

**Sino-British Collaborative Programme,**

**Oxford Brookes University & Chengdu University of Technology**

**BSc (Hons) Software Engineering Programme**

**BSc (Single Honours) Degree Project**

Programme Name: BSc (Hons) Software Engineering Programme

Module code: CHC 6096

Surname: Liu

First Name: Yue

Project Title: Ensemble learning for the classification of Alzheimer disease

Student No: 201918020206

Supervisor: Grace Ugochi. Nneji

2™ Supervisor -

(if applicable)

Date submitted: /2023

*A report submitted as part of the requirements for the degree of BSc (Hons) in Software Engineering*

*At*

*Oxford Brookes University*

**Student Conduct Regulations:**

**Ensemble learning for the classification of Alzheimer disease**

Liu Yue

201918020206

# Abstract

Alzheimer's disease is the most common type of dementia and the majority of sufferers are elderly. With the global aging process, the number of people with Alzheimer's disease is growing rapidly worldwide, and with this comes the pressure on the healthcare economy due to the increased cost of treatment. The use of magnetic resonance imaging (MRI) for the early diagnosis of Alzheimer's disease is therefore a key part of the solution to this problem. This project uses ensemble learning to combine three models which are ResNet, AlexNet, and MobileNet to avoid the potential failures of individual models and to combine the benefits of individual models. The data set of this project has a total of 33984 cross-sectional MRI images of Alzheimer's disease. The ratio of the training set, validation set, and test set is 6:2:2. The utilization of this ensemble model achieves an identification accuracy of 98.2%. The project will assist medical practitioners in applying automated systems in the identification of Alzheimer’s disease, thus saving more lives, practitioners’ time, and medical resources in medical centers.

# Keywords

Ensemble learning, ResNet, AlexNet, MobileNet, Deep learning, Alzheimer’s disease, Image recognition.

# Abbreviations

**DL** Deep Learning

**ResNet** Residual Networks

**GUI** Graphical User Interfaces

**AD** Alzheimer's disease

**MRI** Magnetic Resonance Imaging

**IDE** Integrated Development Environment

**SVM** Support Vector Machines

**AUC** Area Under Curve

**DEMNET** Dementia Network

**BAGGING** Bootstrap Aggregation

**MCI**  Mild Cognitive Impairment

**AI** Artificial Intelligence

# Glossary

**Deep learning:** Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning.

**Alzheimer’s Disease:** Alzheimer’s disease is a type of brain disorder that causes problems with memory, thinking and behavior. This is a gradually progressive condition.

**Ensemble learning:** Ensemble learning is a general meta-approach to machine learning that seeks better predictive performance by combining the predictions from multiple models.

**ResNet:** Residual Network is a specific type of neural network that was introduced in 2015 by Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun in their paper “Deep Residual Learning for Image Recognition”.

**AlexNet:** AlexNet is a classic convolutional neural network architecture. It consists of convolutions, max pooling and dense layers as the basic building blocks.

**MobileNet:** Efficient Convolutional Neural Networks for Mobile Vision Applications. MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks.

**Graphical User Interface:** Graphical user interface is a digital interface in which a user interacts with graphical components such as icons, buttons and menus.

**Deep Neural network:** By using a technique called Deep Neural Networks, which mimics the behavior of the human brain, the researchers were able to train it to be more discriminating than previous methods.

**Fine-tuned:** Deep learning requires continuous training in the deep network to update the model parameters (weights) to fit the model that can achieve the expected results.

**F1-Score:** It is an index used to measure the accuracy of the binary model in statistics. It also takes into account the accuracy rate and recall rate of the classification model. The F1 score can be regarded as a harmonic average of model accuracy and recall, with a maximum value of 1 and a minimum value of 0.

**Convolutional layer:** Each Convolutional layer in the convolutional neural network is composed of several convolutional units, and the parameters of each convolutional unit are optimized by the backpropagation algorithm. The purpose of the convolution operation is to extract different features of the input. The first convolution layer may only extract some low-level features such as edges, lines and corners, etc., and more layers of networks can iteratively extract more complex features from low-level features.

**Bagging:** Bootstrap aggregation, is the ensemble learning method that is commonly used to reduce variance within a noisy dataset.

# Acknowledgment

I would like to thank my supervisor, Dr. Grace Ugochi Nneji for her professional guidance and advice. Without her long-term support in all aspects of this project and her patience and dedication, this thesis would not have reached this level of completion. I am also grateful to the project leader, Joojo Walker for his teaching and the other staff, and to Oxford Brookes University and the Chengdu University of Technology for the valuable opportunities they have provided for me. Finally, I would like to thank the community and network for the resources provided, and my family and friends who have supported me and encouraged me. Thank you to all of you for making this project a success.

**Table of Contents**

[Abstract 4](#_Toc10270)

[Keywords 4](#_Toc19211)

[Abbreviations 5](#_Toc22924)

[Glossary 6](#_Toc23328)

[Acknowledgment 8](#_Toc22092)

[Chapter 1 Introduction 12](#_Toc3967)

[1.1. Background 12](#_Toc9326)

[1.1.1. Challenges 13](#_Toc8701)

[1.2. MRI 15](#_Toc14956)

[1.3. Deep Learning 17](#_Toc11064)

[1.4. Convolutional Neural Network 18](#_Toc8759)

[1.4.1. Input Layer 18](#_Toc14854)

[1.4.2. Convolutional Layer 18](#_Toc12646)

[1.4.3. Pooling Layer 19](#_Toc12111)

[1.4.4. Fully Connected Layer 19](#_Toc26668)

[1.4.5. Activation Function 20](#_Toc18875)

[1.4.6. Loss Function 20](#_Toc21704)

[1.4.7. Evaluation 21](#_Toc26595)

[1.4.8. 21](#_Toc1917)

[1.4.9. 21](#_Toc31965)

[1.4.10. r 21](#_Toc17248)

[1.5. Ensemble model 21](#_Toc21088)

[1.6. Aim 21](#_Toc14968)

[1.7. Objectives 21](#_Toc19667)

[1.8. Project Overview 22](#_Toc22537)

[1.8.1. Scope 22](#_Toc18689)

[1.8.2. Audience 23](#_Toc29181)

[Chapter 2 Background Review 24](#_Toc10317)

[2.1. Single CNN Model: 24](#_Toc17093)

[2.2. Dual CNN Models: 25](#_Toc1940)

[2.3. Ensemble Models 25](#_Toc22322)

[Chapter 3 Methodology 28](#_Toc15911)

[3.1. Dataset 28](#_Toc14348)

[3.1.1. Dataset preprocessing 28](#_Toc28030)

[3.1.1.1. Data Resizing 28](#_Toc5495)

[3.1.1.2. Data Normalization 28](#_Toc28560)

[3.2. Ensemble Model 28](#_Toc10852)

[3.2.1. Finetuned Residual Network (ResNet) 28](#_Toc8446)

[3.2.2. AlexNet 30](#_Toc157)

[3.2.3. Finetuned MobileNet 32](#_Toc11627)

[3.2.4. Ensemble Model 34](#_Toc16946)

[3.3. Performance Evaluation Metrics 36](#_Toc30646)

[3.4. evaluations 36](#_Toc27474)

[3.5. Technology(explain and table 36](#_Toc17653)

[3.6. Project Version Management 36](#_Toc2000)

[Chapter 4 Results 39](#_Toc18535)

[4.1. Hyperparameters setting 39](#_Toc16864)

