution provided information about the pixel values of a tensor image. Based on Figure 3.11, the pixel values are between 0 and 255 and it was observed that the pixels distribution had most of the pixels of values 0 and 255, wherein 0 indicates the completely black pixels while 255 being the completely

white pixel. After normalisation, the distribution of the pixels remained the same but differed in pixel values range, which is in the range of 0 to 1 as shown in Figure 3.12.

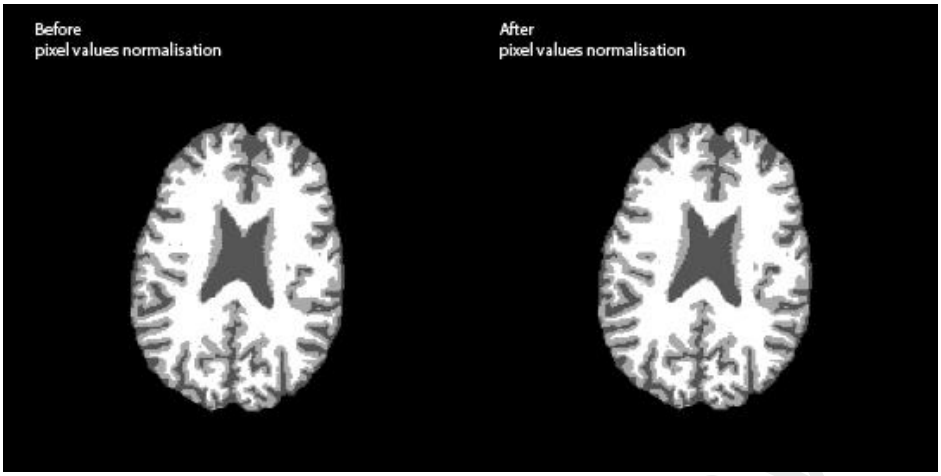

Figure 3.10: Comparison of before and after pixel values normalisation

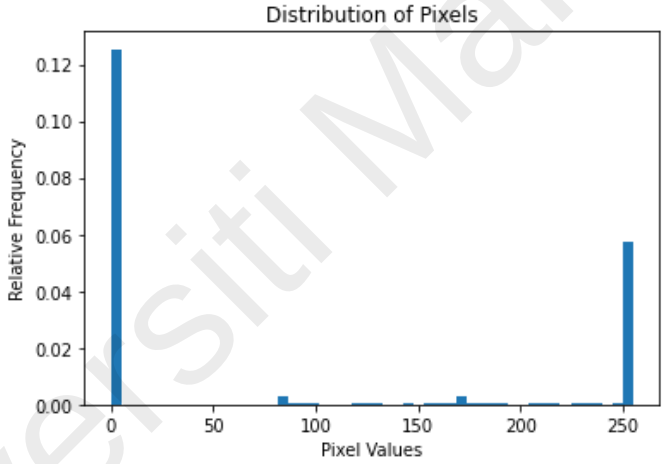

Figure 3.11: Plot of distribution of pixels before applying pixel values normalisation

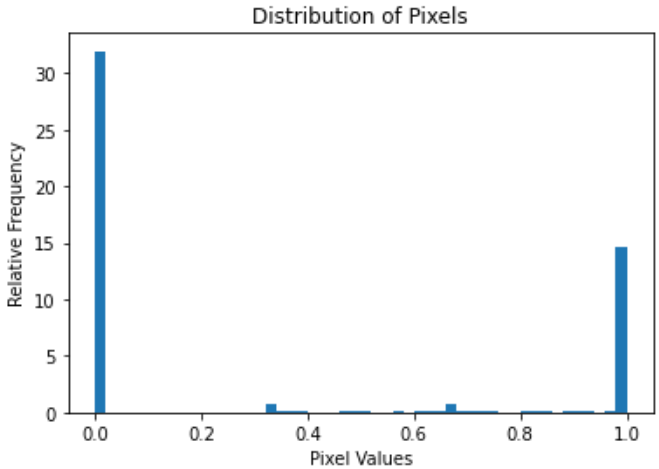

Figure 3.12: Plot of distribution of pixels after applying pixel values normalisation

3.2.1.6 Data augmentation

The process of data augmentation was performed to mitigate the general problem of the small dataset, which is overfitting during training, by applying various transformations on the images from the dataset. Small data will encourage the model to memorise the details of the training set but perform poorly on the validation set. Data augmentation is a method to increase the diversity of data by randomly applying the specified transformation to the dataset, which enhances the ability of the model to generalise.

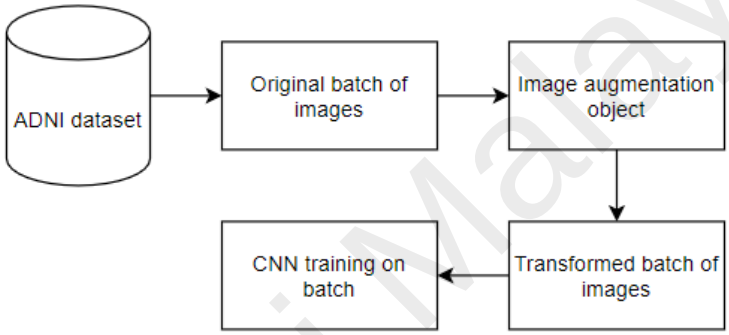

Figure 3.13: Data augmentation flowchart with Keras API

Data augmentation was applied to the training and validation sets, excluding the test set. The type of data augmentation used is known as in-place data augmentation or on-the-fly data augmentation. The augmentation was done during the training process instead of the generation of images prior to training. The Keras library provided a useful class called ImageDataGenerator, as previously used for pixel values normalisation, for this form of data augmentation. During training, an input batch of images was directed to the ImageDataGenerator. The ImageDataGenerator transformed a batch of images with a range of transformations as shown in Table 3.3, randomly. Then, the transformed or augmented images were returned to the calling function. The transformations used were rotation of 15 degrees, zoom range of 0.10 degree, height shift range of 0.10 degree, and width shift range of 0.10 degree. An example of a collection of augmented brain MRI images is shown in Figure 3.14.

Table 3.3: Data augmentation

Type of augmentation	Value
Rotation range	15
Zoom range	0.10
Height shift range	0.10
Width shift range	0.10
Shear range	0.10
Horizontal flip	True

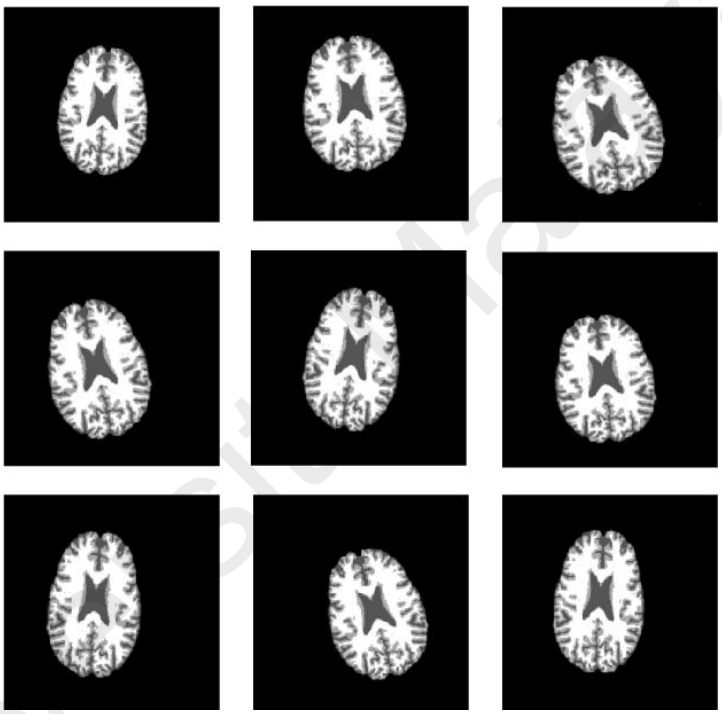

Figure 3.14: Collection of augmented sMRI images

3.3 Dataset Splitting

To guarantee unbiased model classification performance, the dataset was ensured to have balanced classes. The dataset was divided into 2 different sets which are (1) training and validation set and (2) testing set with a split ratio of 80:20. The resulting sizes of training and test set for multiclass classification (AD-CN-MCI) are summarised in Table 3.4. The dataset consisted of 6138 images; There were 3927 training images, 984 validation images, and 1227 testing images.

Figure 3.15 illustrates the data splitting process, where 80% of the pre-processed sMRI data were used for training and validation while the remaining 20% of the data for testing the CNN models. 80% of the separated data were then randomly separated into two subsets, 80% for training data and 20% for validation data. Data augmentation was applied to the training set, excluding the validation and test sets. The CNN models were trained on the augmented brain images and validated using the validation data without data augmentation alongside hyperparameters tuning. After model training, the CNN models were tested for their classification performance with the testing set that was not previously seen by the models. The classifier performance was gauged with performance metrics such as accuracy, precision, recall, and F1-score, which were calculated independently using the testing data.

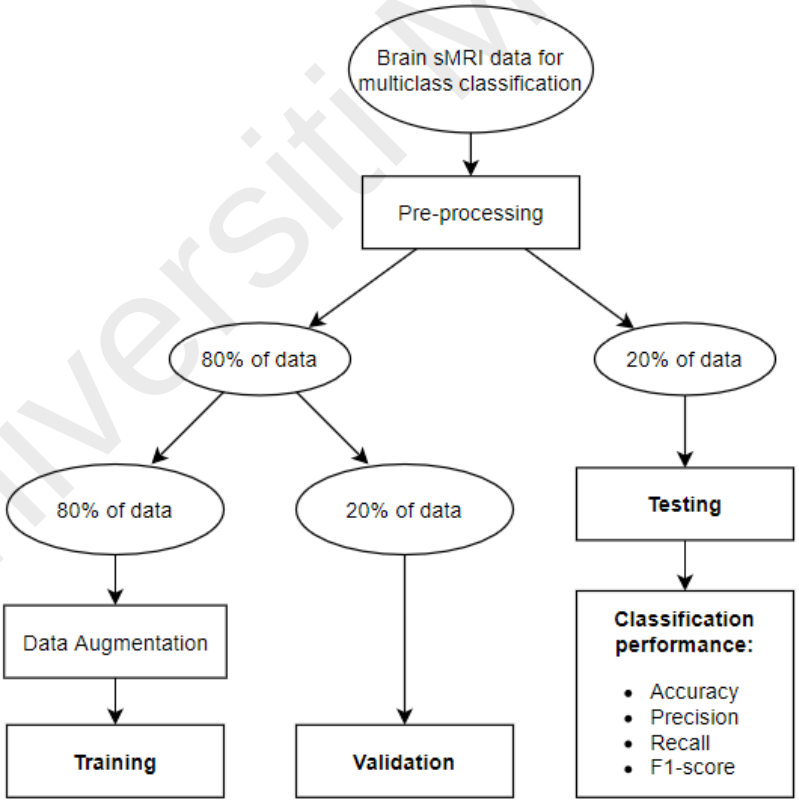

Figure 3.15: Flowchart of data splitting

Table 3.4: Sizes of training, validation, and testing set

Class label	Training set	Validation set	Testing set	Total images
CN	1308	328	408	2044
MCI	1307	327	409	2043
AD	1312	329	410	2051

3.4 CNN Architectures

Here, details of the CNN models implemented in this project will be introduced. Three different CNN models were being experimented with to perform the 3-way classification task. The first model is a CNN trained from scratch. On top of that, the second and third models employed transfer learning technique. Instead of training a model from scratch, CNN models with pre-trained ImageNet weights such as VGG-16 and ResNet-50 were employed. These models were trained to classify 1000 different classes of images using the ImageNet database consisting of more than a million images.

3.4.1 CNN From Scratch

The 2D CNN architecture which was trained from scratch is represented in Figure 3.16. Briefly, the architecture comprised of the following: 5 convolutional layers followed by ReLU activation; 5 layers of max-pooling layers; 2 dropout layers; a flatten layer; a fully connected layer with 256 neurons followed by a dropout layer and a batch normalisation layer; and ultimately an output layer with softmax activation which provided the probability of prediction for each class in the range of 0 to 1. The class with the largest probability was indicated as the predicted class.

The pre-processed axial view brain sMRI data were fed into the CNN first layer. The second layer was a convolutional layer to perform convolution operation of input images and filter with resultant of multiple feature maps. There was a total of 5 convolution layers with 16-32-64-128-256 feature maps sequentially. All the convolution filters had sizes of 2×2, stride of 1, and “same” padding that ensured the output has the same dimension as

the input. Each convolutional layer was followed by a max-pooling layer applied with 2×2 region. The pooling layers acted as downsampling layers with generation of multiple pooled maps. The last two pooling layers were followed by a dropout layer with a dropout rate of 0.5, meaning 50% of the nodes would be dropped out in the layers for ensuring regularisation, thereby preventing overfitting. Next, the pooled feature maps were flattened to a 1D vector as input to the subsequent fully connected layer with 256 neurons. Before the final layer, a batch normalisation layer was included before the dropout layer to further improve the regularisation of the model. The final layer is the output layer with 3 nodes incorporating softmax activation function to determine the probabilities of each possible class of the classification task. Finally, a vector consisting of probabilities belonging to the AD, CN, and MCI classes was obtained as the final classification result.

