NeuroImaging-based Dementia Risk Assessment with Pre-Trained CNN Models

by

Mostofa Shariar Sanim

ID: CSE190120355

(Bachelor of Science) in (Computer Science and Engineering)

Northern University of Business & Technology Khulna

Khulna 9100, Bangladesh

March, 2023

Acknowledgement

I would like to acknowledge and give my warmest thanks to my supervisor Ferdib-Al-Islam who made this work possible. His guidance and advice carried me through all the stages of writing my thesis. I would also like to thank my committee members for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you.

I would also like to give special thanks the Assistant professor and Head of CSE Md. Wali Ullah. He has provided us valuable suggestions, guidance, consultant and encouragement to complete this work. Also, thanks to all our faculty members for their affection and cordial co-operation.

Finally, I would like to thank God, for letting me through all the difficulties. I have experienced your guidance day by day. You are the one who let me finish my degree. I will keep on trusting you for my future.

Contents

Chapter	Title	Page
	Declaration	i
	Approval	ii
	Acknowledgement	iii
	Contents	iv
	List of Tables	vi
	List of Figures	vii
	Abstract	viii
CHAPTER I	Introduction	1 - 3
	1.1 Overview	1
	1.2 Motivation	2
	1.3 Research Question	2
	1.4 Objective of research	3
	1.5 Organization of the Thesis	3
CHAPTER II	**Literature Review**	**4 - 12**
	2.1 Introduction	4
	2.2 Data Preprocessing	4
	2.3 Normalization	5
	2.4 Data Distribution	5
	2.5 Confusion Matrix	5
	2.5.1 True Positive (TP)	6
	2.5.2 True negative (TN)	6
	2.5.3 False Positive (FP)	6
	2.5.4 False Negative (FN)	7
	2.6 Classification Report	7
	2.6.1 Accuracy	7
	2.6.2 Precision	7
	2.6.3 Recall	8
	2.6.4 F1-Score	8

Chapter	Title	Page
	2.6.5 Support	8
	2.7 Training	8
	2.8 Testing	9
	2.9 NumPy	9
	2.10 Pandas	9
	2.11 TensorFlow	9
	2.12 Keras Library	10
	2.13 Flattening	10
	2.14 Dense	10
	2.15 Convolutional Neural Network	11
	2.16 Adam Optimizer	12
	2.17 Epoch	12
CHAPTER III	**Related Works**	**12 – 13**
CHAPTER IV	**Methodology**	**14 - 20**
	4.1 Overview	14
	4.2 Method	14
	4.3 Data Collection	15
	4.4 Data preprocessing	15
	4.5 Splitting Dataset	16
	4.6 Evaluation metrics	16
	4.7 Model Training	18
	4.7.1 Inception v3	18
	4.7.2 Densenet-169	20
CHAPTER V	**Result and Discussion**	**22 - 34**
	5.1 Overview	22
	5.2 Experimental result	23
	5.2.1 Experimental result (Inception v3)	23
	5.2.1.1 Defined hyperparameters	24
	5.2.1.2 Model Summary	25

Chapter	Title5	Page
	5.2.1.3 Training model History	25
	5.2.1.4 Training Accuracy	26
	5.2.1.5 Training Loss	26
	5.2.1.6 Evaluation model	27
	5.2.1.7 Confusion Matrix	28
	5.2.2 Experimental result (Densenet-169)	28
	5.2.2.1 Defined hyperparameters	29
	5.2.2.2 Model Summary	30
	5.2.2.3 Training model History	31
	5.2.2.4 Training Accuracy	31
	5.2.2.5 Training Loss	32
	5.2.1.7 Confusion Matrix	33
	5.3 Observation	33
	5.4 Grad-Cam view	34
CHAPTER VI	**Conclusion**	**36**
	5.1 Limitations and Future Work	36
References		37

LIST OF TABLES

Table No.	Description	Page
4.1	Splitting dataset	16
5.1	Experimental result (Inception v3)	24
5.2	Hyperparameters Values (Inception v3)	24
5.3	Evaluation matrix (Inception v3)	27
5.4	Experimental result (Densenet-169)	28
5.5	Hyperparameters Values (Densenet-169)	29
5.6	Evaluation matrix (Densenet-169)	32

LIST OF FIGURES

Figure No	Description	Page
1.1	Healthy Brain vs. Severe AD Brain	1
2.1	Confusion Matrix	6
4.1	Proposed Method	14
4.2	Data Preprocessing	15
4.3	Convolutional Neural Network	18
4.4	Model training method (Inception v3)	19
4.5	Model training method (DenseNet-169)	21
4.6	DenseNet Architecture	21
5.1	Training Data	22
5.2	Testing Data	23
5.3	Model History (Inception v3)	25
5.4	Training and Validation accuracy (Inception v3)	26
5.5	Training and Validation Loss (Inception v3)	26
5.6	Confusion matrix (Inception v3)	28
5.7	Model History (Densenet-169)	31
5.8	Training Accuracy (Densenet-169)	31
5.9	Training Loss (Densenet-169)	32
5.10	Confusion matrix (Densenet-169)	33
5.11	MildDemented_Grad_view	34
5.12	ModerateDemented _Grad_view	34
5.13	NonDemented _Grad_view	34
5.15	VeryMildDemented _Grad_view	34

Abstract

Alzheimer's disease is an incurable degenerative brain disease. Every four seconds a person in the world is diagnosed with Alzheimer's disease. The result was fatal, because it resulted in death. Therefore, it is important to detect the disease early. The main cause of dementia is Alzheimer's disease. Dementia reduces reasoning skills and interpersonal coping skills, affecting people's ability to function independently. The patient will forget recent events in the first If the disease progresses, they will gradually forget all events. It is essential to diagnose as early as possible. This paper proposes a model that takes images of brain MRI samples as input and determines whether a person has mild, moderate or no Alzheimer's disease as output. We are using the auto-detection system Inception V3 and densenet149 architectures is provided for this purpose. This system uses an edge detection algorithm to provide efficient results. The Inception V3 architecture operates with 94.68% of training accuracy and 88.79% of testing accuracy, and can classify images and the densenet149 architecture operates with 81.18% of training accuracy and 79.73% of testing accuracy.