[4.2. Evaluations of the project model 39](#_Toc24489)

[4.2.1. Acc 39](#_Toc29663)

[4.2.2. Loss 39](#_Toc2556)

[4.2.3. precision 39](#_Toc15640)

[4.2.4. Recall 39](#_Toc14945)

[4.2.5. F1 39](#_Toc23007)

[4.3. GUI 39](#_Toc25546)

[Chapter 5 Professional issue 39](#_Toc16776)

[5.1. Project Management 39](#_Toc22461)

[5.1.1. Activities 39](#_Toc24829)

[5.1.2. Schedule 40](#_Toc4886)

[5.1.3. Project Data Management 40](#_Toc16117)

[5.1.4. Project Deliverables 40](#_Toc12744)

[5.2. Risk Analysis 41](#_Toc32555)

[5.3. Professional Issues 41](#_Toc18903)

[Chapter 6 Conclusion 43](#_Toc9221)

[References 44](#_Toc11993)

# Introduction

## Background

Alzheimer's disease (AD) involves the part of the brain that controls thought, memory and language, and is a progressive neurodegenerative disease with an insidious onset, the most common type of dementia with no known cause [1]. The majority of people with Alzheimer's disease are elderly. With the global ageing process, the number of people with Alzheimer's disease is likely to increase by more than 100 million worldwide, and the increasing burden of its prevalence may exceed the capacity for manageable diagnosis and management. At the same time, AD often has a long latency and prodromal period, which will place a huge burden on patients as well as their families when they progress to an advanced stage. At the same time, the cost of treatment will increase significantly, as well as the fact that there is no curable drug for AD, which may impose a heavy burden on society in terms of medical expenses, and material resources, and seriously affect the economic and social stability development. It is therefore of utmost importance to diagnose AD early and to slow down or stop the progression of the disease through early intervention. The earliest clinical means of diagnosing AD can only rely on the patient's performance on cognitive tests, and the early symptoms of AD are so similar to the decline in memory capacity caused by normal ageing that early diagnosis is difficult to achieve. With advances in medical imaging technology and the discovery of some biological markers associated with AD, the degree of brain atrophy can be an important basis for the diagnosis of AD, and Magnetic Resonance Imaging (MRI) technology can accurately capture changes in the volume of brain areas caused by brain atrophy, thus helping in the diagnosis of AD [2]. Moreover, MRI has the advantages of high soft tissue resolution and the ability to perform multi plane and multi sequence imaging. The use of deep learning and other methods to analyse MRI images has been widely studied and will become one of the breakthroughs in the early diagnosis of AD, particularly Convolutional Neural Networks (CNNs) are capable of learning and extracting representative features from raw input images, have shown promise in the accurate identification of Alzheimer's disease based on MRI images.[2]. A sample of AD image is shown in Figure 1

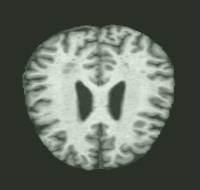


Figure 1: A sample of Alzheimer’s image

### Challenges

* Category imbalance:

It's possible that there are more examples of one type of Alzheimer's disease than the others in the distribution of the disease in medical photos, which will result in biased model training. Particularly, there are comparatively few research on the applicability of computer-aided diagnosis of AD in diagnosing various progressive stages of AD, with the majority of the literature focusing more on binary categorization or diagnosis of AD and MCI phases. This also includes dataset bias, which can happen when particular patient groups are over- or under-represented. This may result in deceptive ensemble learning models that are biased towards particular patient populations, which could limit the model's ability to be generalized to other cohorts.

* Feature extraction:

For accurate classification, it is crucial to extract useful characteristics from medical images, but this can be difficult given the complexity and high dimensionality of medical images. Brain MRI pictures, for instance, may be vulnerable to image quality fluctuation due to variances in acquisition techniques, hardware, and software utilized by various centers, which may result in variations in image resolution, contrast, and noise. The classification and detection of Alzheimer's disease may be impacted by this variability in the accuracy and dependability of the retrieved data.

* Data pre-processing:

Pre-processing is an important step in preparing image data for Alzheimer's disease diagnosis using ensemble learning algorithms. However, separate pre-processing techniques such as segmentation, normalisation or denoising may affect the accuracy of the neural network in different ways. Data acquisition and pre-processing can introduce biases in the image dataset that directly affect the accuracy and consistency of the extracted features. Bias can be introduced through variations in the acquisition process, differences in imaging scanners and other automated image processing techniques.

* Overfitting:

Overfitting happens when a machine learning model learns noise in the training data rather than the underlying patterns or characteristics crucial to the classification. This occurs when the model is too sophisticated or the training data set is too little in comparison to the number of features. Overfitting may occur in the context of integration learning when the individual models in the integration are too complex or the integration is too large relative to the training dataset, or when different models learn from the same features, resulting in related predictions, which may also increase the likelihood of overfitting. The bias-variance trade-off is one of the key contributors to overfitting in integration learning. The challenge of achieving the ideal balance between underfitting and overfitting in a machine learning model is known as the bias-variance trade-off. Data sources and imaging methods are not standardized, which negatively affects the ability to recognize images of Alzheimer's disease. This could cause noise, and noise might cause issues with ensemble learning. Underfitting occurs when the model is too simple and fails to capture the underlying patterns in the data, while overfitting occurs when the model is too complex and fits the noise in the data.

* Hyper-parameter tuning:

Ensemble learning algorithms contain multiple models with different hyper-parameters, which makes hyper-parameter tuning a challenge. Setting optimal values for hyperparameters such as learning rate, dropout rate or batch size is an iterative process that requires care to avoid over-fitting.

* Computational complexity:

Integration models require significant computational power to produce results, which may be limited by available computational resources.

* Integration complexity:

Choosing the right base model for an ensemble can be a challenge. Different types of models may perform better for different types of Alzheimer's disease images, so selecting the most appropriate model requires domain expertise and careful evaluation. And the models used in the integration may differ in accuracy or generalisation behaviour, making the performance of the integration difficult to control. As the number of underlying models in the integration increases, the complexity of the overall system increases. Ensuring that the ensemble is properly optimised and calibrated can be challenging.

* Training time:

Ensemble learning typically requires a longer training time than individual models, as multiple models need to be trained and combined. This is particularly challenging for large and complex datasets, such as those related to Alzheimer's disease.