The model from scratch has a total of 4.37 million parameters, wherein 4370483 parameters were trainable and 512 parameters were not trainable contributed by the batch normalisation layer whose mean and variance were updated during through layer updates instead of gradient descent (Ioffe & Szegedy, 2015). In Keras, the instance of BatchNormalization has both trainable and non-trainable parameters. It has 4 parameters which are gamma weights, beta weights, moving mean, and moving variance. The first two parameters are trainable through gradient descent which contributes 512 parameters, while the last two parameters are not trainable which contributes to the remaining 512 parameters.

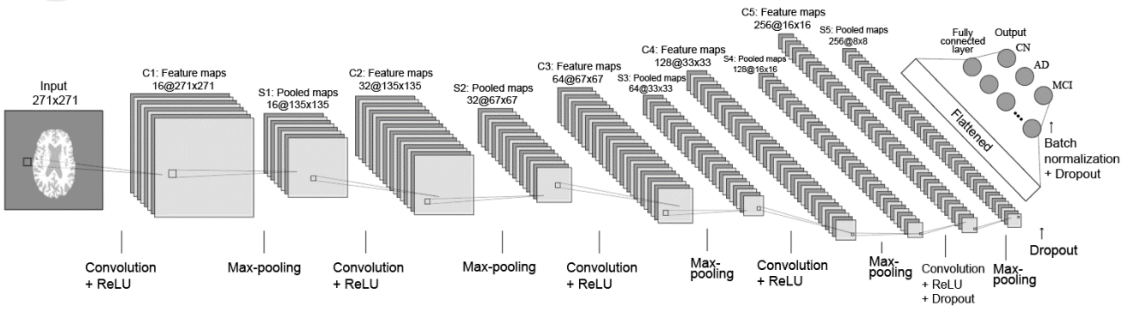

Figure 3.16: Layout of CNN trained from scratch

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 271, 271, 16)	80
max_pooling2d (MaxPooling2D)	(None, 135, 135, 16)	0
conv2d_1 (Conv2D)	(None, 135, 135, 32)	2080
max_pooling2d_1 (MaxPooling2	(None, 67, 67, 32)	0
conv2d_2 (Conv2D)	(None, 67, 67, 64)	8256
max_pooling2d_2 (MaxPooling2	(None, 33, 33, 64)	0
conv2d_3 (Conv2D)	(None, 33, 33, 128)	32896
max_pooling2d_3 (MaxPooling2	(None, 16, 16, 128)	0
dropout (Dropout)	(None, 16, 16, 128)	0
conv2d_4 (Conv2D)	(None, 16, 16, 256)	131328
max_pooling2d_4 (MaxPooling2	(None, 8, 8, 256)	0
dropout_1 (Dropout)	(None, 8, 8, 256)	0
flatten (Flatten)	(None, 16384)	0
dense (Dense)	(None, 256)	4194560
batch_normalization (BatchNo	(None, 256)	1024
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 3)	771

Total params: 4,370,995
Trainable params: 4,370,483
Non-trainable params: 512

Figure 3.17: Summary of CNN trained from scratch

3.4.2 VGG-16

In this project, the pre-trained VGG-16 model was used in the form of feature extractor. In 2014, VGG-16, a very deep CNN model was proposed by Karen Simonyan and Andrew Zisserman from Visual Geometry Group Lab of Oxford University won the ILSVRC 2014 competition (Simonyan & Zisserman, 2014).

The architecture of VGG-16 is appealing due to its uniform architecture. The model requires input in the form of image with a fixed dimension of 224×224×3. The overall architecture consists of 5 blocks of convolutional layer and max-pooling layer followed by a dense classifier that outputs 1000 class scores. The first block consists of two layers of convolutional layers with 64 filters of size 3×3 and “same” padding followed by a max-pooling layer with size of 2×2. The second block is similar to the first block, but both the convolutional layers have 128 filters of size 3×3. The last three blocks, each of them,

composed of 3 convolutional layers and subsequently a max-pooling layer. The third block has 3 convolutional layers of filter size 3×3 and 256 filters and a max-pooling layer. Then, the last two blocks have the same set of layers which include 3 convolutional layers, each has filter size of 3×3 and 512 filters and a subsequent max-pooling layer. After a series of convolution and max pooling operations, the resultant feature map has shape of (7, 7, 512). The feature map is flattened to a 1D feature vector with size of (1, 25088). Here, there are 3 FC layers. The first and second layers have 4096 neurons, while the third layer has 1000 neurons with softmax activation to output 1000 class scores.

VGG-16 comes with 138,357,533 (138 million) trainable parameters attributed to the vast amount of neurons in the fully connected layers, making it one of the largest CNN architecture. Training VGG-16 from scratch can be challenging, slow, and computationally costly. However, transfer learning technique allowed VGG-16 with pre-trained weights to be used for feature extraction of images from other domains.

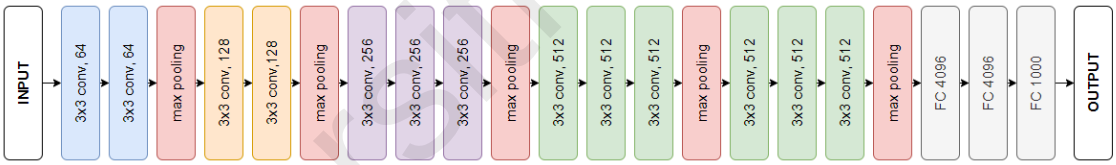

Figure 3.18: Block diagram of VGG-16

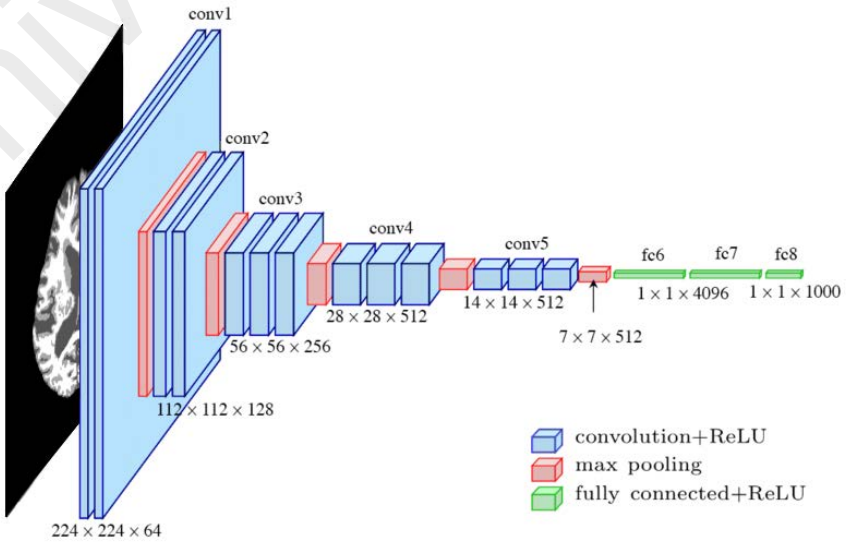

Figure 3.19: Overview of VGG-16 architecture

In this project, VGG-16 with pre-trained weights was used as a bootstrap feature extractor for feature extraction from the pre-processed brain sMRI images. The extracted features were then directed to a new classifier, which is trained from scratch (see Figure 3.20). To implement transfer learning using the pre-trained VGG-16 model for the 3-way classification task, the original densely connected classifier was removed since its output was used to generate 1000 class scores of ImageNet images classification. Also, the information captured by the fully connected layers might not be useful for addressing the problem of this project as the ImageNet classes are not related to the AD domain (Simonyan & Zisserman, 2014). Representations found in the fully connected layers are unable to tell where objects are located in input image. Figure 3.21 illustrates the part of the densely connected classifier of the pre-trained VGG-16 model being discarded.

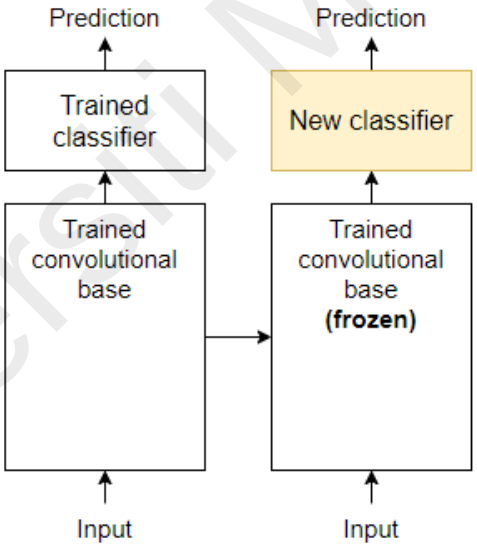

Figure 3.20: Swapping classifiers while keeping the same convolutional base

Figure 3.21: Layers removed from the pre-trained VGG-16

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 271, 271, 3)]	0
block1_conv1 (Conv2D)	(None, 271, 271, 64)	1792
block1_conv2 (Conv2D)	(None, 271, 271, 64)	36928
block1_pool (MaxPooling2D)	(None, 135, 135, 64)	0
block2_conv1 (Conv2D)	(None, 135, 135, 128)	73856
block2_conv2 (Conv2D)	(None, 135, 135, 128)	147584
block2_pool (MaxPooling2D)	(None, 67, 67, 128)	0
block3_conv1 (Conv2D)	(None, 67, 67, 256)	295168
block3_conv2 (Conv2D)	(None, 67, 67, 256)	590080
block3_conv3 (Conv2D)	(None, 67, 67, 256)	590080
block3_pool (MaxPooling2D)	(None, 33, 33, 256)	0
block4_conv1 (Conv2D)	(None, 33, 33, 512)	1180160
block4_conv2 (Conv2D)	(None, 33, 33, 512)	2359808
block4_conv3 (Conv2D)	(None, 33, 33, 512)	2359808
block4_pool (MaxPooling2D)	(None, 16, 16, 512)	0
block5_conv1 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block5_pool (MaxPooling2D)	(None, 8, 8, 512)	0

Total params: 14,714,688

Figure 3.22: Architecture summary of VGG-16 convolutional base

After removing the densely connected classifier, the remaining convolutional base had a total of 14.7 million parameters as depicted in Figure 3.22. A newly designed densely connected classifier was added on top of the convolutional base in which there are 2 FC layers with 256 neurons and 128 neurons, respectively. A dropout layer with a dropout ratio of 0.5 was added after every dense layer to combat the overfitting problem. Also, a batch normalisation layer was included after the first fully connected layer. The last fully connected layer has 3 nodes with softmax activation to output prediction of the 3-way classification task. The finalised VGG-16 model ready for transfer learning is shown in Figure 3.23.

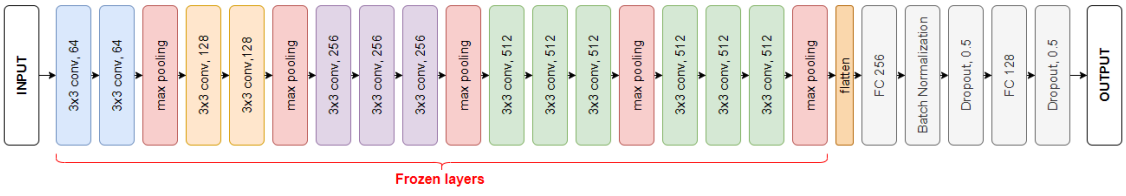

Figure 3.23: Block diagram of the pre-trained VGG-16 for transfer learning

It is important to note that the dataset containing grayscale images could not be directly fed to the VGG-16 model since it is a pre-trained model and its input configuration cannot be modified. VGG-16 requires input in the form of RGB image consisting 3 channels. However, grayscale image has just 1 channel. The naive solution is to repeat all the image arrays in the dataset 3 times on a new dimension. Hence, the same image would be over all 3 channels. This was done by specifying the colour mode as 'rgb' in the flow_from_directory method from the Keras library.

Before compiling the model, the convolutional base or all the pre-trained layers before the densely connected classifier were frozen. Freezing is mandatory to avoid weights of pre-trained layers being updated during training which will modify the representations that were previously learned by the convolutional base. In Keras, the *trainable* attribute of every layer in the convolutional base was set to False to freeze the layers.

After adding the densely connected classifier on top of the frozen convolutional base, the total parameters were 23.1 million, wherein 8.4 million parameters were trainable and 14.7 million non-trainable parameters, including 512 parameters that were non-trainable contributed by the batch normalisation layer. Moreover, the number of trainable layers decreased from 34 to 8 after freezing.