CHAPTER I

Introduction

1.1 Overview

One of the most important organs in our body is the brain. The brain directs and facilitates all the actions and reactions that allow us to think and believe. Alzheimer's disease is an gradual, degenerative form of brain deterioration. Dementia, a term used to describe memory loss and other cognitive impairments severe enough to interfere with everyday living, is most frequently caused by this. 60 to 80 percent of instances of dementia are caused by Alzheimer's disease. It gradually impairs thinking and memory abilities as well as the capacity to complete even the most basic activities. Symptoms of the late-onset kind often begin to show in the majority of patients in their mid-60s. Rarely, early-onset Alzheimer's strikes between the ages of 30 and 60. For older persons, Alzheimer's disease is the most typical cause of dementia. This damage initially takes place in parts of the brain involved in memory, including the entorhinal cortex and hippocampus. Later, it impacts parts of the cerebral cortex that are involved in language, thought, and social interaction. Eventually, the brain's many other regions suffer harm.

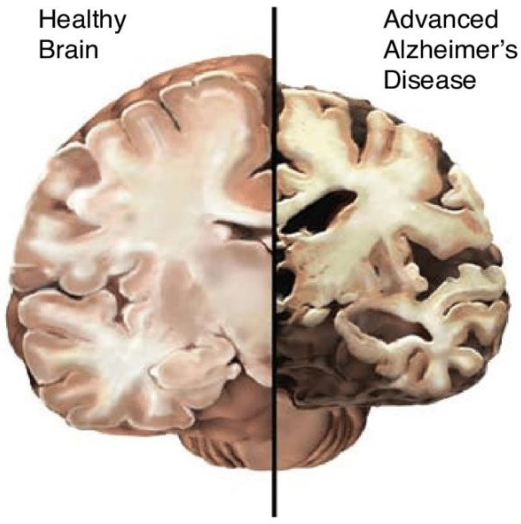

Fig 1.1: Healthy Brain vs. Severe AD Brain [1]

1.2 Motivation

Along with psychological evaluations, brain MRI scanning and analysis is one of the most popular and early forms of Alzheimer's disease detection testing. The doctors study the MRI images and consider any potential indicators of Alzheimer's disease, such as tumors and degenerative changes in brain tissue. The existence of Alzheimer's disease can be detected through manual inspections of MRI data, although this method tends to slow down the speed at which judgments can be reached. The prevalence of dementia symptoms in people with Alzheimer's disease ranges from 60% to 80%, making it the second most severe neurological ailment in the world. This illness has a very high chance of affecting older adults. There is currently no treatment for this illness, however early intervention can decrease the progression of dementia. A healthy diet, regular exercise, social interaction, avoiding head injuries, reading, learning an instrument, and engaging in intellectual pursuits have all been linked to a lower risk of Alzheimer's disease; these activities can improve overall brain health and cognitive function. A model has employed on transfer learning of Inception Net v3 for detecting Alzheimer's disease.

1.3 Research Questions

Focus on the following research questions in order to describe the problem of dementia prediction.

1. What is the problem of the detection of Alzheimer's disease?
2. How to solve this problem?
3. What pre-trained models are used to this dementia prediction problem?
4. What are the limitations and challenges to develop the system?

For the first questions: “What is the problem of the detection of Alzheimer's disease” this problem here is to inspect the quality of MRI images and accuracy given by system.

The second question of “How to solve this problem” is related to image processing and machine learning techniques. Several researches have been done for classifying different types of images like 2D MRI images or 3D scan images. Deep learning is a type of machine learning that improves the effectiveness of systems in problem-solving.

The third question of “What pre-trained models are used to this dementia prediction problem” is used two models like Inception v3 and Densenet149 to find the best system performance.

The final question of the main challenges to develop the system is related to various types of image dataset. More datasets were used to train the system to get more accurate result.

1.4 Objectives of Research

The Main objective of the research is to provide some effective techniques for detecting of Alzheimer's disease from brain MRI scanned images, to improve the efficiency, in terms of accuracy and speed. The primary objectives behind the development of this systems are as follows:

- To design an end-to-end framework for early detection of Alzheimer's disease and medical image classification for various Alzheimer's disease stages.
- Outlining the stages and process of diagnosing Alzheimer's disease, from data collection through illness classification.

1.5 Organization of the Thesis

This thesis has been organized into six chapters. Each chapter provides different concepts of this research.

Chapter I (Introduction): The background of this research is given in this chapter. The objectives and motivation are also within this section.

Chapter II (Literature Review): This chapter presents the procedures, concepts and approaches.

Chapter III (Related Works): This chapter presents the related work and drawbacks of existing system.

Chapter IIV (Proposed Methodology): This chapter describes proposed methodology of this work.

Chapter V (Results and Discussion): This chapter presents the experimental outcomes of the suggested methodology and discuss about how well this work performed using the existing techniques.

Chapter VI (Conclusion): This chapter summarizes the thesis and offers suggestions and recommendations for further research.