## MRI

Imaging detection is now a practical, affordable, reliable, and accurate diagnostic technique. When AD occurs, it may be identified by looking for certain observable structural abnormalities on medical images, such as brain atrophy, white matter degeneration, widening and depth of sulci, narrowing of gyrus, enlargement of ventricles, thinning of cerebral cortex, and so on. Among these, MRI technology has made several reasonably well-liked technological advancements, including functional MRI (fMRI) and PET-MRI. The functional MRI can identify changes in neural connections and functional activity in particular brain areas. The AD protein marker amyloid (A β) plaques and tau protein deposition, another crucial protein implicated in the onset of AD, may both be seen with PET-MRI imaging. Due to its higher soft tissue contrast, multiplanar imaging capability, and lack of ionizing radiation compared to other imaging techniques, magnetic resonance imaging (MRI) has gained widespread acceptance as a non-invasive imaging modality for detecting AD. Table 1.2 provides detailed information on different imaging technologies. [4-7]

With the continuous development of science and technology, various neuroimaging data are also being generated in large quantities. Other imaging methods, such as computed tomography (CT) and PET, are also used for the diagnosis of Alzheimer's disease, but there are some limitations. Due to the use of ionizing radiation, the soft tissue contrast of CT is inferior to that of MRI, and therefore cannot be used for functional imaging. On the other hand, PET imaging provides functional and molecular data with high sensitivity, but is limited by radiation exposure and costly. Compared to these modes, MRI has fewer limitations and is a multifunctional imaging mode that can provide multiple anatomical and functional information. Figure 2 shows the ratio of MRI images in deep learning mode medical images. [8]

|  |  |  |
| --- | --- | --- |
| Categories | Mechanism | Diagnostic features |
| MRI | High-resolution structural images of the brain reflect abnormal morphological changes in brain tissue. | Cortical thinning, gray matter loss, and hippocampal atrophy in specific anatomical regions (inner and outer frontal parietal lobes and posterior cingulate cortex) of MCI/AD patients. |
| fMRI | Detect brain activity by detecting changes in oxygen consumption of brain nerve cells under resting or external stimuli. | Under task stimuli related to memory, emotion, and cognition, there was a significant decrease in the activation status of the medial frontal and parietal lobes in MCI/AD patients; Under resting state, MCI/AD patients exhibit abnormal activity in brain regions related to memory, executive function, and attention, such as the hippocampus, parahippocampal gyrus, and left inferior frontal gyrus, as well as in default network core brain regions such as the posterior cingulate gyrus, anterior cuneiform lobe, and left thalamus. |
| DTI | The Brownian motion of water molecules is traced to probe the directional condition of nerve fibres in the white matter of the brain, to detect the degree of freedom and diffusion of water molecules in the direction of diffusion, and to assess the integrity of the white matter of the brain and the functional state of the brain. | Impaired integrity of hippocampus, frontotemporal parietal lobe, cingulate gyrus, geniculate corpus callosum and hooked bundle in MCI/AD patients with abnormal diffusion of water molecules. |
| PET | Fluorodeoxyglucose PET (FDG-PET) was used to detect changes in glucose metabolism. | Decreased FDG uptake in the temporoparietal cortex in patients with cognitive impairment. Metabolic abnormalities in brain regions can be detected about 10 years earlier. |

Table 1.2 Mechanisms of Various Neuroimaging and AD Diagnostic Features

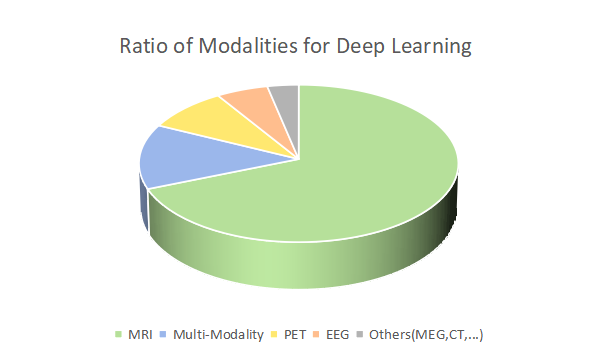


Figure 2 Ratio of Modalities for Deep Learning

## Deep Learning

Deep learning is a subfield of artificial intelligence (AI) and a new research direction in the field of machine learning. In recent years, breakthroughs have been made in various applications such as speech recognition and computer vision. The goal is to enable computers to simulate human learning methods, establish models to simulate the neural connection structure of the human brain, and describe data features through multiple transformation stages when processing signals such as images, sound, and text. The biggest difference between deep learning and traditional machine learning is that it does not require human design of features and rules, but allows computers to learn and extract features themselves, and achieve efficient classification and prediction through multi-layer neural networks. The core of deep learning is neural networks. A neural network is a computational model composed of multiple nodes, where each node can accept some inputs and use a nonlinear function to weight, linearly combine, and activate these inputs, which are then output to the next node or the final output layer. In deep learning, "depth" refers to the number of layers in a neural network. Each additional layer can extract higher-level features from the data, further improving the accuracy of classification and prediction. Deep learning requires a large amount of data to train the model, so that the model can learn more accurate features and patterns. Some classic models in deep learning include Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Generative Adversarial Network (GAN). [9-11]

## Convolutional Neural Network

Convolutional Neural Network (CNN) is one of the most widely used neural networks in deep learning. CNN takes image processing as its main research direction. It can learn image features from a large amount of data through a series of operations such as convolution, non-linear activation and pooling, and has achieved great success in image classification, object detection, semantic segmentation and other fields. The CNN contains many convolution and pooling layers. Each convolution layer has multiple convolution cores, also known as filters or feature detectors. Full connectivity and normalization layers are often used in the structure of CNNs.[9,11,12]



### Input Layer

The Input layer is the first layer of the neural network and is where the data enters the network. This layer's primary job is to collect input data from the outside world, convert it into a format that the network can use, and then transfer it to the following layer for processing. The input data is frequently shown as a matrix or vector. Each node in the input layer corresponds to a characteristic of the input data, and the quantity of nodes depends on how dimensional the input data are. In a CNN processing an image, the input layer generally represents a matrix of pixels of a picture. A picture can be represented by a three-dimensional matrix. The length and width of the 3D matrix represent the size of the image, while the depth of the 3D matrix represents the colour channel of the image. For example, a black and white image has a depth of 1, while in RGB colour mode the image has a depth of 3. [10,13]

### Convolutional Layer

The core of a convolutional neural network is the convolutional layer, and the core part of the convolutional layer is the convolutional operation. Each convolution kernel performs a convolution operation on the input of the previous layer and outputs the corresponding feature map. The convolution kernel is equivalent to a learnable filter that extracts local features of the graph over different image regions at different locations and sizes. The operation of inner product (multiplying and summing element by element) of an image (different data windows of data) and a filter matrix (a fixed set of weights: since the multiple weights of each neuron are fixed, it can again be seen as a constant filterfilter) is the convolution operation.

In this operation, the following parameters are present:

* Depth: the number of neurons, which determines the thickness of the depth of the output. It also represents the number of filters.
* Step stride: determines how many steps to slide to the edge.
* Zero-padding: the outer edge is supplemented by a number of circles of zeros to facilitate sliding from the initial position to the end position in steps.

CNN uses convolutional operations to properly capture local feature information of an image at multiple scales, enhancing the accuracy of image classification and detection.

### Pooling Layer

In addition to the convolutional layer, the pooling layer in the CNN also has an important role to play. Pooling mainly performs a downsampling operation, i.e. the size of the image feature map is reduced in order to reduce the computational effort and number of parameters of the subsequent classifier. The pooling operation takes the overall statistical features of the neighbouring regions of the input matrix as the output of that position, mainly Average Pooling, Max Pooling, etc. Max Pooling selects the maximum feature value in each sub-region of the feature map as the pooling result for that sub-region, while Average Pooling calculates the average of the feature values in each sub-region. By pooling, noise and irrelevant information in the feature map can be effectively reduced and, more importantly, pooling preserves the translation invariance of the image. Hyperparameters of the pooling layer: pooling window and pooling step size. The pooling operation can also be thought of as a convolution operation. [10,13,14]

### Fully Connected Layer

The final classification outcome is often provided by 1 to 2 fully connected layers at the conclusion of the CNN after numerous rounds of convolutional layer and pooling layer processing. The most fundamental layer in neural networks is known as the Fully linked Layer, sometimes referred to as the Dense Layer since every node is linked to every other node in the preceding layer. A fully connected layer's primary job is to output the input to the following layer by performing linear transformation and nonlinear activation on the characteristics of the previous layer's input.