After compiling, the model was trained end to end with the frozen convolutional base integrated with the new densely connected classifier. The pre-trained layers processed the dataset and extracted visual representations for prediction. The outputs of the last pooling layer were then flattened into a 1D feature vector with size of (1, 32768) and directed to the densely connected classifier. The fully connected layers used the extracted features for training and output array of probability that adds up to 1 as the prediction results.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Model)                (None, 8, 8, 512)         14714688
_________________________________________________________________
flatten (Flatten)            (None, 32768)             0
_________________________________________________________________
dense (Dense)                (None, 256)               8388864
_________________________________________________________________
batch_normalization (BatchNo (None, 256)               1024
_________________________________________________________________
dropout (Dropout)            (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 387
=================================================================
Total params: 23,137,859
Trainable params: 8,422,659
Non-trainable params: 14,715,200
_________________________________________________________________
No. of trainable layers before freezing = 34
No. of trainable layers after freezing = 8
```

Figure 3.24: VGG-16 architecture summary for transfer learning

3.4.3 ResNet-50

Similar to VGG-16, the pre-trained ResNet-50 model was as a feature extractor and swapped a new densely connected classifier for prediction. Residual Networks are the first deeper neural network that enabled the training of hundreds or even thousands of layers while maintaining compelling performance (He et al., 2016). In 2015, a variant of the ResNet model, which is ResNet-152 composed of 152 layers, won the ILSVRC 2015 challenge.

Training deep neural networks are challenging because stacking of more layers causes the notorious vanishing gradient or also known as exploding gradient problem. While the gradient is back-propagating back to the earlier layers, multiplication operations that are done repetitively may cause the gradient infinitely small. Consequently, network performance becomes saturated or starts to degrade substantially as the network goes deeper.

The main reason of ResNet is able to train such deep network is that it has recipe that is not previously seen in other deep neural network called the residual connections. Figure 3.25 illustrates the residual block implemented in ResNet that skips one or more layers.

Residual block allowed ResNet to connect the previous layer to the current layer as well as the layer at the back of the previous layer. As a result, each layer can see more than just its previous layer's observations. In addition, the batch normalisation layer in ResNet is placed after every convolutional layer. Batch normalisation normalises layer weights and thus, higher learning rates can be used during training. This helps to train deep networks faster and minimise vanishing gradient problem.

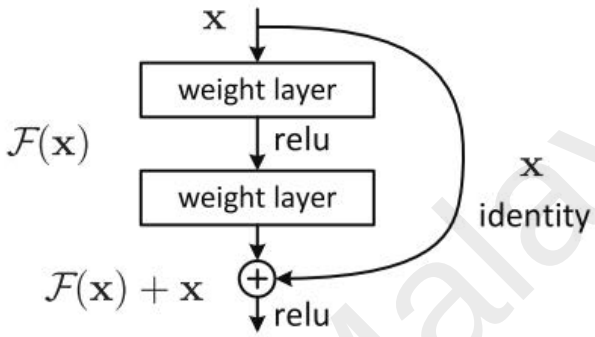

Figure 3.25: Residual block

ResNet-50 as a variant of ResNet model has 48 convolutional layers, 1 layer of max pooling, and 1 layer of average pooling. The residual block used in ResNet-50 has a slight tweak compared to its other variant. Instead of skipping two connections as shown in Figure 3.25, ResNet-50 uses shortcut connections that skip three layers.

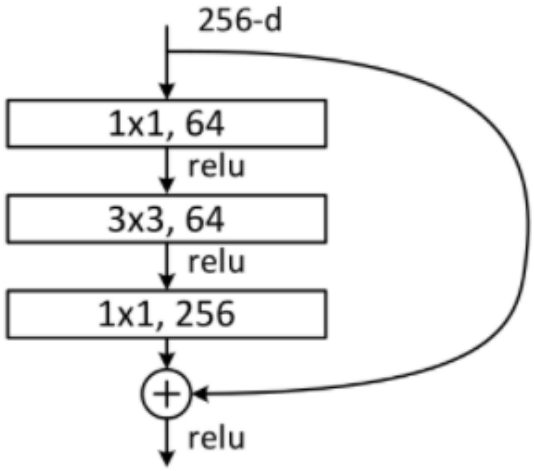

Figure 3.26: Residual block used in ResNet-50

Like VGG-16, ResNet-50 can be divided into 5 different blocks for clarity. Table 3.5 summarises the model architecture where FLOPs is the floating-point operations per

second. The architecture requires input size in the form of image with a fixed dimension of 224×224×3. The first block consists of 1 convolutional layer with filter size of 7×7 and 64 filters. The second block start with a max-pooling layer of size 3×3. Subsequently, there are 3 convolutional layers with size of 1×1, 64 filters followed by size of 3×3, 64 filters and at last size of 1×1, 256 filters. These 3 layers are repeated 3 times, resulting in 9 layers in this step. In the third block, there are 3 layers with filter of 1×1, 128 after that a filter of 3×3, 128 and at last a filter of 1×1, 512. This step is repeated 4 times resulting a total of 12 layers. Next, the fourth block has 3 layers with filter of 1×1, 256 after that a filter of 3×3, 256 and at last a filter of 1×1, 1024, and this is step is repeated 6 time, giving a total of 18 layers. Then again, 3 layers with filter of 1×1, 512 after that a filter of 3×3, 512 and at last a filter of 1×1, 2048. This step is repeated 3 times and yielding a total of 9 layers. Then, this is followed by a layer of average pooling and a 1000 nodes fully connected layer at the end with softmax activation that outputs 1000 class scores.

Table 3.5: ResNet-50 architecture summary

Layer name	Output size	Layer description
conv1	112×112	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2
		$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times 3$
conv3	28×28	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times 4$
conv4	14×14	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times 6$
conv5	7×7	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times 3$
	1×1	Average pool, 1000 nodes fully connected layer, softmax
FLOPs	–	3.8×10^9

Note. Table is adapted from He et al. (2016).

Similar to the methodology as seen for VGG-16, ResNet-50 with pre-trained weights was used as a feature extractor to extract important features from the dataset. The extracted features were then directed to a new densely connected classifier, which is trained from scratch. The original densely connected classifier was removed and swapped with a new classifier same as the one used in VGG-16. The extracted features were flattened to a 1D vector and channelled to the classifier. The first layer is a FC layer with 256 neurons followed by a batch normalisation layer and a dropout layer with a 0.5 dropout rate. This is followed by another fully connected layer with 128 neurons and a dropout layer with a dropout ratio of 0.5. The last fully connected layer has 3 nodes with softmax activation to output prediction of the 3-way classification task.

After removing the densely connected classifier, the remaining convolutional base had a total of 23.6 million parameters which can be seen in Appendix A. Then, after adding the densely connected classifier on top of the convolutional base, the total number of parameters increased to 66.1 million. Again, it is important to freeze the convolutional base to prevent the layers in it from being updated. The number of trainable parameters was 42.5 million after the convolutional base was frozen, and the number of trainable weights decreased from 220 to 8. The number of non-trainable parameters was 23.6 million contributed by the frozen layers and mean and variance of batch normalisation layer. The total of non-trainable parameters contributed by all the batch normalisation layers is 53635 parameters. The finalised ResNet-50 model ready for transfer learning is shown in Figure 3.27.

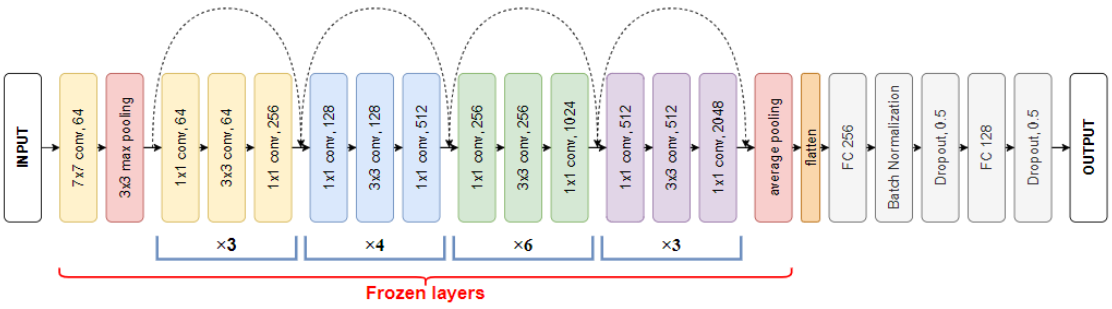

Figure 3.27: Block diagram of the pre-trained ResNet-50 for transfer learning

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 9, 9, 2048)        23587712
_________________________________________________________________
flatten (Flatten)            (None, 165888)            0
_________________________________________________________________
dense (Dense)                (None, 256)               42467584
_________________________________________________________________
batch_normalization (BatchNo (None, 256)               1024
_________________________________________________________________
dropout (Dropout)            (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 387
=================================================================
Total params: 66,089,603
Trainable params: 42,501,379
Non-trainable params: 23,588,224
_________________________________________________________________
No. of trainable weights (before freezing): 220
No. of trainable weights (after freezing): 8
```

Figure 3.28: ResNet-50 architecture summary

3.5 Hyperparameter Tuning

This stage is often referred to as hyperparameter tuning, which is performed to optimise model performance with the best combination of hyperparameters. Hyperparameters are training variables whose value is set manually before starting the learning process. The validation dataset is what the model used for evaluation after every set of predictions. It helps the model to tune its hyperparameters.

Instead of manually tuning the hyperparameters by trial and error, Random search strategy was implemented using the Keras Tuner. A search space was defined that include all hyperparameters that need to be optimised and the desired range. Random search picked random sample points and fed them in different combinations through the algorithm to the model and reported back the combination with the best accuracy. It could provide ease of exploring search space with a great number of parameters and parameter values than Grid search strategy (Bergstra & Bengio, 2012). With a large pool of hyperparameters, it increases the possibility of finding the optimal hyperparameters, at the same time, does not increase computational cost.

The hyperparameters that were optimised are number of nodes for fully connected layer, and learning rate. The best combination of hyperparameters for each model was chosen for training as summarised in Table 3.7 and also implemented and discussed in sub-chapter 3.4 and 3.7.

Table 3.6: Hyperparameters search space for random search strategy

Hyperparameters	Search space
Number of nodes for dense layer 1	64, 128, 256, 512, 1024
Number of nodes for dense layer 2	64, 128, 256, 512, 1024
Learning rate	10^{-5}, 5^{-5}, 10^{-4}, 5^{-4}, 10^{-3}

Table 3.7: Summary of the best combination of hyperparameters

Model	Number of nodes for dense layer 1	Number of nodes for dense layer 2	Learning rate
Scratch	256	-	10^{-4}
VGG-16	256	128	10^{-5}
ResNet-50	256	128	10^{-4}

3.6 Performance evaluation

Evaluation metrics were adopted to gauge the classification performance of the multiclass classifier tested on testing data. Specifically, performance metrics that were utilised are (1) accuracy, (2) precision, (3) recall, and (4) F1-score, specifically selected due to their popularity in DL studies and bioinformatics literature. Equations (3.4) to (3.7) show the formula of each performance metric wherein TP and TN are the number of positive and negative instances identified correctly, respectively. FP and FN are the numbers of misclassified positive and negative cases, respectively.

Accuracy is the number of correctly classified samples divided by the total number of samples. It gauges the overall performance of the model in giving a correctly classified sample.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.4}$$

Recall, also known as sensitivity, is the ratio of correctly classified samples to all the ground truth, which gives the true positive rate.

$$Recall = \frac{TP}{TP + FN} \tag{3.5}$$

Analogously, precision can be defined as the true negatives which are correctly identified.

$$Precision = \frac{TP}{TP + FP} \tag{3.6}$$

F1-score is the harmonic mean of precision and recall for measuring the classification performance of a model on a dataset. In contrast to accuracy, F1-score emphasises the false negatives and false positives. A perfect F1-score value is 1.0, indicating a model achieved perfect precision and recall.

$$F1 - score = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \tag{3.7}$$

3.7 Experimental Setups

All the deep learning models were built using Keras, an open-source high-level neural network API for building deep models, with TensorFlow as backend (Abadi et al., 2016; Chollet, 2021). Keras was chosen as it allows fast prototyping and parallel computing using GPU. In this project, training, validation and testing routines were performed on Google Colab to execute Python 3 codes for data pre-processing and developing CNN model. It offered several GPU models such as the NVIDIA Tesla K80, T4, P4, and P100. The model of GPU would be given at random based on availability on Google Colab. There was no published usage limit on idle timeout period, RAM size, and disk size. Most

of the time, RAM size of about 13GB and disk size of around 70GB would be allocated for GPU accelerated runtime.

Three different models were trained on training data of 3927 images as described in Table 3.4 to do multiclass (3-way) classification of AD-CN-MCI. The initial network weights were initialised randomly without using any weight initialisation technique. Adam optimiser with default parameters (β_1 = 0.9, β_2 = 0.999, and $\varepsilon = 1^{-7}$) was implemented with empirically decided learning rate of 0.0001. The Adam stochastic optimiser was preferred due to its low memory requirement, efficient computation, and suitable for a problem with large data (Kingma & Ba, 2014).