CHAPTER II

Literature Review

2.1 Introduction

Alzheimer's disease (AD) is a devastating neurodegenerative disorder that affects millions of people worldwide. Early detection and accurate diagnosis of AD are critical for effective disease management and treatment. In recent years, significant progress has been made in developing biomarkers and imaging techniques that can aid in the early detection and diagnosis of AD. However, there is still a need for a comprehensive review of the current state of research on these approaches to assess their potential for clinical use.

This research aims to conduct a literature review of the latest advancements in biomarkers and imaging technologies for early detection and diagnosis of AD. By critically analyzing existing research, this study aims to identify the most effective and reliable biomarkers and imaging techniques for detecting AD in its early stages. Additionally, the study aims to explore the challenges and limitations associated with these approaches and identify potential solutions to overcome them.

The literature review will focus on the most recent research findings from peer-reviewed journals, conference proceedings, and relevant books. The study can help identify areas of research that need further investigation and highlight potential solutions to overcome challenges and limitations associated with these approaches. Ultimately, this study can contribute to the development of effective strategies for early detection and management of AD, with the goal of improving outcomes for patients and their families.

2.2 Data Preprocessing

Data preprocessing is a crucial step in the process of preparing data for machine learning. It refers to a set of techniques and procedures that are applied to raw data in order to transform it into a format that is suitable for analysis and modeling [2].

2.3 Normalization

The term "Normalization" refers to the transformation of the functions to give them the same measurements. As a result, training stability and model performance are enhanced. [3] Data is represented in pictures as numbers between 0 and 255. The numbers need to be flattened, transformed to float32, and then divided by 255 before being sent to the neural network so that floating point values are in the 0–1 range. It seems that you should not feed data to a neural network that takes relatively large values or heterogeneous data (for example, data where one function is in the range 0-1 and another function is in the range 100-200), because this prevents the network becoming convergent and causes massive gradient updates.

2.4 Data Distribution

A collection of data, or scores, on a certain variable is known as data distribution. After being arranged visually, these scores are often shown in descending order from smallest to greatest [4]. This distribution gives information on how the observations are grouped or distributed. The probability that an observation will have a value greater than, equal to, or less than a specified value may also be determined. In fact, Data Distribution enables data scientists to gain a deeper understanding of the unique characteristics of data. Specially to comprehend its primary patterns so that we may utilize it to anticipate the whole population (even though we never had the opportunity to examine the whole population).

2.5 Confusion Matrix

A table that lists how many accurate and inaccurate predictions a classifier made is called a confusion matrix. It is a N*N matrix used to assess the performance of a classification model, where N is the total number of target classes. The matrix contrasts the predicted goal values with the actual goal values. This gives a complete picture of how well the classification model performs and the many kinds of mistakes it produces. It may be used to measure the efficacy of a classification model by calculating performance measures including accuracy, precision, recall, and fl-score.

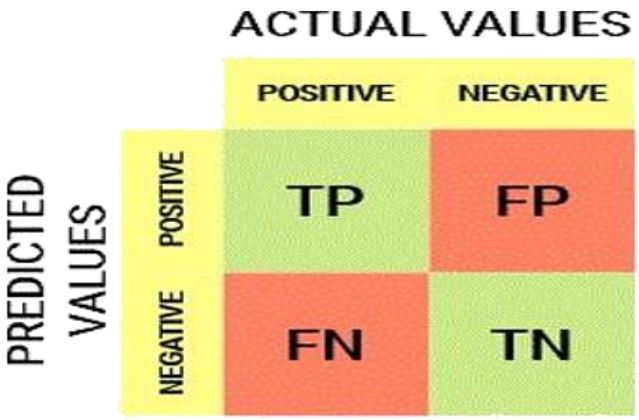

Fig 2.1: Confusion Matrix [5]

In this figure, the target variable has two values: Positive and Negative.The column represents the actual values of target variable. The rows represent the predicted values of the target variable [6].

2.5.1 True Positive (TP)

True Positive (TP) is a term used in statistics and machine learning to describe the number of correct positive predictions made by a model. In other words, a true positive occurs when a model correctly identifies a positive instance from a dataset as positive.

2.5.2 True Negative (TN)

True Negative (TN) is a term used in statistics and machine learning to describe the number of correct negative predictions made by a model. In other words, a true negative occurs when a model correctly identifies a negative instance from a dataset as negative.

2.5.3 False Positive (FP) – Type 1 error

False Positive (FP) is a term used in statistics and machine learning to describe the number of incorrect positive predictions made by a model. In other words, a false positive occurs when a model incorrectly identifies a negative instance from a dataset as positive. It is also known as Type-1 error.

2.5.4 False Negative (FN) – Type 2 error

False Negative (FN) is a term used in statistics and machine learning to describe the number of incorrect negative predictions made by a model. In other words, a false negative occurs when a model incorrectly identifies a positive instance from a dataset as negative. It is also known as Type-2 error.

2.6 Classification Report

It's one of the metrics used to assess the success of a classification-based machine learning model. It shows the precision, recall, F1 score, and support of your model. It allows us to gain a better understanding of our trained model's overall performance. To comprehend a machine learning model's categorization report [7].