To improve the performance of the model, many completely connected layers can even be layered. Different sizes of fully connected layers can be selected in model design depending on the task's needs. The utilization of the whole connection layer is not without issue, though. It is subject to overfitting and needs extensive parameter training. The whole connection layer also consumes a significant amount of computational power and storage space when processing huge amounts of picture data. [9,10,15]

### Activation Function

Activation functions are an integral part of Convolutional Neural Networks (CNNs) and are used to introduce non-linearity to the output of neural network models. To put it simply, activation functions control a neuron's output by converting its input signal to an output signal. In essence, it is a mathematical function that adjusts a neuron's output to establish a threshold for activation. The importance of activation functions in CNNs is shown by the fact that without them, a neural network would be unable to recognize and learn from linearly inseparable patterns in the input data. There are several activation functions available for use in CNNs, each with its own unique characteristics, strengths, and weaknesses. Some of the commonly used activation functions are Sigmoid, Tanh, ReLU, Leaky ReLU, and Softmax. For example, Sigmoid and Tanh have smooth curves that make them suitable for binary classification tasks, while ReLU and its variants are better suited for deep learning architectures due to their computational efficiency. [10.16]

### Loss Function

A essential part of a Convolutional Neural Network (CNN), the Loss Function updates model parameters to reduce prediction error, which is a critical step in the optimization process. Simply described, it is a mathematical function that calculates the discrepancy between output that was expected and output that was actually produced. The aim of the Loss Function is to find the optimal set of weights that can help the CNN model make accurate predictions. Some of the common Loss Functions are: Mean Squared Error (MSE), Binary Cross-Entropy, Categorical ross-Entropy, Hinge Loss. [10.17]

### Evaluation

### 

### 

### r

## Ensemble model

## Aim

The project aims to take advantage of ensemble model for the efficient classification of Alzheimer disease, so as to take early measures to reduce mortality and reduce cost of medical resources.

## Objectives

1. Get to know what ensemble learning is.
2. Researches on more articles related AD
3. Analyzing the performance of each single model
4. Analyzing the performance of the ensemble model.
5. To preprocess the MRI dataset and prepare the data for training.
6. To train three CNN models (ResNet, AlexNet, and MobileNet) on the preprocessed data.
7. Model tweaking and fitting.
8. To develop an ensemble model by combining the predictions of the individual models.
9. Evaluating the performance of the model by using different performance evaluation metrics such as accuracy, recall, precision, F1-score, ROC curve.
10. To evaluate the performance of the individual models and compare with the ensemble model.
11. To develop a graphical user interface that can be used to identify Alzheimer's disease from MRI images.
12. Final presentation of this project to the targeted audience.

## Project Overview

### Scope

Magnetic resonance imaging is an important biological tissue imaging technique. Compared with other imaging techniques, MRI imaging has many advantages in terms of non-invasiveness and high spatial resolution, and is now widely used in medical imaging, especially for exploring complex and fine brain structures and functions. As a result, an increasing number of researchers are using deep learning methods to analyse magnetic resonance imaging (MRI) for the early diagnosis of brain diseases when no symptoms are apparent in the early stages. It is well known that Convolutional Neural Networks (CNN)'s fast feature extraction capability makes its pattern recognition in image data analysis very effective. In addition, it has high accuracy in image classification, such as medical imaging [3]. But CNN model application could face one or more challenges which could be overfitting, be underfitting, the gradient disappears, etc. Thus, the purpose of this project is to take the merit of single channel CNN models and concatenate them for a robust performance which is refer to as Ensemble learning for the classification of Alzheimer disease.

The significant of this study include:

* Enables early screening, identification and disease risk warning of patients at high risk of Alzheimer's disease in the elderly population.
* Helps researchers to conduct effective research into Alzheimer's disease, leading to a better understanding of the disease's pathology and progression.
* Contributes to the development of novel drugs and measures to slow the progression of the disease.
* Inform the identification of the early stages of AD and the slowing of AD onset.
* Reduces mortality in people with Alzheimer's disease.
* Reduces the probability of human-induced misdiagnosis or missed diagnosis.
* Helps to reduce the workload of doctors and improve efficiency.
* May provide a method as well as a reference for further research into early diagnosis and prevention techniques for related brain diseases.

### Audience

* Alzheimer's patients and their families.
* Medical professionals, such as neurologists and radiologists.
* Hospital.
* Medical Imaging Researcher.
* Medical magnetic resonance imaging manufacturer.
* Researchers and scholars interested in using deep learning techniques for medical diagnosis and image recognition.

# Background Review

Deep neural network has been utilized for analyzing and predicting medical images for physician and radiologists during diagnostics decision like in the case of alzheimer’s disease using single CNN model and ensemble models.

## Single CNN Model:

Deep learning techniques were applied throughout the investigation to measure brain activity images and MRI diagnoses to avoid Alzheimer's disease, ultimately achieving a classification accuracy of 92.85% in a single CNN model like AlexNet. (Yildirim et. al., 2020) study classification of alzheimer's disease MRI images with cnn based hybrid method. It was tried to determine at which stage the disease is or whether it is Alzheimer using brain images. An Alzheimer’s stage detection system is proposed based on deep features using a pre-trained AlexNet model, by transferring the initial layers from pre-trained AlexNet model and extract the deep features from the Convolutional Neural Network (CNN) (Nawaz et. al., 2020). Early diagnosis of the disease, by detection of the preliminary stage, called Mild Cognitive Impairment (MCI), remains a challenging issue.

In this respect (Miled et. al., 2020) introduce, a powerful classification architecture that implements the pre-trained network AlexNet to automatically extract the most prominent features from Magnetic Resonance Imaging (MRI) images in order to detect the Alzheimer's disease at the MCI stage. (Kazemi et. al., 2018) study a deep learning pipeline to classify different stages of alzheimer's disease from fmri data. Convolutional neuronal network architecture AlexNet was applied to fMRI datasets to classify different stages of the disease. (Lee et. al., 2019) propose a novel framework for structural magnetic resonance image (sMRI) classification of Alzheimer’s disease (AD) with data combination, outlier removal, and entropy-based data selection using AlexNet. In Lu et al’s research, when the number of iterations reaches 200, the accuracy of VGG 16 converges to 93% on the training set, and the accuracy of MobileNet converges to 98%, proving that the MobileNet network model is superior to the VGG 16 network model in terms of AD MRI image classification [7]. Fulton et al employed a ResNet50 model in diagnosing three classes and achieved 98.99% [4].

Since the convolutional layer of the general convolutional neural network (CNN) cannot satisfactorily extract long-distance correlation in the feature space, a deep residual network (ResNet) model, based on spatial transformer networks (STN) and the non-local attention mechanism, is proposed for the early diagnosis of AD (Sun et. al., 2021). (Li et. al., 2022) propose two improved ResNet algorithms that introduced the Contextual Transformer (CoT) module, group convolution, and Channel Shuffle mechanism into the traditional ResNet residual blocks.

## Dual CNN Models:

(Fedorov et. al., 2019) investigate the use of variants of DIM in a setting of progression to Alzheimer's disease in comparison with supervised AlexNet and ResNet inspired convolutional neural networks. (Al-Adhaileh, 2021) study diagnosis and classification of alzheimer's disease by using a convolution neural network algorithm. Two deep neural network techniques, AlexNet and Restnet50, were applied for the classification and recognition of AD.