Batch size is commonly restricted by GPU memory (Abrol et al., 2020). It was observed that batch size had little to no degradation on performance. Hence, the batch sizes used for the models were set at the maximum value that could be handled by the GPU device memory to speed up computations. Batch size of 512 in 75 epochs was used for the model trained from scratch, whereas a batch size of 256 in 75 epochs was used for VGG-16 and ResNet-50. Epoch is defined as the number of passes of the whole dataset to the deep learning model. In this case of this project, it required 8 steps to complete 1 epoch for the CNN from scratch. For VGG-16 and ResNet-50, it took 15 steps per epoch.

For the training and validation process, the evaluation metrics chosen were accuracy, precision, and recall, which is also implemented for testing performance. Furthermore, the loss function used was categorical cross-entropy. Cross-entropy is also known as log loss function, suitable for measuring model classification performance whose output is an array of class scores in the range of 0 to 1. Cross-entropy loss decreases as the predicted output converges to the actual label. In the case of class number more than 2 (multiclass), categorical cross-entropy loss formula can be derived as:

$$L(y, p) = -\sum_{c=1}^{N} \log(p_{o,c}) \tag{3.8}$$

where N is the number of classes, y is the actual value, and p is the predicted value.

Additionally, to ease the process of model training, 2 types of 'callbacks' in Keras were implemented during training such as EarlyStop and ModelCheckpoint. EarlyStop allowed the models to stop training when their performance do not improve over 5 epochs by monitoring the validation loss. This is one of the approaches to prevent a model from overfitting. Next, ModelCheckpoint ensured that models always save the best weights while training to prevent loss of progression. Saving the weights is more efficient than saving the information of the entire model as a large network like VGG-16 could take up at least 500MB of memory.

Table 3.8: Training parameters for the model trained from scratch

Parameter	Value
Number of epochs	100
Batch size	512
Weight initialiser	Xavier uniform
Optimiser	Adam
Adam parameters	$\beta_1 = 0.9$, $\beta_2 = 0.999$
Learning rate	10^{-4}
Loss function	Categorical cross-entropy
Metrics	Accuracy
Data augmentation	Rotation, zoom, height shift, width shift, shear, horizontal flip

Table 3.9: Training parameters for VGG-16

Parameter	Value
Number of epochs	100
Batch size	256
Weight initialiser	Xavier uniform
Optimiser	Adam
Adam parameters	$\beta_1 = 0.9$, $\beta_2 = 0.999$
Learning rate	10^{-5}
Loss function	Categorical cross-entropy
Metrics	Accuracy
Data augmentation	Rotation, zoom, height shift, width shift, shear, horizontal flip

Table 3.10: Training parameters for ResNet-50

Parameter	Value
Number of epochs	100
Batch size	256
Weight initialiser	Xavier uniform
Optimiser	Adam
Adam parameters	$\beta_1 = 0.9$, $\beta_2 = 0.999$
Learning rate	10^{-4}
Loss function	Categorical cross-entropy
Metrics	Accuracy
Data augmentation	Rotation, zoom, height shift, width shift, shear, horizontal flip

3.8 Summary

The dataset was retrieved from the ADNI database, which is composed of 1.5T T1 weighted sMRI volume. All the sMRI volumes were pre-processed using a series of pre-processing functions. For example, skull stripping, N4 bias field correction, tissue segmentation, extraction of 2D images from 3D volumes, pixel value normalisation, and data augmentation. 20% of the dataset was reserved as testing data. The remaining data were used for train-validation split with a ratio of 80:20. After necessary pre-processing and splitting, the dataset consisted of 6138 images, in which 3927 were training images, 984 were validation images, and 1227 were testing images.

Table 3.11: Summary of ADNI dataset

Class label	Training set	Validation set	Testing set	Total images
CN	1308	328	408	2044
MCI	1307	327	409	2043
AD	1312	329	410	2051
Total	3927	984	1227	6138

To address the problem of classifying AD, CN, and MCI, which is a multiclass or 3-way classification problem, 3 different models were implemented, including a CNN

trained from scratch, VGG-16, and ResNet-50. The parameters used for training for each model is summarised in Table 3.12.

Table 3.12: Summary of training parameters for every model

Parameter	Value		
	Scratch	**VGG-16**	**ResNet-50**
Number of epochs	100	100	100
Batch size	512	256	256
Weight initialiser	Xavier uniform	Xavier uniform	Xavier uniform
Optimiser	Adam	Adam	Adam
Adam parameters	$\beta_1 = 0.9$, $\beta_2 = 0.999$	$\beta_1 = 0.9$, $\beta_2 = 0.999$	$\beta_1 = 0.9$, $\beta_2 = 0.999$
Learning rate	10^{-4}	10^{-5}	10^{-4}
Loss function	Categorical cross-entropy	Categorical cross-entropy	Categorical cross-entropy
Metrics	Accuracy	Accuracy	Accuracy
Data augmentation	Rotation, zoom, height shift, width shift, shear, horizontal flip	Rotation, zoom, height shift, width shift, shear, horizontal flip	Rotation, zoom, height shift, width shift, shear, horizontal flip

CHAPTER 4: RESULT AND DISCUSSION

4.1 Experimental Results

Following the experimental setup, the experimental results of multiclass classification of AD-CN-MCI are presented in this section. The experimental results are separated into performance of training and validation and classification performance on test data. After that, a comparison is made between the best-performing model and the state-of-the-art model in recent literature.

4.1.1 Training and Validation Performance

Table 4.1 reports the training and validation performance of the 3 different CNN models being experimented. Using the python Matploblib library, graphs of accuracy and loss on training and validation against epochs were plotted for the 3 separate models. From Figure 4.1 to Figure 4.3, the plot on the left shows the accuracy plot while the plot on the right shows the plot of loss function for the 3-way classification task. Learning curves are extensively used to diagnose deep learning algorithms. In specific, examining the learning curves of a model during training can allow diagnosis of learning problems such as underfit, overfit or even unrepresentative data.

Table 4.1: Summary of training and validation performance

Model	Training time	Epoch	Training		Validation	
			Accuracy	Loss	Accuracy	Loss
Scratch	46 mins	97	0.8755	0.3102	0.7270	0.7094
VGG-16	1 hour 15 mins	57	0.9492	0.1511	0.8066	0.5263
ResNet-50	1 hour 31 mins	56	0.9164	0.2150	0.7686	0.5901

All the model training was performed with early stopping with a patience level of 15 epochs as an approach to prevent overfitting. The training and validation routines were halted at a point when the validation loss started to degrade. The CNN trained from scratch took 46 minutes to finish training which is the fastest among the 3 models. While for deep CNN like VGG-16 and ResNet-50 with multiple stacking layers, training can be

computationally expensive and required much more training time than the shallow model trained from scratch. ResNet-50 spent a duration of 1 hour and 31 minutes to complete training in 56 epochs, which is 16 minutes slower than VGG-16 that took 1 hour and 15 minutes to train in 57 epochs. The longer training time of ResNet-50 can be associated with a large number of trainable parameters. The developed ResNet-50 model with a frozen convolutional base and swapped densely connected classifier comprised of 42.5 million trainable parameters. On the other hand, the VGG-16 model with an identical densely connected classifier and a frozen convolutional base that had 8.4 million trainable parameters. Interestingly, ResNet-50 expended only 21.33% more period of time than VGG-16 on training despite the fact that it had 5 folds of the number of trainable parameters compared to VGG-16. The main reason behind this could be the inclusion of multiple batch normalisation layers between convolutional layer and non-linear activation function, thereby allowing a higher learning rate to be used (He et al., 2016).

Overall, the training performance for all 3 models was satisfactory. All 3 models were able to reach a training accuracy of 99% without early stopping but resulted in overfitting on validation data. The training plot provided information about the learning process going on within a model. A good training learning curve should have an arc line that curves upward and converge when accuracy approaches 100%. However, good training performance does not matter if the model does not generalise well on test data. The training plot shown in Figure 4.2 illustrates a good learning curve that shows improvement of learning performance over epoch. During the early learning stage, training plots in Figure 4.1 to Figure 4.3 show that the validation accuracy curves are increasing with the training accuracy curves. During the mid-stage of learning, the validation accuracy curve begins to have smaller increments and eventually plateau while the training accuracy curves continue to improve until they are approaching 100% accuracy. VGG-16 attained the highest validation accuracy of 80.66%, followed by

ResNet-50 with validation accuracy of 76.86% and the CNN from scratch with a validation accuracy of 72.70%.

The cost or loss function quantifies the performance of a model at classifying the input images from the dataset. Loss value indicates how well a model behaves after each epoch of optimisation. While training DL network, the target is to minimise the error calculated using the loss function, at the same time ensuring an increase in testing accuracy. VGG-16 converged at a training loss value of 0.1511, while CNN from scratch and ResNet-50 were able to reach training loss values of 0.3102 and 0.2150, respectively. For the performance of validation measured in terms of loss value, VGG-16 achieved the lowest loss value of 0.5263. ResNet-50 achieved the second-highest loss value of 0.5901 followed by CNN from scratch with loss value of 0.7094.