2.6.1 Accuracy

Accuracy is a performance evaluation metric used in machine learning to measure the proportion of correctly classified instances out of the total number of instances in a dataset. In other words, accuracy measures how well a model predicts the correct class labels. It is calculated by dividing the number of correctly classified instances by the total number of instances in the dataset.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{2.1}$$

2.6.2 Precision

Precision is a performance evaluation metric used in machine learning to measure the proportion of true positive predictions out of the total number of positive predictions made by a model. In other words, precision measures the accuracy of positive predictions made by the model. It is calculated by dividing the number of true positive predictions by the total number of positive predictions made by the model.

$$Precision = \frac{TP}{(TP + FP)} \tag{2.2}$$

2.6.3 Recall

Recall is a performance evaluation metric used in machine learning to measure the proportion of true positive predictions out of the total number of actual positive instances in the dataset. In other words, recall measures the ability of the model to correctly identify positive instances from the dataset. It is calculated by dividing the number of true positive predictions by the total number of actual positive instances in the dataset.

$$Recall = \frac{TP}{(TP + FN)} \tag{2.3}$$

2.6.4 F1-Score

F1-score is a performance evaluation metric used in machine learning that provides a balanced measure of precision and recall.

$$fl - score = \frac{2 * Precision * Recall}{(Precision + Recall)} \tag{2.4}$$

2.6.5 Support

Support is an important metric to consider when evaluating the performance of a classification model, as it provides information on the distribution of instances across the classes in the dataset. In the context of performance evaluation metrics in machine learning, support refers to the number of instances of each class in a given dataset. It is typically included in classification reports alongside precision, recall, and F1-score for each class.

2.7 Training

A training dataset is a set of data that is used to train a machine learning model. In other words, the model uses this data to learn patterns and relationships in the input data and to develop algorithms that can be used to make predictions about new, unseen data. This model was utilized in 25 epochs, with 32 images being trained in each epoch at a time specified in the code. This model will be repeatedly trained on the same data in the training set throughout each epoch, and it will keep picking up new knowledge about the characteristics of the data.

2.8 Testing

The testing procedure was used in this final phase to verify that the new model functions as intended. Hence, the predicted and actual class labels were compared by running example pictures through the convolutional neural network. After the model has been trained, a set of data called the test set is used to test the model. The training set and validation set are distinct from the test set. Use this model to forecast the results of the unlabeled data in the test set once it has been trained and verified using the training and validation sets. Before deploying the model to production, the test set offers a last-minute verification that it is generalizing correctly.

2.9 NumPy

NumPy (Numerical Python) is a powerful library in Python for performing numerical computations. It provides support for multi-dimensional arrays and matrices, along with a large collection of mathematical functions to operate on these arrays. NumPy is often used for scientific computing, data analysis, and machine learning tasks. NumPy is an essential library for scientific computing and data analysis in Python [8].

2.10 Pandas

Pandas is a popular open-source library in Python that is used for data manipulation and analysis. It provides data structures for efficiently storing and manipulating large datasets, as well as a wide range of functions for working with data. It provides a wide range of functions for working with data in Data Frames and Series. These include functions for selecting and filtering data, sorting and grouping data, and aggregating data. Pandas also provides tools for cleaning and transforming data, such as removing missing values, merging data from multiple sources, and pivoting data.

2.11 TensorFlow

TensorFlow is an open-source machine learning library developed by Google that is widely used in both research and industry. It provides a flexible and powerful platform for building and training machine learning models, including deep neural networks. TensorFlow has main concentration on training dataset [9]. It provides a wide range of built-in functions and classes for creating and manipulating computational graphs. These include functions for defining and

initializing variables, building neural network layers, and optimizing model parameters. TensorFlow also provides support for distributed computing, allowing models to be trained across multiple devices or even multiple machines.

2.12 Keras Library

Python-based Keras is a high-level deep learning package. It includes a number of useful tools for data preprocessing, such as image augmentation and sequence preprocessing. These tools can be used to increase the size of the training data and improve the generalization of the model. It is a powerful and flexible deep learning library that provides a user-friendly interface for building and training complex models [10].

2.13 Flattening

Flattening refers to the process of converting a data into a 1-dimensional vector or array for inputting it to the next layer. Convolutional layer output is flattened to produce a single, lengthy feature vector. Moreover, it has a connection to the final classification model, also known as a fully-connected layer. In other words, all of the pixel data is combined into a single line and connected to the final layer. [11]

2.14 Dense

Each neuron in the dense layer, which is a straightforward layer of neurons, gets input from every neuron in the preceding layer [12]. Based on the results of the convolutional layers, a dense layer is utilized to categorize the images. The neurons in each layer of the neural network calculate the weighted average of their input, and this average is then passed through a non-linear function known as an "activation function."

output = layers. Dense (4, activation='Softmax', name='output') (x)

Here, output layer has 4 neurons with Softmax activation function. Softmax activation function is used when it has 4 or more than 4 classes.

2.15 Convolutional Neural Network

A convolutional neural network (CNN) is a subset of machine learning. A part of machine learning is convolutional neural networks. It is one of the many distinct kinds of artificial neural networks that are used to numerous applications and data sources. Specifically used for image identification and other tasks involving the processing of pixel data, CNN are a type of network design for deep learning algorithms. It is a type of deep neural network that is commonly used in computer vision applications, such as image classification, object detection, and segmentation. CNNs are designed to automatically learn features from raw input data, such as images, by processing them through a series of convolutional and pooling layers. The assumption that the inputs are pictures in CNN architectures enables the model to be trained with certain attributes [13]. There are a number of procedures that must be followed in order for a convolutional neural network to function correctly.

2.16 Adam Optimizer

The Adam optimizer is a commonly used optimization algorithm that is often used to train deep neural networks. The Adam optimizer is a stochastic gradient descent (SGD) optimization algorithm that uses adaptive learning rates to update the model parameters during training. The specific hyperparameters used for the optimizer (such as the learning rate, beta1, and beta2) may vary depending on the specific implementation and training objectives.[14]

2.17 Epoch

In the context of machine learning, an epoch refers to a single iteration of the entire dataset through a machine learning algorithm during training. During the training process, the machine learning algorithm tries to learn the patterns and relationships in the training dataset by adjusting the model's parameters.