## Ensemble Models

(Acharya et. al., 2021) aim to classify MRI of Alzheimer disease patients into multiple class by using VGG16, ResNet -50 and AlexNet as transfer learning models along with convolution neural networks. A summary of the researches based on AD can be seen in Table1

Table 1: Summarize the other related work for Alzheimer disease classification.

|  |  |  |  |
| --- | --- | --- | --- |
| Authors | CNN Type | Proposed Model/Technique | Performance Metrics |
| Alanazi et al [6] | Single/Dual /Ensemble | AlexNet+SVM hybrid models  ResNet-50+SVM hybrid models | Accuracy = 94.8%  Accuracy = 93.3% |
| Plocharski et al [8] |  | Model for extracting medical features using a medical superficial sulcal pattern | Accuracy = 87.9%  Sensitivity = 90%  Specificity = 86.7% |
|  | Elakkiya et al [9] | DEMentia NETwork | Accuracy = 95.23% |
|  | Rehman et al [10] | The proposed hybrid classical quantum network+ResnNet34 | Accuracy = 97.2% |

Alanazi et al compared the hybrid algorithm between machine learning and deep learning in deep learning, and found that the accuracy, sensitivity, specificity and AUC values of the AlexNet+SVM model were 94.8%, 93%, 97.75% and 99.7% respectively, which was better [6].

Plocharski et al developed an algorithm to calculate the inner surface of the groove for feature extraction, and based on this, proposed a model to distinguish AD patients from ordinary people, the accuracy, sensitivity and specificity of the model were 87.9%, 90% and 86.7% [8].

Elakkiya et al used the DEMNET (Dementia Network) model based on CNN to detect the four different stages of AD, and used the SMOTE method for data enhancement. The results of the model reached 95.23% of the test accuracy [9].

Rehman et al proposed hybrid classical quantum network for automatic detection of AD and ResnNet34 for feature extraction, achieving the highest test accuracy of 97.2% [10].

Ensemble learning：

(Ezzati et. al., 2019) study optimizing machine learning methods to improve predictive models of alzheimer's disease. Identifying optimal features or algorithms is still a challenge. (An et. al., 2020) present a deep ensemble learning framework that aims to harness deep learning algorithms to integrate multisource data and tap the 'wisdom of experts'. (Lopez-Martin et. al., 2020) present a deep learning model to detect early symptoms of Alzheimer's disease using synchronization measures obtained with magnetoencephalography. (Nanni et. al., 2020) evaluate the potential of ensemble transfer-learning techniques, pretrained on generic images and then transferred to structural brain MRI, for the early diagnosis and prognosis of AD, with respect to a fusion of conventional-ML approaches based on Support Vector Machine directly applied to structural brain MRI. (Kang et. al., 2021) propose an ensemble learning (EL) architecture based on 2D CNNs, using a multi-model and multi-slice ensemble. (Zhang et. al., 2021) study diagnosis of alzheimer's disease with ensemble learning classifier and 3d convolutional neural network. Based on MRI data, a method combining a 3D convolutional neural network and ensemble learning is proposed to improve the diagnosis accuracy. There are three main objectives: i) to present a fully automated deep-ensemble approach for dementia-level classification from brain images, ii) to compare different deep learning architectures to obtain the most suitable one for the task, and (iii) evaluate the robustness of the proposed strategy in a deep learning framework to detect Alzheimer's disease and recognise different levels of dementia (Loddo et. al., 2021). (Li et. al., 2022) study ensemble of convolutional neural networks and multilayer perceptron for the diagnosis of mild cognitive impairment and alzheimer's disease. To capture the anatomical changes in the brain caused by AD/MCI, deep learning-based MRI image analysis methods have been proposed in recent years. An integrative mulitresolutional ensemble deep learning-based framework is proposed to achieve better predictive performance for the diagnosis of Alzheimer disease (Razzak et. al., 2022). (Zhang et. al., 2022) present a novel tensor multi-task learning (MTL) algorithm based on similarity measurement of spatio-temporal variability of brain biomarkers to model AD progression.

# Methodology

This section will explain the approach carried out which include, the dataset and its preprocessing, the model and the performance evaluation metrics.

## Dataset

This dataset is gotten from Kaggle dataset and it contains 33984 cross-sectional MRI images of the brain with Alzheimer's disease. 8960 MRI images contains mild dementia, 6464 MRI images is moderate dementia, 9600 MRI images is non-dementia, and 8960 MRI images is the very mild dementia. The split ratio for the training set, validation set, and test set is 6:2:2.

### Dataset preprocessing

* In this phase, data preprocessing was performed on the data so it could fit for the model. The data preprocessing include data resizing and normalization. Data processing technology

First, data reading is required. Since the pixels of each image are different, you need to use the cv library to reset all image sizes to a uniform size. Second, in order to make training more stable, you can normalize the image size to between 0 and 1, that is, divide all data by 255.

### Data Resizing

### Data Normalization

## Ensemble Model

### Finetuned Residual Network (ResNet)

The residual network(ResNet) is constructed from Residual Building Blocks, it does not increase the complexity of the network while increasing the depth of the network, and the effect is far better than other networks such as VGG and Google Net. With the increase of the number of floors, this advantage becomes more and more obvious. The quick connection of ResNet makes the network easier to optimize. The internal residual block uses a skip connection, which alleviates the problem of gradient disappearance caused by increasing depth in the deep neural network. This is a layer of convolution, a residual block containing convolution and pooling, then a layer of convolution is added, a layer of tiling is added, a layer of full joins is added, then a layer of dropout, and then the output layer.

The first step in fine-tuning ResNet is to load a dataset from a specified path, which contains brain scans of patients with varying degrees of dementia. Then adjust the size of the dataset to a fixed size, in this case 64 x 64 pixels, and normalize the pixel values by dividing each pixel value by 255. This normalization process helps to improve the performance of the model by ensuring that the input values are within a certain range.

Next, divide the dataset into a training set and a validation set. The training set is used to impart patterns in the data to the model, while the validation set is used to evaluate the model's performance on new, invisible data. The dataset is disrupted to ensure that the model does not learn any specific order in the data.

Then use a combination of convolutional layers and dense layers to define the ResNet architecture. The input of the model is a 64 x 64 x 3 image, where the last dimension corresponds to the RGB channel of the image. The first layer has 16 filters and 3 × A convolutional layer of kernel size 3. Next is a series of residual blocks responsible for capturing more complex features in the data. Each residual block consists of two convolutional layers, each with a kernel size of 3 x 3 and an optional maximum pooling layer. The residual block allows the model to learn the residual features of the input and then add them back to the original input. This helps the model learn the underlying patterns in the data while retaining the original information.

The output of the residual block passes through a series of dense layers, which are responsible for classifying the input into one of four categories - non dementia, mild dementia, moderate dementia, and VeryMild dementia. The dense layer has 512, 256 and 128 nodes respectively, followed by a discard layer, which is used to prevent overfitting by randomly discarding nodes during training.

Then use the Adam optimizer and the classification cross entropy loss function to compile the ResNet model, which is usually used for multi class classification problems. This model is trained using a training set, with a batch size of 128 and a total of 20 periods. A validation segmentation of 0.2 is used to validate the performance of the model after each epoch. The performance of this model is evaluated using accuracy metrics, which measure the percentage of correct predictions made by the model.

After training, the model is evaluated on the test set to measure its performance on invisible data. Test losses and accuracy will be printed to the console. Visualize the performance of the model using two graphs, one for training and verifying accuracy, and the other for training and verifying losses.