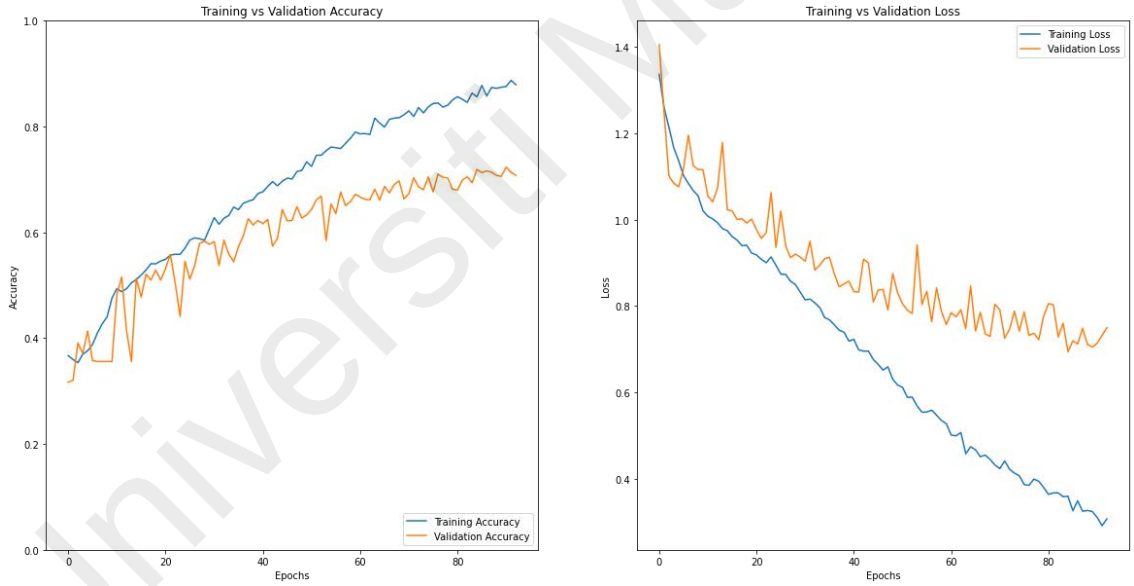

Figure 4.1: Plot of accuracy and loss for CNN from scratch

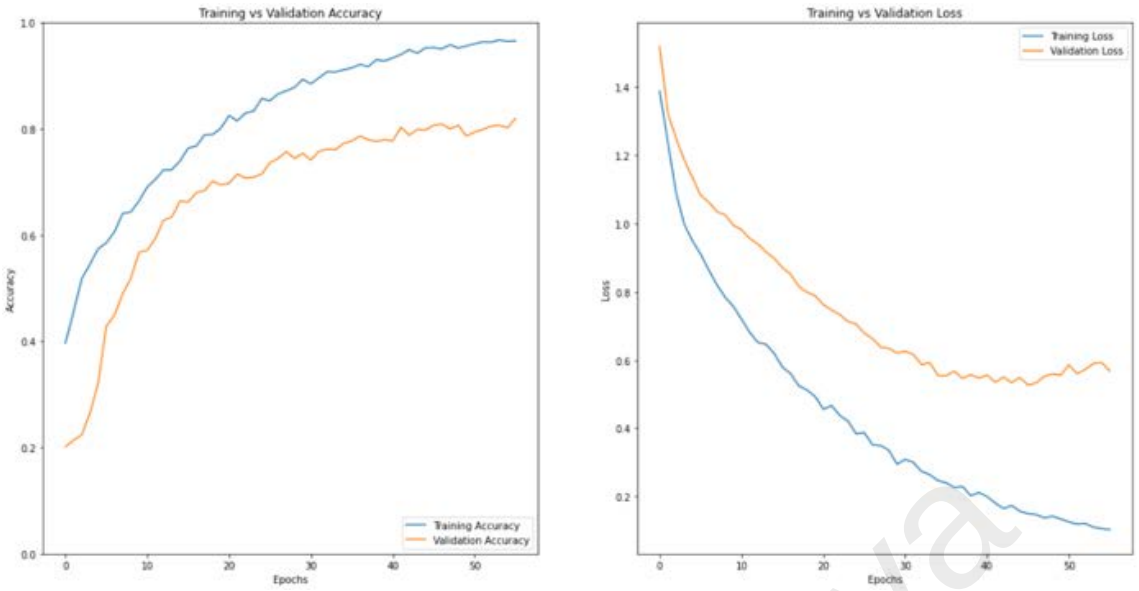

Figure 4.2: Plot of accuracy and loss for VGG-16

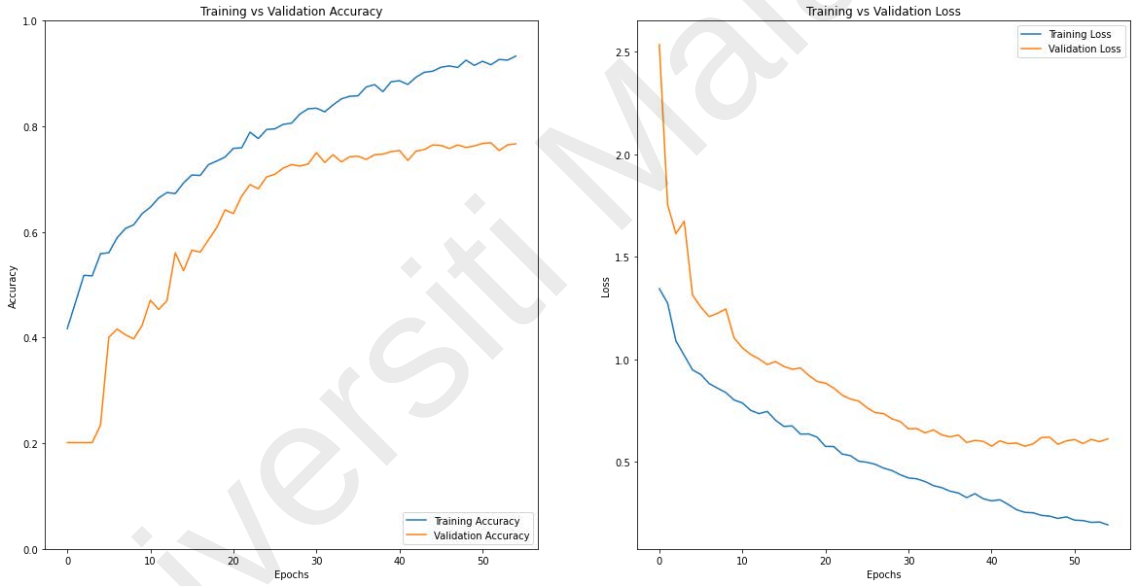

Figure 4.3: Plot of accuracy and loss for ResNet-50

Looking at the plot of training versus validation loss, it is observed that the 3 models have a similar graph trend. Generalisation gap is shown in between the training and validation loss curves as visualised in Figure 4.4 as an indication of overfitting. Even though the training loss does converge to zero, but the validation loss has decreased to a minimum and has begun to increase. At this point, early stopping was implemented to prevent further overfitting of the models. These models had extracted all the information

that they could learn and are deemed to be more specialised to the training data than they are able to generalise to the validation data.

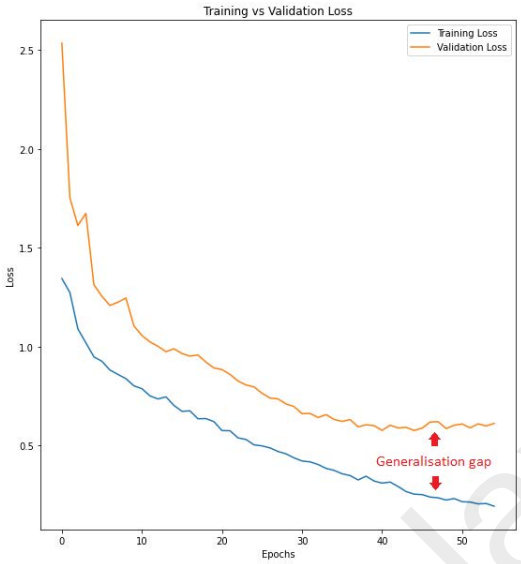

Figure 4.4: Illustration of generalisation gap

Among the 3 models, it is observed that CNN from scratch has the most serious overfitting problem as observed from the largest gap between the training and validation loss curve. This could be due to the small dataset and the nature of the shallow model with small capacity to learn to classify a difficult problem. Regularisation methods such as dropout and batch normalisation were included and fine-tuned to find the best hyperparameters. The problem of overfitting improved slightly with regularisation, but the sign of overfitting was still significant.

Thus, transfer learning method was experimented because of its ability to provide satisfying results on small datasets as seen in recent literature. Deep models such as VGG-16 and ResNet-50 with pre-trained weights were used for feature extraction without learning the convolutional bases from scratch. For classification, a new densely connected classifier trained from scratch was added for both models to output classification scores. In this case, both VGG-16 and ResNet-50 are found to perform better than the CNN trained from scratch. However, overfitting problem still persists in both models despite

different regularisation methods were being applied such as dropout, batch normalisation, and data augmentation.

4.1.2 Testing Performance

After all the models were trained and validated, the 20% held out testing data were tested on each and every model. Confusion matrix was used as a tool to examine model classification performance with a summary of prediction results. The number of predictions that are correctly or incorrectly predicted is summarised systematically in a table with count values broken down by each class. Since it is a 3-way classification task that has 3 different classes, the confusion matrix is a table with 3 rows and 3 columns. The rows (y-axis) are taken as predicted lab and the columns (x-axis) are taken as the predicted label. Figure 4.5, Figure 4.6, and Figure 4.7 depicted confusion matrices that describe the performance of classification on test data for each model. Each of the confusion matrices is visualised as a colour-coded heatmap using the seaborn library. It can be observed that all the plotted confusion matrices have darker cells for the diagonal elements. This indicates that a large amount of data is being predicted correctly to their respective label. Conversely, the off-diagonal elements with light shades indicate misclassifications done by the model.

CNN from scratch predicted the MCI group with the highest accuracy and the CN group with the lowest accuracy. It classified 304 out of 409 MCI images and 291 of 408 CN images correctly. In contrast, VGG-16 and ResNet-50 have AD group with the highest classification accuracy while MCI group has the lowest classification accuracy. VGG-16 predicted 344 images, and ResNet-50 classified 341 images of AD out of 410 AD images. While for MCI group, VGG-16 predicted 288 images, and ResNet-50 classified 282 images of AD out of 409 AD images correctly.

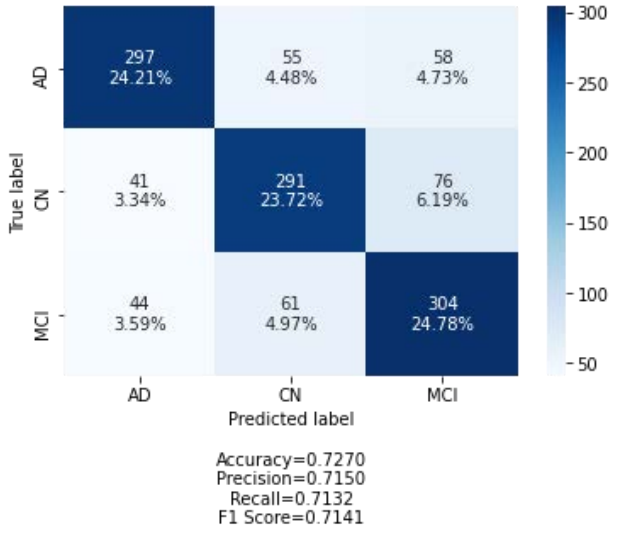

Figure 4.5: Confusion matrix for CNN from scratch

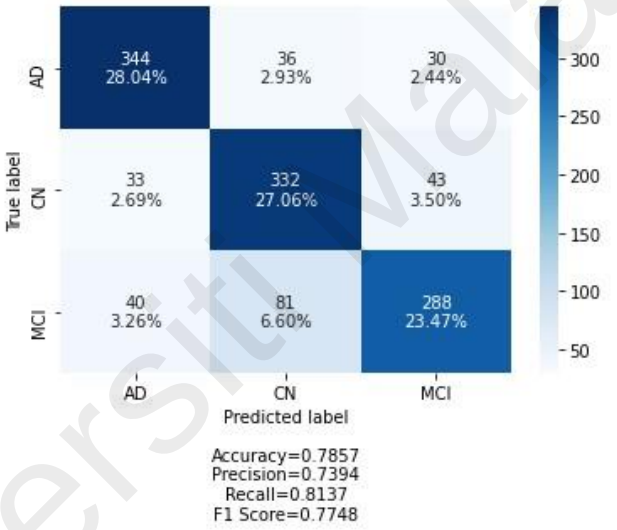

Figure 4.6: Confusion matrix for VGG-16

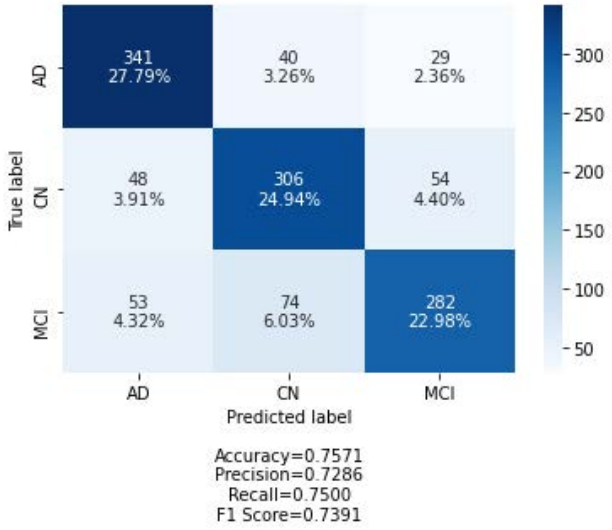

Figure 4.7: Confusion matrix for ResNet-50

To further evaluate the classification model, classification metrics such as accuracy, precision, recall, and F1-score were calculated with the aid of the confusion matrices. The formula for each metric is stated in sub-chapter 3.6. The results shown in Table 4.2 below is correlated with the graph trends shown in the plots of training and validation. For each classification model (CNN from scratch, VGG-16, and ResNet-50), the reported classification performance on test data is accuracy of 72.70%, 78.57%, and 75.71% respectively, precision of 71.50%, 73.94%, and 72.86% respectively, recall of 71.32%, 81.37%, and 75.00% respectively, and F1-score of 71.41%, 77.48%, and 73.91% respectively. Based on Figure 4.8, it is observed that VGG-16 that achieved the lowest loss value of 0.5263, performed the best on test data with accuracy of 78.57%. The lowest testing accuracy of 72.70% is obtained using the CNN from scratch.

Table 4.2: Accuracy, precision, recall, and F1-score of different models on test data

Model	Accuracy	Precision	Recall	F1-score
Scratch	0.7270	0.7150	0.7132	0.7141
VGG-16	0.7857	0.7394	0.8137	0.7748
ResNet-50	0.7571	0.7286	0.7500	0.7391

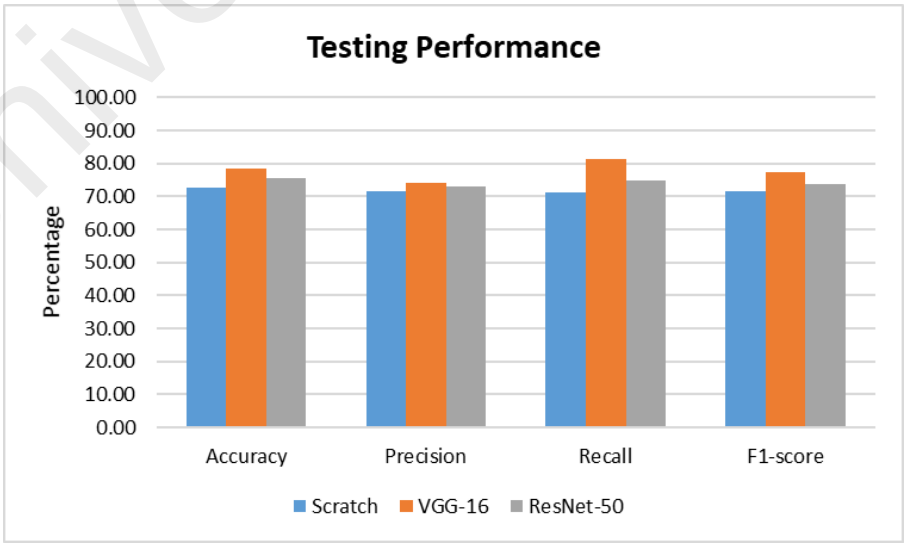

Figure 4.8: Comparison of classification performance on test data

For further in-depth evaluation of performance on test data, the classification results for each class label are reported in Table 4.3. Similar to what was being analysed using the confusion matrices, the AD group has the highest accuracy value for VGG-16 and ResNet-50. VGG-16 performed the greatest in predicting AD class with accuracy of 83.90%, precision of 82.49%, recall of 83.90%, and F1-score of 83.19%. Interestingly, ResNet-50 has the lowest accuracy on predicting the MCI class. Overall, using VGG-16 improved the performance values for all 3 classes.