2.18 RMSprop Optimizer

RMSprop (short for Root Mean Square Propagation) is an optimization algorithm commonly used in machine learning and deep learning for stochastic gradient descent (SGD). It is an extension of the gradient descent algorithm that tries to overcome its shortcomings such as slow convergence, oscillations, and difficulties with saddle points.

CHAPTER III

Related Works

Various researchers have employed various methods to diagnose Alzheimer's Disease over the years. The following sentences provide a brief overview of the works accomplished so far.

John et al., focus on the detection of Alzheimer's disease using fractal edge detection method [15], in their work. They increase the efficiency of their job and the accuracy of the edge detection mechanism by using filtering techniques like the Sobel and Prewitt filter. Their research uses a mechanism that divides the gray and white matter in the hippocampus mass of the brain to categorize MRI pictures. The changes in brain matter are categorized using fuzzy logic as either the presence of Alzheimer's disease, the potential for Alzheimer's disease to arise, or a healthy brain.

In their research on local MRI analysis for the diagnosis of early and prodromal Alzheimer's disease [16], Chincarini et al. conducted tests to find the existence of conversion probabilities for Alzheimer's disease. In order to do this, characteristics from the hippocampus area of the brain are studied in the volume of interest. By dividing the MRI data into numerous slices and examining the portion that produced the most characteristics, the T1-weighted MRI data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) were subjected to processing in order to extract the volumes of interest. The Random Forest method was used to further analyze these portions, which produced the areas displaying pertinent characteristics.

Bryan [17] observed that variations in cross-site and cross-vendor estimates constrained the use of AI in cerebral blood flow imaging techniques for Alzheimer's disease. Such kinds can be strongly standardized in human eyesight, but they require major advancements in learning how to avoid dangers from ignored and underestimated measured mistakes.

Using MRI surface morphometry mapping, Devanand et al. evaluated local deformations of the hippocampus, parahippocampal gyrus, and entorhinal cortex in predicting conversion from moderate cognitive impairment to having Alzheimer's disease [18]. Their research follows baseline surface morphological surface brain MRI for participants at a single study location. The demographic, clinical, and MRI factors of patients (converters, non-converters to AD), and controls, were compared using ANOVA, ANCOVA, and chi-square tests. 134 patients' hippocampus volumetric and surface morphometry data are taken into account for categorization utilizing the aforementioned techniques.

Moradi et al., used the concept of Low-Density Separation (LDS) in the development of a machine learning framework for early MRI-based Alzheimer's conversion prediction, in MCI subjects [19]. This technique applies the LDS approach to pictures produced from a 1.5 T scanner-based ANDI baseline T1-weighted MP-RAGE sequence. Both supervised and unstructured data are used to train the model at the same time. With the best accuracy in differentiating between people with Alzheimer's disease and those with normal cognition, the elasticnet algorithm is fine-tuned by the unsupervised data to locate the Alzheimer's disease in the brain MRI. To categorize the patients with a higher risk of developing Alzheimer's disease from mild cognitive impairment and the patients with a lower likelihood of such a situation, the characteristics derived from the LSD framework are put to a Random Forest framework.

Escudero et al. [20] proposed a ML approach using biomarkers in their paper. Using a technique for learning locally weighted and biomarkers, they evaluated a customized classifier for the condition. The approach does a first classification attempt before deciding which biomarker to purchase. They separated the MCI patients who converted to AD within a year from those who did not.

CHAPTER IV

Methodology

4.1 Overview

The proposed system is how to identify whether a person's brain is healthy or suffering from Alzheimer's disease. Help with computer vision and deep learning algorithm by using NumPy, Pandas, Tensor flow, Keras library. Those model consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers for Alzheimer's disease detection that can help improve early diagnosis and treatment of the disease.

4.2 Method

- Data Collection.
- Data Preprocessing.
- Model Training (Inception v3 and Densenet149).
- Model Evaluation.
- Results and Discussion.

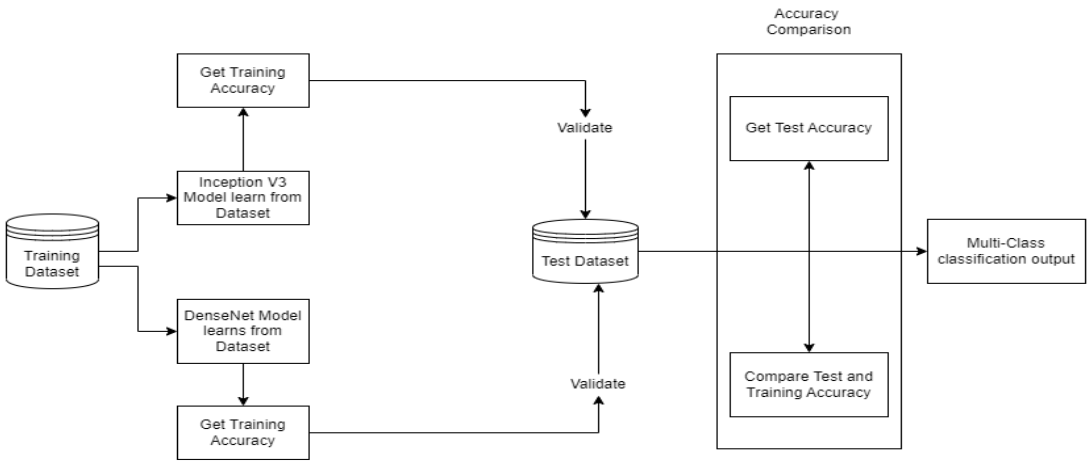

Fig 4.1: Proposed Method

In this figure 4.1, it describes the system architecture diagram provides a conceptual and behavioral overview of the system. It's just a view that shows how the dataset is obtained from the database and how the project modules use this dataset to train various models.