Finally, save the model to a file for future use. This process fine-tuned the ResNet architecture for specific tasks, including loading data, defining the ResNet architecture, compiling models, training models, evaluating models, and saving models. The Keras API in TensorFlow is used to detect dementia in brain scans.

The Finetuned ResNet Architecture used in this project is shown in Figure...

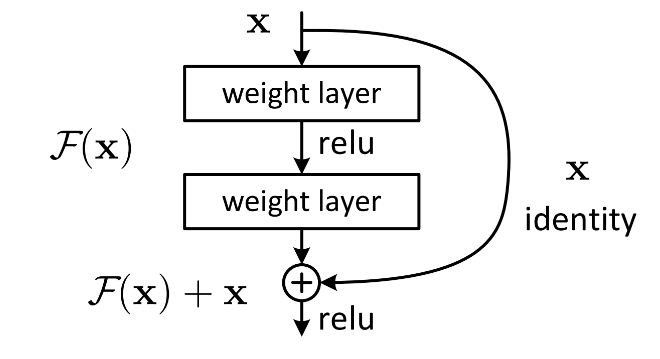
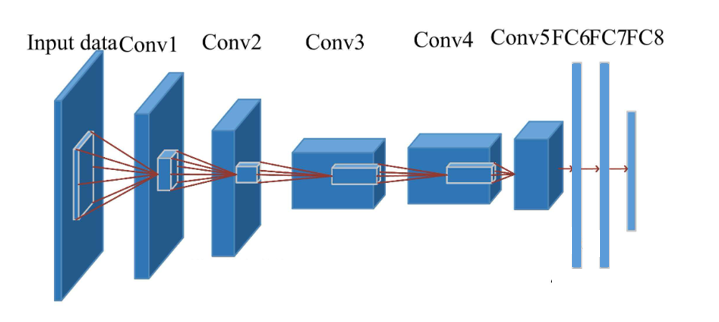


Figure 1: Finetuned ResNet Architecture

### AlexNet

AlexNet has an eight layer structure. The first five layers are convolutional neural networks, and the sixth to eighth layers are traditional neural networks. It uses the ReLU activation function to prevent the gradient from disappearing and the Dropout to prevent over fitting . In this network structure, the first layer is a convolutional layer, followed by four convolutional layer, a Flatten layer, two consecutive Dense fully-connected layers, each with a Dropout layer, and the final layer is the output layer. The whole network can be seen as the input layer is operated by convolutional layers, followed by a series of fully connected layers, and finally the output layer is used to obtain the prediction result.

In the figure, x is weighted by the first layer, and then F (x)+x is obtained after the nonlinear variation of the Relu function and the second layer weighting. This is a linear stack with two layers constituting a residual learning module. The network composed of residual modules is called ResNet. The difference between ResNet and ordinary networks is the introduction of jump connections, which can help the information from the previous residual block flow unimpeded into the next residual block. The problem of gradient disappearance and degradation caused by too deep a network is avoided.



**Figure 2: AlexNet Architecture**

Firstly, import the relevant libraries. This includes TensorFlow, Numpy, OpenCV, Matplotlib, and Keras API for TensorFlow. CUDA\_ VISIBLE\_ The DEVICES environment variable is set to 0, indicating that the GPU device to be used for calculation is 0.

The next step includes loading training and testing images. The position of the original image is using train\_ Picture and test\_ The picture variable specifies. Then adjust the size of the image to 64x64x3, which is the input size required for the AlexNet model. The pixel values of each image are normalized to the range [0,1] and converted into an array. The image array and its corresponding labels are then attached to the variable train\_ img\_ final、train\_ img\_ label、test\_ img\_ Final and test\_ img\_ The corresponding array defined by Lappel.

In the case of the above code, only the training image is used to train the model. Retain test images to evaluate the accuracy of the training model.

The next step involves mixing training and testing data. Then use the to in the Keras API\_ The categorical function converts the training data into a hot coded label.

Then use the Function API of TensorFlow's Keras API to define the architecture of the AlexNet model. The Alexnet function defined in the code is an implementation of the Alexnet model.

The input shape is defined as (64, 64, 3) because the image is resized to that shape. After the input layer, there are five convolutional layers, each with a different filter size and number of filters. The rectifier linear unit (ReLU) activation function is used for each layer.

After each convolutional layer is the maximum pooling layer with a pool size of (3,3). After the last convolutional layer, there are three dense layers. The first dense layer has 512 neurons and uses ReLU activation function. The second dense layer has 256 neurons and also uses ReLU activation function. An abscission layer with 0.5 abscission rate is added after each dense layer to reduce overfitting. The final dense layer has four neurons, corresponding to four types of images.

This model uses categorical\_ Crossntry loss function and Adam optimizer. Precision indicators are also specified.

Then use the fitting function of the Keras API to train the model. Split the training data into training data and validation data using 0.2 validation segmentation. Train the model for 20 periods using a batch size of 64.

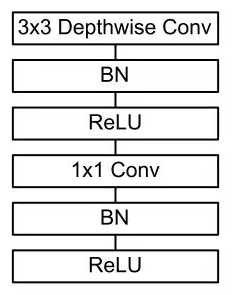
Then, the historical object returned by the fitting function is used to generate a graph of the training and validation accuracy and training and validation losses for each epoch. Printing will be saved and displayed using the Matplotlib library.

Finally, use the save function of the Keras API for TensorFlow to save the trained model to a file. The save function saves the architecture and weights of the model to a file. Can use Keras API for load\_ The model function loads the model.

The model is trained using training data and evaluated using test data. Draw the accuracy and loss of the model to help visualize its performance during training. Then save the trained model to a file for future use.

### Finetuned MobileNet

The MobileNet structure uses depthwith separable convolution to replace the standard convolution operation, and calls these two structures repeatedly to reduce the amount of model parameters and increase the amount of model calculation. Each layer is followed by a batchnorm and a ReLU nonlinear layer. Finally, the Flatten layer and the full connection layer are used to classify the images. Point convolution and deep convolution structure are the core of MobileNet, which makes MobileNet more efficient and more suitable for mobile devices. Point convolution is mainly responsible for integrating the information in the feature map, while depth convolution is responsible for extracting features.



**Figure 3: MobileNet Architecture**

The focus of this architecture is to provide a lightweight and efficient model that can run well on mobile and embedded devices with limited computing resources. In the original architecture, the author proposed using deep separable convolutions to decompose standard convolutions into deep convolutions and point by point convolutions. This significantly reduces the number of parameters and computational costs of the model.

Firstly, import the necessary modules and set environment variables to train using GPU. Then, the code defines some parameters, such as image size, random seed for data shuffling, and the class that the model needs to classify.

Then use OpenCV to load training and testing data from their respective directories. The size of the image is adjusted to a specified image size of 64x64 pixels, and normalized by dividing each pixel value by 255.0. Then convert the image into an array and attach it to the training and testing image array. In addition, corresponding labels are created based on the category of the image.

Before training the model, use random seed to mix and wash the training and test data to ensure that the data is randomly divided during training and testing. The label also uses Keras' to '\_ The categorical() function is converted to categorical data.

Then, several functions were defined to build the MobileNet architecture. depth\_ point\_ Conv2d () and pointwise\_ The conf() function is used to implement deep separable convolution, which is the main building block of MobileNet.