Table 4.3: Testing accuracy, precision, recall and F1-score for all class label

Model	Class label	Accuracy	Precision	Recall	F1-score
Scratch	AD	0.7244	0.7775	0.7244	0.7500
	CN	0.7132	0.7150	0.7132	0.7141
	MCI	0.7433	0.6941	0.7433	0.7178
VGG-16	AD	0.8390	0.8249	0.8390	0.8319
	CN	0.8137	0.7394	0.8137	0.7748
	MCI	0.7042	0.7978	0.7042	0.7481
ResNet-50	AD	0.8317	0.7715	0.8317	0.8005
	CN	0.7500	0.7286	0.7500	0.7391
	MCI	0.6895	0.7726	0.6895	0.7287

4.1.3 Comparison with Previous Literature

In this section, the classification performance of AD-CN-MCI of the best-performed VGG-16 in this project is compared to the deep learning model discussed in recent literature as shown in Table 4.4. To identify previous works that performed similar multiclass classification (AD vs CN vs MCI), the PubMed electronic database was searched using specific keywords as follows: (“deep learning” OR “machine learning” OR “convolutional neural network” OR “CNN”) AND (“Alzheimer’s” OR “dementia”) AND (“prediction” OR “classification” OR “multiclass OR “multi-class” OR “mild cognitive impairment”) AND (MRI OR “magnetic resonance imaging” OR PET OR “positron emission tomography”). The result of the literature search was retrieved as of

August 28, 2021. Each of the search results was screened to identify the content of work that has applied deep learning to classify AD-CN-MCI.

Table 4.4: Summary of comparison with different models in previous works

Study	Type of CNN	Modalities	Dataset	Number of subjects	Architecture	Accuracy (%)
Proposed VGG-16	2D CNN	sMRI	ADNI	AD = 229 CN = 398 MCI = 192	VGG-16	78.57
(Gupta et al., 2013)	2D CNN	sMRI	ADNI	AD = 200 CN = 411 MCI = 232	Stacked autoencoder	85.00
(Payan & Montana, 2015)	2D CNN	sMRI	ADNI	AD = 755 CN = 755 MCI = 755	Sparse autoencoder	85.53
(Basheera & Sai Ram, 2020)	3D CNN	sMRI	ADNI	AD = 28 CN = 65 MCI = 32	3D CNN	86.70
(Payan & Montana, 2015)	3D CNN	sMRI	ADNI	AD = 755 CN = 755 MCI = 755	Sparse autoencoder	89.47
(Asl et al., 2018)	3D CNN	sMRI	ADNI	AD = 70 CN = 70 MCI = 70	3D deeply supervised adaptive CNN	94.80
(Jain et al., 2019)	2D CNN	sMRI	ADNI	AD = 40 CN = 50 MCI = 50	VGG-16	95.73

Several studies had conducted multiclass classification of AD-CN-MCI by extracting features and perform predictions using different methods. Study Jain et al. (2019) achieved the highest accuracy of 95.73% in this comparison. Similar to the proposed VGG-16 in this project, Jain and colleagues implemented pre-trained VGG-16 for extracting features and train a classifier consisting of one FC layer with 256 neurons followed by a dropout layer. The most distinctive part of their study is that they do not include any segmentation process in the pre-processing pipeline. For extraction of 2D slices from the 3D MRI volume, a sorting mechanism based on image entropy was incorporated to pick the top 32 slices with the most information.

Considering studies using 3D CNN, Basheera and Sai Ram (2020), Payan and Montana (2015), and Asl et al. (2018) reported higher performance than the majority of the studies using 2D CNN with accuracy of 86.70%, 89.47%, and 94.80% respectively. Study by Payan and Montana (2015) reported an increase in classification accuracy by 3.94% after switching from 2D CNN to 3D CNN. Since 3D CNN takes in 3D images as input, the boosted performance can be associated with 3D convolutions that are able to capture spatial 3D representations. Asl et al. (2018) employed 3 stacked 3D-convolutional autoencoder (3D-CAE) networks pre-trained on the CAD-Dementia dataset for feature extraction.

4.2 Discussion

From the results obtained, the VGG-16 model outperformed the CNN trained from scratch and the ResNet-50 model. It is of best testing performance with accuracy of 78.57%, precision of 73.94%, recall of 81.37%, and F1-score of 77.48%. Comparing its performance to other related works, VGG-16 has performance below the average. Being trained on the ImageNet dataset, VGG-16 was able to extract representations using its convolutional base for learning the multiclass classification task. Despite the great performance on learning the representations, VGG-16 still encountered the typical overfitting problem due to the small dataset used. Several regularisation methods were used such as dropout, batch normalisation, data augmentation, and early stopping. However, the sign of overfitting can still be noticed. This could be due to high complexity of the classification task. The subtle discrepancies between the MCI and AD images require a large amount of data to learn the representation to classify them. With the small dataset being used in this project, the VGG-16 model could not learn the problem completely, and hence the overfitting problem. Another possible reason could be the dataset being used in this project is very much different from the ImageNet dataset. The VGG-16 was pre-trained on very general images from the ImageNet which does not

include any medical images. Hence, the high-level features learned by the higher layers of the VGG-16 are not sufficient to differentiate the classes in this project.

Based on the relevant work and the results obtained, it is of importance to choose a proper training strategy for the model. Hence, the model is able to spend the least time training while trying to cover as many cases as possible. An adequate model capacity is essential for model generalisation. Model depth should be kept as small as possible to prevent a model from overfitting on training data. The greater the depth, the more cases that the model can memorise. As a consequence, the final system will perform worse on unseen data.

Another possible reason behind inferior performance could be insufficient data augmentation. The data augmentation used is not aggressive enough to create diversity for the original dataset. An example of aggressive data augmentation can be seen in the study by Basaia et al. (2019). Apart from general augmentation transformations such as rotation, zooming, and scaling, the study implemented deformation, cropping, and flipping.

Theoretically, transfer learning could ensure good results on a small dataset, but the high complexity of classifying AD-CN-MCI as reflected in the work of recent literature suggests that training on large data is always preferable for better model generalisation. While this project only used the axial brain images, all studies stated in Table 4.4 that have better performance included brain images in axial, coronal, and sagittal view. Study reported that the sagittal plane also contains typical manifestations of abnormalities of AD (Kumar et al., 2021). Small dataset is the main constraint that impedes the models from attaining a good performance which is further discussed in sub-chapter 4.3, and the best solution is to increase the number of data for training in future work.

The advantages of this work are elaborated as follows. In general, most of the studies emphasised performing binary classification of different phenotypes of AD. In this

project, 3 different classes (AD, CN, and MCI) are classified directly using a single classifier. This study is less common as most of the studies deal with the problem of multiple class labels by dividing the problem into several binary sub-problems. Moreover, tissue segmented sMRI brain images were used, which substantially lower the requirement of computational costs in terms of power and time. Secondly, MRI images were segmented into GM, WM, and CSF for training and testing the model. Moreover, models were tested using an independent set of images held out from the dataset. In addition, the performance of popular deep transfer learning models such as VGG-16 and ResNet-50 were evaluated to study their performance on images not from the ImageNet domain.

4.3 Limitations and Possible Solutions

This study is not without limitations. Here, multiple inherent limitations of the work are outlined alongside possible solutions that could be implemented in future work. Similar to other neuroimaging studies, one crucial constraint is a limited number of data available for training. The most common solution to address this problem is to apply data augmentation. This study did apply several data augmentation techniques such as rotation, zooming, scaling, and horizontal flip. However, it was suspected that the degree of augmentation is too small to achieve the purpose of increasing data diversity. It is expected that a more aggressive transformation for data augmentation could further improve model performance. Another problem regarding data augmentation is the increase in training time, where each epoch is taking longer to train. Since in-place data augmentation method was used, the augmentation process was done during the training routine. In Keras, the ImageDataGenerator class used for data augmentation utilise CPU for computation which is slower than GPU. Nevertheless, data augmentation is utilised not for speed but for improved performance. One possible solution is to perform data augmentation in the form of dataset generation and expansion prior to training. Every

single image in a dataset will undergo random transformations and generate new images for the dataset. For instance, 6138 images available for this study will be doubled with the generation of new images, resulting in a total of 12276 images. The expanded dataset with newly generated images with randomly applied transformation is saved in local memory which eliminates the use of CPU during training. Additionally, implementing the TensorFlow data module could solve this problem by prefetching the input data before the next computation step.

Classifying AD, CN, and MCI is a challenging task due to the subtle discrepancy among the classes. With the small dataset, it was avoidable that all the 3 models have significant overfitting problems despite data augmentation and other regularisation methods have been applied. Study reported increased classification performance by increasing data size (Casanova et al., 2012). In this study, image data was restricted to only the axial view of brain sMRI. With an increase in the availability of data, it is highly expected that using brain sMRI slices from all different views, including axial, coronal, and sagittal views, could minimise overfitting and provide improvement in classification performance. On the other hand, it is of importance to note that an increase in data size will increase computational time. The best-performed VGG-16 in this project took 1 hour and 15 minutes to finish training using Google Colab. If all the brain slices are included for training, it is estimated to take at least half a day for model training with the given setup. A better hardware setup equipped with multiple high-end GPUs might be required instead of using the virtual machine provided by Google Colab.

Cross-validation is a tool for talking overfitting, especially for a dataset with a small amount of data. It evaluates the model's ability to generalise to an independent dataset to gauge the performance of the algorithm and ensure better use of data. K-fold is the most extensively used cross-validation technique, wherein the 'k' parameter decides the number of equal-sized sections the dataset is going to be divided. Figure 4.9 visualises

the splitting of a dataset using 5-fold cross-validation. One by one, a section is selected as test/validation data and the rest will be the training data. Model is trained on k-1 sections and use the leftover one section for validation or testing. This process is repeated k times until all possible combination is evaluated.

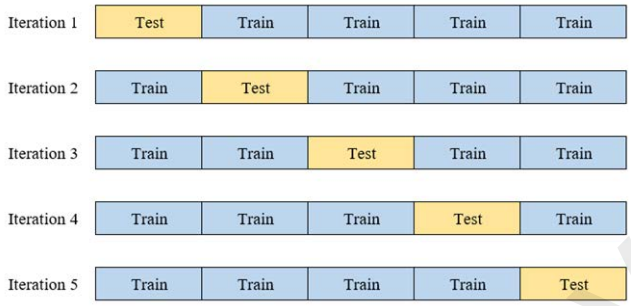

Figure 4.9: Illustration of 5-fold cross-validation

One of the main downsides of this project is not implementing cross-validation given its small dataset. Cross-validation was left aside because data leakage was noticed in the models. Data leakage happens when information outside of the training set is sourced for building the model, thereby developing a model that is very optimistic which is unviable for practical use. It was noticed that the training and validation accuracy would keep improving and always ended up with accuracies of at least 95% during the 5th iteration, which seems too good to be true. It was suspected that the problem lies between the data preparation steps. Efforts were put in to troubleshoot the algorithm, but due to limitation of expertise and restricted timeframe, the problem persisted, and hence cross-validation was put out of scope to direct the focus on developing the CNN models.

CHAPTER 5: CONCLUSION AND FUTURE IMPROVEMENT

5.1 Conclusion

In this study, a series of experiments have been conducted using pre-processed axial sMRI brain images retrieved from the ADNI database with different deep learning CNN architectures. To classify brain sMRI images of 3 distinct classes of AD, CN, and MCI, 3 different CNN models were built, namely a CNN from scratch, VGG-16, and ResNet-50 to address the problem. Out of the 3 models, VGG-16 achieved the best testing performance with accuracy of 78.57%, precision of 73.94%, recall of 81.37%, and F1-score of 77.48%. The results showed that despite the fact that VGG-16 was trained on very general images from the ImageNet dataset, it had the ability to extract useful features for the classification task. The pre-trained VGG-16 model obtained better performance than shallow CNN and classical machine learning algorithms using the same dataset. However, its performance is considered subpar when compared to other literature which also employed deep learning techniques. Increasing the number of data for training is the main factor for improving classification performance. The work of this project is a motivation for more expanded studies on computer-assisted AD diagnosis systems that can provide automated early diagnosis of AD and the detection of more phenotypes of AD.