4.3 Data collection

The data is taken from an open online dataset library known as Kaggle [21], and the dataset hasn't yet been utilized in several other academic papers or initiatives. The dataset is open-source. Nearly 6,400 photos from this dataset are divided into four classes: MildDemented, ModerateDemented, VeryMildDemented and Non-Demented. Those datasets are divided into two phases. They are Training data and testing data. This training data is used to adjust the parameters of the model in order to minimize the error or loss function and the test data is used to measure the accuracy and generalization ability of the model on new, unseen data.

4.4 Data preprocessing

In this case data preprocessing means image preprocessing. Because the datasets we collect are in image format. Image processing is a computer technology applied to images that helps to process, analyze and extract useful information from image [2].

Computers see an input image as an array of pixels, and the resolution of the image is a factor. It will see (height * width * dimension) based on the image resolution. It uses a picture of an RGB matrix with dimensions of 250 × 250 * 3. (Here 3 refers RGB values).

The figure below shows the state after data pre-processing:

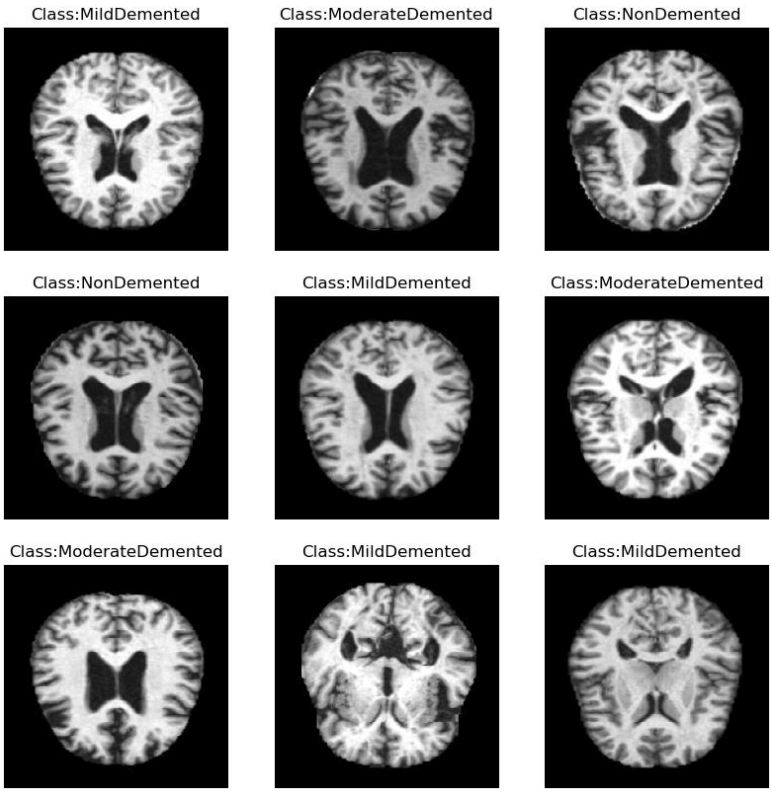

Fig 4.2: Data preprocessing

Therefore, in the image preprocessing step, the image data path must be loaded first, and then this data path is divided into four categories. After converting all images to table size. But at this time, each image has a different matrix or a different pixel size.

4.5 Splitting Dataset

The features are then split between a train dataset and a test dataset. Each deep learning model has two phases, training and testing, where it predicts the data that is given to it. Since the data is utilized in training, this indicates that deep learning model has two phases. The model use the same dataset separated from the original Kaggle dataset and are divided in 9:1 ratio, which has 80% training and 200% validation dataset.

Dataset	**Splitting no. of images**
Training Data	4097
Validation set	1024

Table 4.1: Splitting dataset

4.6 Evaluation metrics

where TP is the number of true positive samples, TN is the number of true negative samples, FP is the number of false positive samples, and FN is the number of false negative samples in the confusion matrix. Sensitivity and specificity are two statistical measures of the performance of a binary classification test that are widely used in medicine. Sensitivity, known as the true positive rate, measures the proportion of positives that are correctly identified. Specificity, known as the true negative rate, measures the proportion of negatives that are correctly identified.

The terms "TP." "FP," "TN," and "FN" refer to the test result and the classification correctness. As a sample, if a disease is the condition, "TP" signifies "correctly diagnosed as diseased," "FP" refers to "incorrectly diagnosed as diseased," "TN" means "correctly diagnosed as not diseased," and "FN" denotes "incorrectly diagnosed as not diseased."

In this case, if the sensitivity of the test is 98%, then the rate of false negatives is 2% (100% - 98%). This means that 2% of the individuals who actually have the condition will be incorrectly identified as not having the condition. Similarly, if the specificity of the test is 92%, then the rate of false positives is 8% (100% - 92%). This means that 8% of the individuals who do not have the condition will be incorrectly identified as having the condition.

Macro averaging is used for models with two or more targets. Some metrics for calculating the overall average are described. First, Macro Average Accuracy calculates the average accuracy for each category. This is called macroscopic resolution. Overall accuracy scores can be determined mathematically by averaging all accuracy scores for different categories.

It is defined in equation (3.5) –

$$Macro - precision = \frac{\sum_{i=1}^{n} Precision}{number\ of\ classes} \tag{4.1}$$

Macro resolution is lower for patterns that not only perform well in general categories, but also perform poorly in rarer categories. As such, it is a harmonic measure of general accuracy. Second, the overall recall average is the average of all the different class recall scores. It is known as macro-recall.

It is defined in equation (3.6) –

$$Macro - recall = \frac{\sum_{i=1}^{n} recall}{number\ of\ classes} \tag{4.2}$$

Finally, the harmonic mean of the macro-precision and the macro-recall was represented by the macro-averaged fl-score, also known as the macro fl-score.