The mobilenet() function is used to define the mobilenet architecture. This function adopts an input tensor and applies a series of convolutions and depth separable convolution layers. This function also applies BatchNormalization and ReLU activation function after each convolution layer to stabilize training and introduce nonlinearity. Then, the application depth of this function can separate several iterations of the convolution and end with an average pooling layer, followed by several fully connected layers with ReLU activation and regularization discarding.

Once the model architecture is defined, the code uses random gradient descent (SGD) as the optimizer, the learning rate is 0.002, and the classification cross entropy is used as the loss function to compile the model. Then use the fit () function to train the model, which takes training data and labels as inputs and trains the model for a specified number of periods and batch sizes.

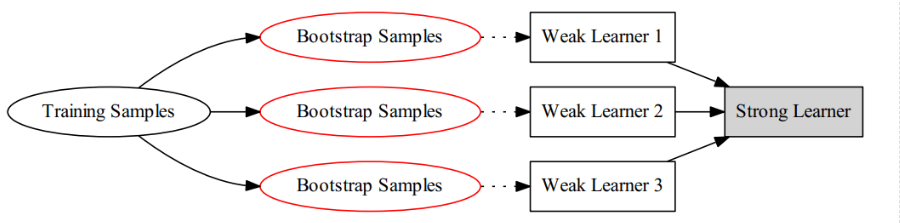
After training, the code draws the training and validation accuracy and losses for each epoch, and saves the drawing to disk. The model is also saved to disk using Keras' save() function.

To test the model, use Keras' load\_ The model() function loads the saved model from disk and uses the evaluate() function to evaluate the model on test data. Test losses and accuracy will be printed to the console.

This process uses deep separable convolution to reduce the number of model parameters and computational costs. The SGD with low learning rate and classification cross entropy loss is used to train the model, and the training data is mixed before each epoch. Finally, evaluate the model on a separate test dataset and print the test loss and accuracy to the console.

### Ensemble Model

For a given training sample S, M training samples were drawn from the training sample S in each round using Booststraping, and n rounds were conducted to obtain a total of n sets of samples. After obtaining the sample sets, one prediction model is obtained each time using one sample set, for n sample sets, a total of n prediction models can be obtained. The classification results are then obtained by applying voting to the previously obtained n models.



**Figure 4: Structure of the model**

* Algorithm

Bagging: To obtain an integration with strong generalisation performance, the ensemble learning in the integration should be as independent of each other as possible, or try to make the base learners as undifferentiated as possible. Given a training dataset, one possible approach is to sample the training samples to produce a thousand different subsets, and then train a base learner from each subset of the data. This is expected to result in a large variation in the base learners due to the different training data, however, in order to obtain better integration, it is also desirable that individual learning is not too poor, and if each subset sampled is completely different, then each learner uses only a small fraction of the training data, not even enough to learn effectively, neither of which is clearly guaranteed to produce a better base learner. Therefore, to solve this problem, consider using sampled subsets that are overlapping with each other.

* Full connection

The convolution part is mainly the operation between convolution kernel and data, and the multiplication and addition of matrix.

Activation function part, relu activation function and softmax activation function.

The pooling part, which can be said to be downsampling, effectively reduces network parameters, mainly including maximum pooling and average pooling. The basis is to calculate the maximum and average.

The fully connected part is the product of a matrix and a vector. The output is a vector. For example, the matrix of 1 \* M is multiplied by the matrix of M \* N to obtain a 1 \* N matrix.

* Back propagation

It is necessary to derive the gradient, calculate the gradient value of each training update parameter, and then update the weight and offset, that is, the addition and subtraction of matrix vectors

* Optimization strategy

Some Adam optimizers are used for backpropagation, or the size of batch training is adjusted to avoid local optimal solution

**3.3**

### Performance Evaluation Metrics

This phase will discuss the evaluation metrics used to checkmate the performance of the model which include accuracy and loss.

What is Accuracy/loss formulae

### evaluations

### Technology(explain and table

Framework: Tensorflow

IDE: Vscode

Language: Python

CUDA, cuDNN, Tensorflow-gpu，keras

Central processing Unit (CPU): Intel(R) Core (TM) i7-9750H CPU @ 2.60GHz 2.59 GHz

Graphic Processing Unit (GPU): NVIDIA GeForce GTX 1650

### Project Version Management

In the GitHub account, create the repository and complete the repository information Settings, complete the git download, and commit the code to the staging area, then commit to the local git repository, and then commit it to github.

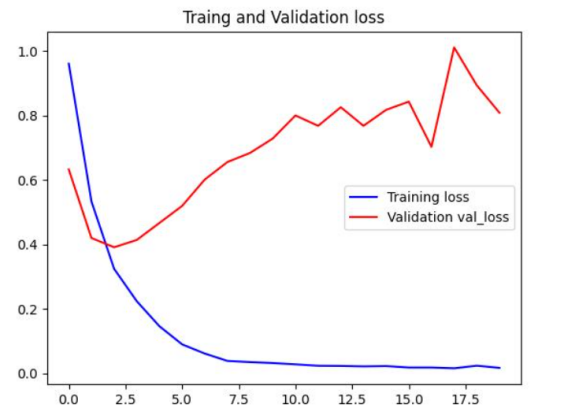
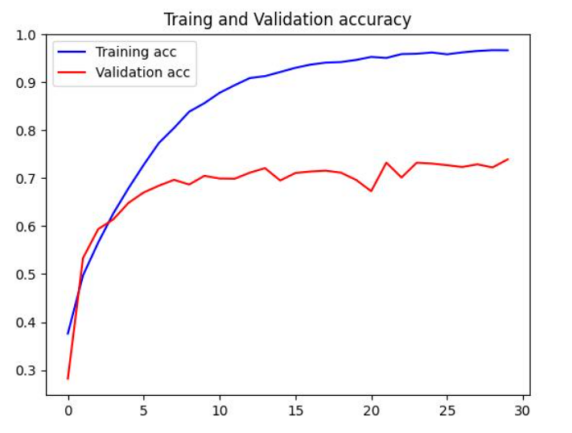
Evaluation

First, the training data is divided into training set and validation set. The validation set is not fed into the training of the neural network, but only the training set is fed into the training of the neural network and the parameters are updated. The validation set is responsible for the learning ability of the network and outputs the training results of the training set and validation set to the terminal.Then, after the training, the test set is partially input to the neural network, and the network is not trained at this time, only the data is read and output, and the prediction is verified to be correct.

|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| True | True Positive | True Negative |
| False | False Positive | False Negative |

Design and Implementation

First, three commonly used models, ResNet, AlexNet, and MobleNet, are selected based on the needs of the project. The dataset is divided into 6:2:2 and the pictures are set to a uniform size after processing with the cv library because the data needs to be read. With ResNet, the epoch in the code is set to 20, which means the total number of rounds is 20, batch\_ The size setting of 32 means that the neural network will retrieve 32 data at one time for training. Each epoch during training tests the accuracy in the training set and the verification set, and saves the model. Then write the calculation loss, reverse propagation, and optimizer, where the optimizer optimizes reverse propagation.



**Figure 5: ResNet ACC Figure 6: ResNet Loss**

It can be seen from the figure that the model may be over-fitted and still needs to be improved. The AlexNet model and MobileNet model still have problems that need to be improved. The integration algorithm is currently selected as the bagging algorithm.