5.2 Future Improvement

For future work, a list of improvements can be suggested. Effort should be put in to try other pre-trained CNN such as AlexNet, Xception, Inception, MobileNet, other variants of VGG and ResNet as well as the more recent state-of-the-art network as base model for feature extraction. In addition, classification performance could also be enhanced through fine-tuning. For example, unfreeze some layers or even half of the model for training jointly with the classifier with a very small learning rate. However,

this approach must be provided with sufficient availability of data and resources that can handle the increased computational costs (Jain et al., 2019).

Future work should also include visualisation tool to visualise filters and feature maps in CNN. The 2D filters that the model has learned can be visualised to uncover the types of features the model will detect. Moreover, we can visualise the output of the convolutional layer, which is an activation map to examine what are the features that contributed to the prediction results. For example, study applied Gradient-weighted Class Activation Mapping (Grad-CAM) to visualise what regions of the brain image are detected in the feature maps by CNN (Iizuka et al., 2019). Visualisation of captured features is of great significance to the AD research community to enable them to visualise interaction among captured features and to what extent the contribution of each feature to the final classification result. To neuroscience researchers, this information could provide insight in getting clues about the biological mechanism of AD progression which is still under extensive study.

Apart from using all slices from the brain sMRI volume as an approach to increase the amount of data, one should also try using MRI acquired using stronger magnetic flux density, for example, 3T MRI that has twice the field strength of 1.5T MRI, which has better signal-to-ratio to clearly visualise biomarkers that were invisible or a slightly ambiguous in a 1.5T MRI. A recent study demonstrated work involving evaluation of model on ADNI and Milan dataset and achieved better performance than the approach using data from a single dataset (Basaia et al., 2019). The study overcame the caveat of limited reproducibility of findings due to the usage of data from single-center dataset. This provides a strong reason for future work to examine the use of multiple datasets to train deep learning framework for further classification improvement. Examples of other databases providing MRI scans of Alzheimer's disease patients are Minimal Interval

Resonance Imaging in Alzheimer's Disease (MIRIAD) and Open Access Series of Imaging Studies (OASIS).

Last but not least, a few different methodologies for improving the classification performance to distinguish between AD, CN, and MCI could be explored in the future. One approach is to incorporate multi-modal data for the study. Recently, literature revealed extensive evidence of the benefits of multi-modal research in gaining insight into brain structure and function and decoding brain complexity. Usually, multi-model studies employ multiple types of structural modalities, demographic data, and cognitive behavioural performance measurements. Indeed, several multi-model studies demonstrated improved prediction performance in comparison to single modality study while studying classification of AD (Liu et al., 2018; Zhang et al., 2019). Multi-modal study requires feature fusion to combine features of different modalities in order to obtain a single feature vector. Another approach to enhance performance is to enrich the feature learning process by fusing low-dimensional features such as clinical scores with the MRI features space. For instance, study employed 4 different data modalities such as demographic data, cognitive performance scores, CSF biomarker measurements, entorhinal cortical thickness, and hippocampus volume (Lee et al., 2019). Likewise, there are other approaches which focused on other high-dimensional features extracted from sMRI, fMRI, electronic health records, genetics information (Bi et al., 2019; Venugopalan et al., 2021).

REFERENCES

Abrol, A., Bhattarai, M., Fedorov, A., Du, Y., Plis, S., & Calhoun, V. (2020). Deep residual learning for neuroimaging: An application to predict progression to Alzheimer's disease. *Journal of Neuroscience Methods*, *339*, 108701. https://doi.org/https://doi.org/10.1016/j.jneumeth.2020.108701

Aderghal, K., Afdel, K., Benois-Pineau, J., & Catheline, G. (2020). Improving Alzheimer's stage categorization with Convolutional Neural Network using transfer learning and different magnetic resonance imaging modalities. *Heliyon*, *6*(12), e05652. https://doi.org/https://doi.org/10.1016/j.heliyon.2020.e05652

Alzheimer's Association. (2020). 2020 Alzheimer's disease facts and figures [https://doi.org/10.1002/alz.12068]. *Alzheimer's & Dementia*, *16*(3), 391-460. https://doi.org/https://doi.org/10.1002/alz.12068

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., & Chen, G. (2016). Deep speech 2: End-to-end speech recognition in english and mandarin. International conference on machine learning,

Arevalo-Rodriguez, I., Smailagic, N., Roqué i Figuls, M., Ciapponi, A., Sanchez‐Perez, E., Giannakou, A., Pedraza, O. L., Bonfill Cosp, X., & Cullum, S. (2015). Mini‐Mental State Examination (MMSE) for the detection of Alzheimer's disease and other dementias in people with mild cognitive impairment (MCI). *Cochrane Database of Systematic Reviews*(3). https://doi.org/10.1002/14651858.CD010783.pub2

Asl, E. H., Ghazal, M., Mahmoud, A., Aslantas, A., Shalaby, A., Casanova, M., Barnes, G., Gimel'farb, G., Keynton, R., & Baz, A. E. (2018). Alzheimer's disease diagnostics by a 3D deeply supervised adaptable convolutional network. *23*(3), 584-596. https://doi.org/10.2741/4606

Basaia, S., Agosta, F., Wagner, L., Canu, E., Magnani, G., Santangelo, R., & Filippi, M. (2019). Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks. *NeuroImage: Clinical*, *21*, 101645. https://doi.org/https://doi.org/10.1016/j.nicl.2018.101645

Basheera, S., & Sai Ram, M. S. (2020). A novel CNN based Alzheimer's disease classification using hybrid enhanced ICA segmented gray matter of MRI. *Computerized Medical Imaging and Graphics*, *81*, 101713. https://doi.org/https://doi.org/10.1016/j.compmedimag.2020.101713

Battista, P., Salvatore, C., Berlingeri, M., Cerasa, A., & Castiglioni, I. (2020). Artificial intelligence and neuropsychological measures: The case of Alzheimer's disease. *Neuroscience & Biobehavioral Reviews*, *114*, 211-228. https://doi.org/https://doi.org/10.1016/j.neubiorev.2020.04.026

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *13*(null %J J. Mach. Learn. Res.), 281–305.

Bi, X.-a., Cai, R., Wang, Y., & Liu, Y. (2019). Effective Diagnosis of Alzheimer's Disease via Multimodal Fusion Analysis Framework [Original Research]. *10*(976). https://doi.org/10.3389/fgene.2019.00976

Bi, X., Li, S., Xiao, B., Li, Y., Wang, G., & Ma, X. (2020). Computer aided Alzheimer's disease diagnosis by an unsupervised deep learning technology. *Neurocomputing*, *392*, 296-304. https://doi.org/https://doi.org/10.1016/j.neucom.2018.11.111

Brookmeyer, R., Johnson, E., Ziegler-Graham, K., & Arrighi, H. M. (2007). Forecasting the global burden of Alzheimer's disease [https://doi.org/10.1016/j.jalz.2007.04.381]. *Alzheimer's & Dementia*, *3*(3), 186-191. https://doi.org/https://doi.org/10.1016/j.jalz.2007.04.381

Buckner, R. L. (2004). Memory and executive function in aging and AD: multiple factors that cause decline and reserve factors that compensate. *Neuron*, *44*(1), 195-208. https://doi.org/10.1016/j.neuron.2004.09.006

Casanova, R., Hsu, F.-C., & Mark A. Espeland, f. t. A. s. D. N. I. (2012). Classification of Structural MRI Images in Alzheimer's Disease from the Perspective of Ill-Posed Problems. *PLOS ONE*, *7*(10), e44877. https://doi.org/10.1371/journal.pone.0044877

Choi, J.-S., Lee, E., & Suk, H.-I. (2018, 2018//). Regional Abnormality Representation Learning in Structural MRI for AD/MCI Diagnosis. Machine Learning in Medical Imaging, Cham.

Chollet, F. (2016). *Batch Normalization Questions*. https://github.com/keras-team/keras/issues/1802

Cuingnet, R., Gerardin, E., Tessieras, J., Auzias, G., Lehéricy, S., Habert, M.-O., Chupin, M., Benali, H., & Colliot, O. (2011). Automatic classification of patients with Alzheimer's disease from structural MRI: A comparison of ten methods using the ADNI database. *NeuroImage*, *56*(2), 766-781. https://doi.org/https://doi.org/10.1016/j.neuroimage.2010.06.013

Dario, A., Sundaram, A., Rishita, A., Jingliang, B., Eric, B., Carl, C., Jared, C., Bryan, C., Qiang, C., Guoliang, C., Jie, C., Jingdong, C., Zhijie, C., Mike, C., Adam, C., Greg, D., Ke, D., Niandong, D., Erich, E., Jesse, E., Weiwei, F., Linxi, F., Christopher, F., Liang, G., Caixia, G., Awni, H., Tony, H., Lappi, J., Bing, J., Cai, J., Billy, J., Patrick, L., Libby, L., Junjie, L., Yang, L., Weigao, L., Xiangang, L., Dongpeng, M., Sharan, N., Andrew, N., Sherjil, O., Yiping, P., Ryan, P., Sheng, Q., Zongfeng, Q., Jonathan, R., Vinay, R., Sanjeev, S., David, S., Shubho, S., Kavya, S., Anuroop, S., Haiyuan, T., Liliang, T., Chong, W., Jidong, W., Kaifu, W., Yi, W., Zhijian, W., Zhiqian, W., Shuang, W., Likai, W., Bo, X., Wen, X., Yan, X., Dani, Y., Bin, Y., Jun, Z., & Zhenyao, Z. (2016, 2016/06/11). *Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin* Proceedings of the 33rd International Conference on Machine Learning, https://proceedings.mlr.press/v48/amodei16.html

Ebrahimighahnavieh, M. A., Luo, S., & Chiong, R. (2020). Deep learning to detect Alzheimer's disease from neuroimaging: A systematic literature review.

Computer Methods and Programs in Biomedicine, *187*, 105242. https://doi.org/https://doi.org/10.1016/j.cmpb.2019.105242

Eman, M., Seddik, A., & H, M. (2016). Automatic Detection and Classification of Alzheimer's Disease from MRI using TANNN. *International Journal of Computer Applications*, *148*, 30-34. https://doi.org/10.5120/ijca2016911320

Emrani, S., Arain, H. A., DeMarshall, C., & Nuriel, T. (2020). APOE4 is associated with cognitive and pathological heterogeneity in patients with Alzheimer's disease: a systematic review. *Alzheimer's Research & Therapy*, *12*(1), 141. https://doi.org/10.1186/s13195-020-00712-4

Folego, G., Weiler, M., Casseb, R. F., Pires, R., & Rocha, A. (2020). Alzheimer's Disease Detection Through Whole-Brain 3D-CNN MRI [Original Research]. *8*(1193). https://doi.org/10.3389/fbioe.2020.534592

Fung, G., & Stoeckel, J. (2007). SVM feature selection for classification of SPECT images of Alzheimer's disease using spatial information. *Knowledge and Information Systems*, *11*(2), 243-258. https://doi.org/10.1007/s10115-006-0043-5

Goceri, E., & Songül, C. (2017, 5-8 Oct. 2017). Automated detection and extraction of skull from MR head images: Preliminary results. 2017 International Conference on Computer Science and Engineering (UBMK),

Grundke-Iqbal, I., Iqbal, K., Tung, Y. C., Quinlan, M., Wisniewski, H. M., & Binder, L. I. (1986). Abnormal phosphorylation of the microtubule-associated protein tau (tau) in Alzheimer cytoskeletal pathology. *Proceedings of the National Academy of Sciences of the United States of America*, *83*(13), 4913-4917. https://doi.org/10.1073/pnas.83.13.4913

Grundman, M., Petersen, R. C., Ferris, S. H., Thomas, R. G., Aisen, P. S., Bennett, D. A., Foster, N. L., Jack, C. R., Jr., Galasko, D. R., Doody, R., Kaye, J., Sano, M., Mohs, R., Gauthier, S., Kim, H. T., Jin, S., Schultz, A. N., Schafer, K., Mulnard, R., van Dyck, C. H., Mintzer, J., Zamrini, E. Y., Cahn-Weiner, D., & Thal, L. J. (2004). Mild cognitive impairment can be distinguished from Alzheimer disease and normal aging for clinical trials. *Arch Neurol*, *61*(1), 59-66. https://doi.org/10.1001/archneur.61.1.59

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., & Cuadros, J. J. J. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *316*(22), 2402-2410.