It is defined in equation (3.7) –

$$Macro - fl - score = 2 * \frac{MAP * MAR}{MAP^{-1} + MAR^{-1}} \tag{4.3}$$

Where MAP represents the overall average precision and MAR represents the overall average recall.

4.7 Model Training

In machine learning, training a model refers to the process of optimizing the parameters of a machine learning algorithm to fit the data. The goal of model training is to develop a model that can make accurate predictions on new, unseen data. In this research, two CNN models are used. One of these is Inception v3 and the other is Densenet169.

CNN(Convolutional Neural Networks) are a type of neural network commonly used for image classification, object detection, and other computer vision tasks. The basic architecture of a CNN consists of a series of convolutional layers, pooling layers, and fully connected layers.

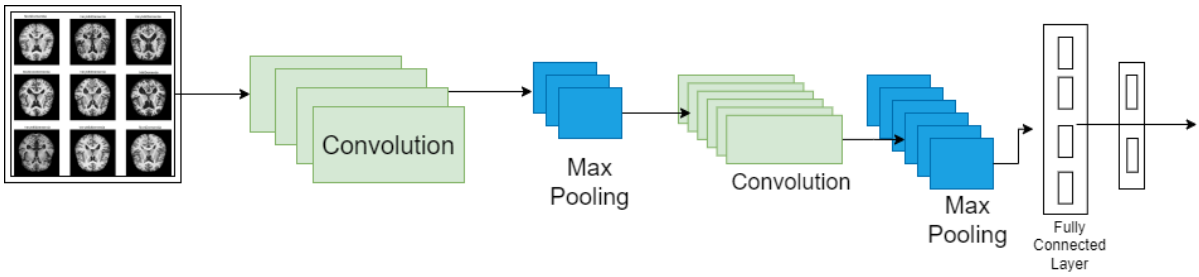

Fig 4.3: Convolutional Neural Network

4.7.1 Inception v3

The Inception v3 model is a convolutional neural network architecture that is commonly used for image classification tasks. Here are the general steps for training an Inception v3 model:

- Data Preparation: The first step in training any neural network is to prepare the data. This involves collecting a dataset of images, labeling them, and splitting them into training, validation, and test sets.

- Preprocessing: The next step is to preprocess the images to make them suitable for input to the neural network. This typically involves resizing the images to a standard size, normalizing the pixel values, and performing data augmentation such as flipping or rotating the images.

- Initialization: The Inception v3 model can be initialized with random weights or pretrained on a large dataset such as ImageNet. Using pretrained weights can significantly speed up the training process and improve the performance of the model.

- Training: The model is then trained using the training set. This involves feeding the images through the network, computing the loss, and using an optimization algorithm such as Stochastic Gradient Descent (SGD) to update the weights of the model.

- Validation: Throughout the training process, the model is evaluated on the validation set to monitor its performance and prevent overfitting. Overfitting occurs when the model becomes too complex and performs well on the training set but poorly on new, unseen data.

- Testing: Once the model has been trained and validated, it can be evaluated on the test set to measure its performance on new, unseen data.

- Fine-tuning: In some cases, the Inception v3 model may not perform well on a specific task or dataset. In such cases, the model can be fine-tuned by adjusting the hyperparameters or architecture of the model, or by training the model on a subset of the original dataset.

Pretrained Inception v3 models are available for use in various deep learning frameworks, including TensorFlow and PyTorch. These pretrained models can be used as a starting point for transfer learning, where the weights of the pretrained model are fine-tuned on a new dataset specific to a particular task or domain. This Inception v3 architecture uses a combination of convolutional layers, pooling layers, and fully connected layers to extract features from input images and classify them into one or more possible categories. The pretrained Inception v3 model can be used as a starting point for transfer learning, where the model's weights are fine-tuned on a smaller dataset for a specific task. This approach can often result in better performance and faster. The Inception v3 model has been used in a wide range of applications, including image classification, object detection, face recognition, and medical image analysis.