# Results

* 1. Hyperparameters setting
  2. Evaluations of the project model
     1. Acc
     2. Loss
     3. precision
     4. Recall
     5. F1
  3. GUI

# Professional issue

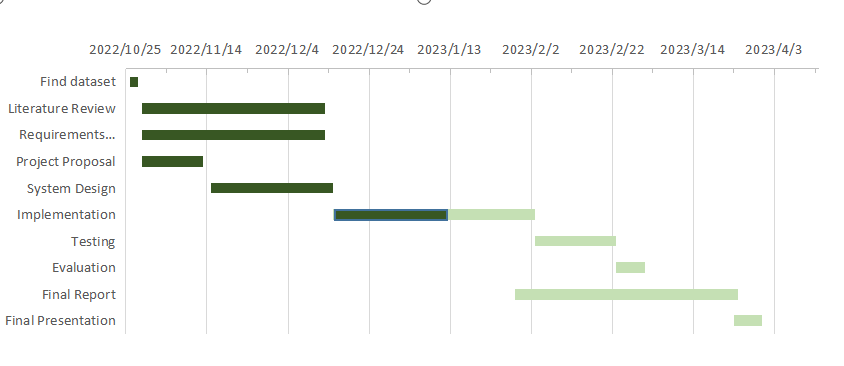
## Project Management

1. 1. 1. Activities

|  |  |  |  |
| --- | --- | --- | --- |
| Activities | Completion time | Completion status | Current progress |
| Find dataset | 1 week | 100% |  |
| Literature Review | 6 weeks | 80% | Risks and problems encountered in implementation still require more access to information to avoid or resolve. |
| Requirements Analysis | 6 weeks | 100% |  |
| Project Proposal | 2 weeks | 100% |  |
| System Design | 4 weeks | 90% | In the process of implementation, it still needs to be adjusted and improved according to the actual situation. |
| Implementation | 7 weeks | 60% | The model establishment has been completed, the training has been realized, and the implementation of bagging algorithm has been progressed. |
| Progress Report | 2 weeks | 70% | The details need to be worked out. |
| Testing | 3 weeks | 20% | Planning for testing is complete. |
| Evaluation | 1 weeks | 20% | Planning for evaluation is complete. |
| Final Report | 7 weeks | 10% |  |
| Final Presentation | 1 weeks | 10% |  |

## Schedule

Completed Uncompleted



**Figure 7: Gantt Diagram**

## Project Data Management

Use cloud folders to store data

* weekly project logs: progress, draft, next steps, supervisor comments, requirements or user stories.
* Project materials: sprint plans/reviews, testing documentation, literature etc.
* Reports: proposal, interim, final.

## Project Deliverables

* Weekly report
* The project proposal
* Image classification and diagnosis system of Alzheimer's disease (Codes)
* The progress report
* Project presentation (PPT)
* Final report - 10000 words

## Risk Analysis

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk ID** | **Potential Risk** | **Cause ID** | **Potential Causes** | **Severity** | **Likelihood** | **Risk** | **Mitigation ID** | **Mitigation** |
| R1.1 | Late in deadline | C1.1.1 | Poor time management | 3 | 3 | 9 | M1.1.1 | Do not make too detailed daily arrangements, but take the number of days as the standard, and more strictly abide by the schedule. |
| C1.1.2 | Illness | 1 | 2 | 2 | M1.1.2 | Do what I can currently do, and plan how to make up for the delayed progress |
| C1.1.3 | Technology selection | 2 | 1 | 2 | M1.1.3 | Consult more information to determine the advantages and disadvantages of technology and the needs of the project to determine the appropriate technology faster. |
| R1.2 | Operating speed | C.1.2.1 | Insufficient video memory | 1 | 3 | 3 | M1.2.1 |  |
| R1.3 | Algorithm selection | C1.3.1 | Defects of bagging algorithm | 1 | 5 | 5 | M1.3.1 | Selecting a Base Classifier with High Stability |
| R1.4 | Picture pixels | C1.4.1 | Picture pixels are 64\*64 due to device limitations, which may result in lower accuracy. | 2 | 5 | 10 | M1.4.1 |  |
| R1.5 | Model selection | C1.5.1 | MobileNet is often used on moving objects | 1 | 3 | 3 | M1.5.1 |  |

## Professional Issues

There are no copyright or plagiarism issues with this project's work. The IDE and framework of its projects and datasets are downloaded from the network, and they are open source and publicly available for research purposes. The ACM Code of Ethics and Professional Conduct provides a basis for personal responsibility and professional conduct for computer scientists who are engaged in system development that directly affects the general public.

# Conclusion

# References

1. An, N., Ding, H., Yang, J., Au, R., & Ang, T. F. A. (2020). Deep ensemble learning for Alzheimer's disease classification. Journal of Biomedical Informatics, 105, 103411. <https://doi.org/10.1016/j.jbi.2020.103411>
2. Bi, X., Liu, W., Liu, H., & Shang, Q. (2021). Artificial Intelligence-based MRI Images for Brain in Prediction of Alzheimer’s Disease. Journal of Healthcare Engineering, 2021, 1–7. https://doi.org/10.1155/2021/8198552
3. Vemuri, P., & Jack, C. R. (2010). Role of structural MRI in Alzheimer’s disease. Alzheimer’s Research & Therapy, 2(4), 23. <https://doi.org/10.1186/alzrt47>
4. Fox N. C., Schott J. M. (2004). Imaging cerebral atrophy: normal ageing to Alzheimer's disease. Lancet 363, 392–394. 10.1016/S0140-6736(04)15441-X [[Abstract](http://europepmc.org/article/MED/15074306)] [[CrossRef](https://dx.doi.org/10.1016/S0140-6736(04)15441-X" \t "https://europepmc.org/article/MED/pmc_ext)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Lancet&title=Imaging+cerebral+atrophy:+normal+ageing+to+Alzheimer's+disease&author=N.+C.+Fox&author=J.+M.+Schott&volume=363&publication_year=2004&pages=392-394&pmid=15074306&doi=10.1016/S0140-6736(04)15441-X&" \t "https://europepmc.org/article/MED/pmc_ext)]
5. Falardeau M. Respect towards people with Alzheimer's disease [J]. Soins Gerontol, 2011(91): 10-12.23.
6. Atiya M, Hyman B T, Albert M S, et al. Structural magnetic resonance imaging in established and prodromal Alzheimer disease: a review[J].Alzheimer Dis Assoc Disord, 2003, 17(3): 177-195.
7. Saima R,Mohamad H,Muhammad A I, et al. A review on neuroimaging-based classification studies and associated feature extraction methods for Alzheimer's disease and its prodromal stages[J].NeuroImage, 2017(3):57-76.
8. Yan, W., Qu, G., Hu, W., Abrol, A., Cai, B., Qiao, C., Plis, S. M., Wang, Y.-P., Sui, J., & Calhoun, V. D. (2022). Deep Learning in Neuroimaging: Promises and challenges. IEEE Signal Processing Magazine, 39(2), 87–98. <https://doi.org/10.1109/MSP.2021.3128348>
9. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
10. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
11. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.
12. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
13. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
14. Sermanet, P., LeCun, Y., & Chintala, S. (2013). Convolutional neural networks applied to house numbers digit classification. In International Conference on Pattern Recognition (pp. 3288-3291). IEEE.
15. Nielsen, M. (2015). Neural networks and deep learning: A textbook. Determination Press.
16. Aggarwal, C. C. (2018). Neural networks and deep learning. Springer.
17. Bishop, C. M. (1995). Neural networks for pattern recognition. Oxford university press.