Gupta, A., Ayhan, M., & Maida, A. (2013). *Natural Image Bases to Represent Neuroimaging Data* Proceedings of the 30th International Conference on Machine Learning, Proceedings of Machine Learning Research. https://proceedings.mlr.press/v28/gupta13b.html

Gupta, Y., Lee, K. H., Choi, K. Y., Lee, J. J., Kim, B. C., Kwon, G. R., the National Research Center for, D., & Alzheimer's Disease Neuroimaging, I. (2019). Early diagnosis of Alzheimer's disease using combined features from voxel-based morphometry and cortical, subcortical, and hippocampus regions of MRI T1 brain

images. *PLOS ONE*, *14*(10), e0222446. https://doi.org/10.1371/journal.pone.0222446

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition,

Hidaka, A., & Kurita, T. (2017). *Consecutive Dimensionality Reduction by Canonical Correlation Analysis for Visualization of Convolutional Neural Networks* (Vol. 2017). https://doi.org/10.5687/sss.2017.160

Hippius, H., & Neundörfer, G. (2003). The discovery of Alzheimer's disease. *Dialogues in clinical neuroscience*, *5*(1), 101-108. https://doi.org/10.31887/DCNS.2003.5.1/hhippius

Iizuka, T., Fukasawa, M., & Kameyama, M. (2019). Deep-learning-based imaging-classification identified cingulate island sign in dementia with Lewy bodies. *Sci Rep*, *9*(1), 8944. https://doi.org/10.1038/s41598-019-45415-5

Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. *Procedia Computer Science*, *132*, 679-688. https://doi.org/https://doi.org/10.1016/j.procs.2018.05.069

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: accelerating deep network training by reducing internal covariate shift* Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, Lille, France.

Iyaswamy, A., Krishnamoorthi, S. K., Song, J. X., Yang, C. B., Kaliyamoorthy, V., Zhang, H., Sreenivasmurthy, S. G., Malampati, S., Wang, Z. Y., Zhu, Z., Tong, B. C., Cheung, K. H., Lu, J. H., Durairajan, S. S. K., & Li, M. (2020). NeuroDefend, a novel Chinese medicine, attenuates amyloid-β and tau pathology in experimental Alzheimer's disease models. *J Food Drug Anal*, *28*(1), 132-146. https://doi.org/10.1016/j.jfda.2019.09.004

Jagannath, A., Jagannath, J., & Melodia, T. J. a. p. a. (2020). Redefining wireless communication for 6G: Signal processing meets deep learning.

Jain, R., Jain, N., Aggarwal, A., & Hemanth, D. J. (2019). Convolutional neural network based Alzheimer's disease classification from magnetic resonance brain images. *Cognitive Systems Research*, *57*, 147-159. https://doi.org/https://doi.org/10.1016/j.cogsys.2018.12.015

Johnson, K. A., Fox, N. C., Sperling, R. A., & Klunk, W. E. (2012). Brain imaging in Alzheimer disease. *Cold Spring Harbor perspectives in medicine*, *2*(4), a006213-a006213. https://doi.org/10.1101/cshperspect.a006213

Kingma, D. P., & Ba, J. J. a. p. a. (2014). Adam: A method for stochastic optimization.

Klöppel, S., Stonnington, C. M., Chu, C., Draganski, B., Scahill, R. I., Rohrer, J. D., Fox, N. C., Jack, C. R., Jr, Ashburner, J., & Frackowiak, R. S. J. (2008). Automatic

classification of MR scans in Alzheimer's disease. *Brain*, *131*(3), 681-689. https://doi.org/10.1093/brain/awm319 %J Brain

Kumar, L. S., Hariharasitaraman, S., Narayanasamy, K., Thinakaran, K., Mahalakshmi, J., & Pandimurugan, V. (2021). AlexNet approach for early stage Alzheimer's disease detection from MRI brain images. *Materials Today: Proceedings*. https://doi.org/https://doi.org/10.1016/j.matpr.2021.04.415

Lee, G., Nho, K., Kang, B., Sohn, K.-A., Kim, D., & Alzheimer's Disease Neuroimaging Initiative. (2019). Predicting Alzheimer's disease progression using multi-modal deep learning approach. *Sci Rep*, *9*(1), 1952. https://doi.org/10.1038/s41598-018-37769-z

Lee, K. B., Cheon, S., & Kim, C. J. I. T. o. S. M. (2017). A Convolutional Neural Network for Fault Classification and Diagnosis in Semiconductor Manufacturing Processes. *30*, 135-142.

Li, F., Cheng, D., Liu, M. J. I. I. C. o. I. S., & Techniques. (2017). Alzheimer's disease classification based on combination of multi-model convolutional networks. 1-5.

Liu, M., Cheng, D., Wang, K., & Wang, Y. (2018). Multi-Modality Cascaded Convolutional Neural Networks for Alzheimer's Disease Diagnosis. *Neuroinformatics*, *16*(3-4), 295-308. https://doi.org/10.1007/s12021-018-9370-4

Liu, M., Li, F., Yan, H., Wang, K., Ma, Y., Shen, L., & Xu, M. (2020). A multi-model deep convolutional neural network for automatic hippocampus segmentation and classification in Alzheimer's disease. *NeuroImage*, *208*, 116459. https://doi.org/https://doi.org/10.1016/j.neuroimage.2019.116459

Liu, S., Liu, S., Cai, W., Pujol, S., Kikinis, R., & Feng, D. D. (2015, 2015//). Multi-Phase Feature Representation Learning for Neurodegenerative Disease Diagnosis. Artificial Life and Computational Intelligence, Cham.

Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(2), 645-657. https://doi.org/10.1109/TGRS.2016.2612821

McKhann, G. M., Knopman, D. S., Chertkow, H., Hyman, B. T., Jack Jr, C. R., Kawas, C. H., Klunk, W. E., Koroshetz, W. J., Manly, J. J., Mayeux, R., Mohs, R. C., Morris, J. C., Rossor, M. N., Scheltens, P., Carrillo, M. C., Thies, B., Weintraub, S., & Phelps, C. H. (2011). The diagnosis of dementia due to Alzheimer's disease: Recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease [https://doi.org/10.1016/j.jalz.2011.03.005]. *Alzheimer's & Dementia*, *7*(3), 263-269. https://doi.org/https://doi.org/10.1016/j.jalz.2011.03.005

Mehmood, A., Yang, S., Feng, Z., Wang, M., Ahmad, A. L. S., Khan, R., Maqsood, M., & Yaqub, M. (2021). A Transfer Learning Approach for Early Diagnosis of Alzheimer's Disease on MRI Images. *Neuroscience*, *460*, 43-52. https://doi.org/https://doi.org/10.1016/j.neuroscience.2021.01.002

Mobed, A., & Hasanzadeh, M. (2020). Biosensing: The best alternative for conventional methods in detection of Alzheimer's disease biomarkers. *International Journal of Biological Macromolecules*, *161*, 59-71. https://doi.org/https://doi.org/10.1016/j.ijbiomac.2020.05.257

Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. ICML,

National Institute of Health. (2017). *What Is Alzheimer's Disease?* National Institute on Aging. Retrieved May 30 from https://www.nia.nih.gov/health/what-alzheimers-disease

Oghabian, M. A., Batouli, S. A. H., Norouzian, M., Ziaei, M., & Sikaroodi, H. (2010). Using functional Magnetic Resonance Imaging to differentiate between healthy aging subjects, Mild Cognitive Impairment, and Alzheimer's patients. *Journal of research in medical sciences : the official journal of Isfahan University of Medical Sciences*, *15*(2), 84-93. https://pubmed.ncbi.nlm.nih.gov/21526064

Oh, K., Chung, Y. C., Kim, K. W., Kim, W. S., & Oh, I. S. (2019). Classification and Visualization of Alzheimer's Disease using Volumetric Convolutional Neural Network and Transfer Learning. *Sci Rep*, *9*(1), 18150. https://doi.org/10.1038/s41598-019-54548-6

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition.

Payan, A., & Montana, G. (2015). Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks. *ICPRAM 2015 - 4th International Conference on Pattern Recognition Applications and Methods, Proceedings*, *2*.

Pelkmans, W., Dicks, E., Barkhof, F., Vrenken, H., Scheltens, P., van der Flier, W. M., & Tijms, B. M. (2019). Gray matter T1-w/T2-w ratios are higher in Alzheimer's disease. *Hum Brain Mapp*, *40*(13), 3900-3909. https://doi.org/10.1002/hbm.24638

Rallabandi, V. P. S., Tulpule, K., & Gattu, M. (2020). Automatic classification of cognitively normal, mild cognitive impairment and Alzheimer's disease using structural MRI analysis. *Informatics in Medicine Unlocked*, *18*, 100305. https://doi.org/https://doi.org/10.1016/j.imu.2020.100305

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211-252. https://doi.org/10.1007/s11263-015-0816-y

Saurabh, R. (2015). Interaction between the Alzheimer's peptide, beta-amyloid and lipid membrane.

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv 1409.1556*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929-1958.

Taipa, R. (2018). Neuroinflammation in early and late onset Alzheimer´s disease: a multimodal analysis.

Tan, C. C., Yu, J. T., & Tan, L. (2014). Biomarkers for preclinical Alzheimer's disease. *J Alzheimers Dis*, *42*(4), 1051-1069. https://doi.org/10.3233/jad-140843

Tatiparti, K., Sau, S., Rauf, M. A., & Iyer, A. K. (2020). Smart treatment strategies for alleviating tauopathy and neuroinflammation to improve clinical outcome in Alzheimer's disease. *Drug Discovery Today*, *25*(12), 2110-2129. https://doi.org/https://doi.org/10.1016/j.drudis.2020.09.025

Tustison, N. J., Avants, B. B., Cook, P. A., Zheng, Y., Egan, A., Yushkevich, P. A., & Gee, J. C. (2010). N4ITK: improved N3 bias correction. *IEEE transactions on medical imaging*, *29*(6), 1310-1320. https://doi.org/10.1109/TMI.2010.2046908

Venugopalan, J., Tong, L., Hassanzadeh, H. R., & Wang, M. D. (2021). Multimodal deep learning models for early detection of Alzheimer's disease stage. *Sci Rep*, *11*(1), 3254. https://doi.org/10.1038/s41598-020-74399-w

Wang, H., Shen, Y., Wang, S., Xiao, T., Deng, L., Wang, X., & Zhao, X. (2019). Ensemble of 3D densely connected convolutional network for diagnosis of mild cognitive impairment and Alzheimer's disease. *Neurocomputing*, *333*, 145-156. https://doi.org/https://doi.org/10.1016/j.neucom.2018.12.018

Ward, A., Tardiff, S., Dye, C., & Arrighi, H. M. (2013). Rate of conversion from prodromal Alzheimer's disease to Alzheimer's dementia: a systematic review of the literature. *Dement Geriatr Cogn Dis Extra*, *3*(1), 320-332. https://doi.org/10.1159/000354370

Weiner, M. W., Veitch, D. P., Aisen, P. S., Beckett, L. A., Cairns, N. J., Green, R. C., Harvey, D., Jack, C. R., Jagust, W., Morris, J. C., Petersen, R. C., Saykin, A. J., Shaw, L. M., Toga, A. W., & Trojanowski, J. Q. (2017). Recent publications from the Alzheimer's Disease Neuroimaging Initiative: Reviewing progress toward improved AD clinical trials. *Alzheimer's & Dementia*, *13*(4), e1-e85. https://doi.org/https://doi.org/10.1016/j.jalz.2016.11.007

Wen, J., Thibeau-Sutre, E., Diaz-Melo, M., Samper-González, J., Routier, A., Bottani, S., Dormont, D., Durrleman, S., Burgos, N., & Colliot, O. (2020). Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation. *Medical Image Analysis*, *63*, 101694. https://doi.org/https://doi.org/10.1016/j.media.2020.101694

Zhang, F., Li, Z., Zhang, B., Du, H., Wang, B., & Zhang, X. (2019). Multi-modal deep learning model for auxiliary diagnosis of Alzheimer's disease. *Neurocomputing*, *361*, 185-195. https://doi.org/https://doi.org/10.1016/j.neucom.2019.04.093

Zhang, J., Zheng, B., Gao, A., Feng, X., Liang, D., & Long, X. (2021). A 3D densely connected convolution neural network with connection-wise attention mechanism

for Alzheimer's disease classification. *Magnetic Resonance Imaging*, *78*, 119-126. https://doi.org/https://doi.org/10.1016/j.mri.2021.02.001

Zhang, L., Wang, S., Liu, B. J. W. I. R. D. M., & Discovery, K. (2018). Deep learning for sentiment analysis: A survey. *8*(4), e1253.

Zhang, Y., Brady, M., & Smith, S. (2001). Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, *20*(1), 45-57. https://doi.org/10.1109/42.906424

Zhang, Y., Wang, S., Xia, K., Jiang, Y., & Qian, P. (2021). Alzheimer's disease multiclass diagnosis via multimodal neuroimaging embedding feature selection and fusion. *Information Fusion*, *66*, 170-183. https://doi.org/https://doi.org/10.1016/j.inffus.2020.09.002

Zhu, X., Suk, H.-I., Zhu, Y., Thung, K.-H., Wu, G., & Shen, D. (2015). Multi-view Classification for Identification of Alzheimer's Disease. *Machine learning in medical imaging. MLMI (Workshop)*, *9352*, 255-262. https://doi.org/10.1007/978-3-319-24888-2_31