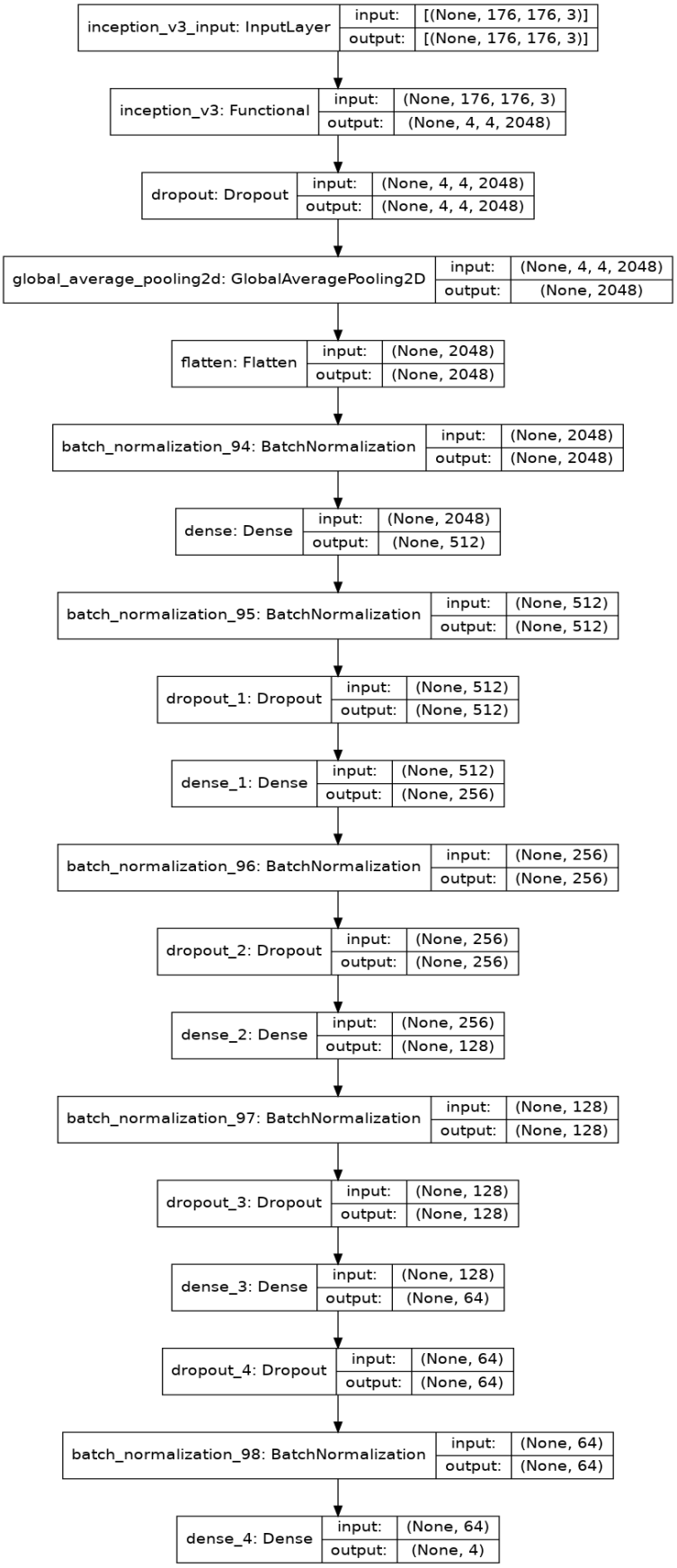

Fig 4.4: Model training method (Inception v3)

4.7.2 Densenet-169

DenseNet-169 is a deep convolutional neural network architecture which consists of 169 layers. It includes some modifications to the original Dense Net design. Specifically, it uses a combination of 1x1, 3x3, and 5x5 convolutional filters in each dense block, which helps to capture features at multiple scales and resolutions. Training method of a DenseNet-169 model involves several steps, including data preparation, model initialization, training, and evaluation. Here is a general overview of the training process:

- Data Preparation: The first step is to prepare the training data. This typically involves collecting a dataset of images, labeling them, and splitting them into training, validation, and test sets. The images may also need to be preprocessed, which can include resizing, cropping, and normalization.

- Model Initialization: The next step is to initialize the DenseNet-169 model. This can involve randomly initializing the weights, or using pretrained weights that have been trained on a large dataset such as ImageNet.

- Training: The model is then trained on the training set using a training algorithm such as stochastic gradient descent (SGD) or Adam. During training, the model parameters are updated to minimize a loss function that measures the difference between the predicted labels and the true labels. The loss function can vary depending on the task, but commonly used ones include cross-entropy and mean squared error.

- Validation: Throughout the training process, the model is evaluated on a validation set to monitor its performance and prevent overfitting. The validation set is used to tune hyperparameters, such as learning rate and regularization strength.

- Testing: Once the model has been trained and validated, it can be evaluated on the test set to measure its performance on new, unseen data.

- Fine-tuning: In some cases, the DenseNet-169 model may not perform well on a specific task or dataset. In such cases, the model can be fine-tuned by adjusting the hyperparameters or architecture of the model, or by training the model on a subset of the original dataset.

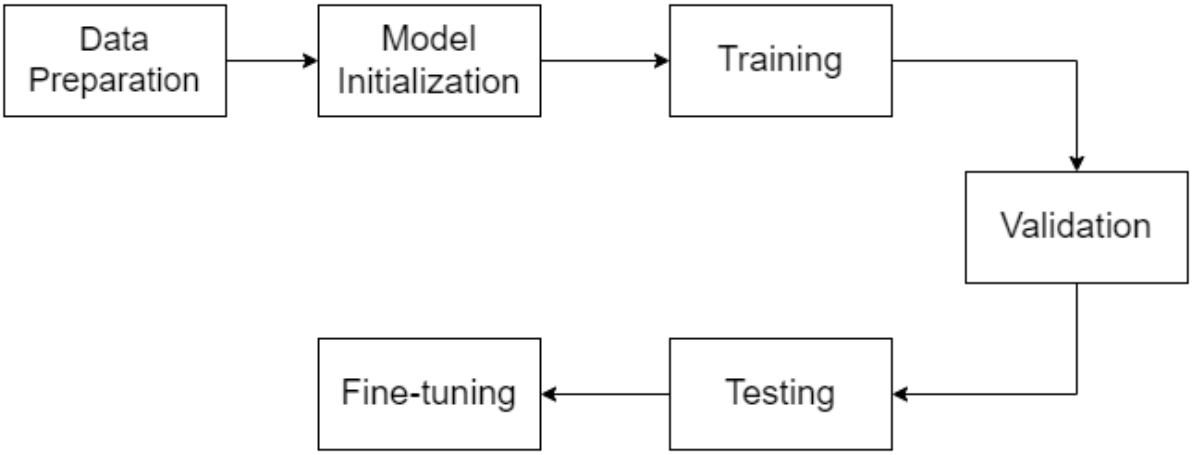

Fig 4.5: Model training method (DenseNet-169)

In DenseNet, each layer receives the feature maps from all preceding layers and passes its own feature maps to all subsequent layers. This approach leads to more efficient use of parameters and reduces the vanishing gradient problem, as information from earlier layers can be easily propagated to later layers.

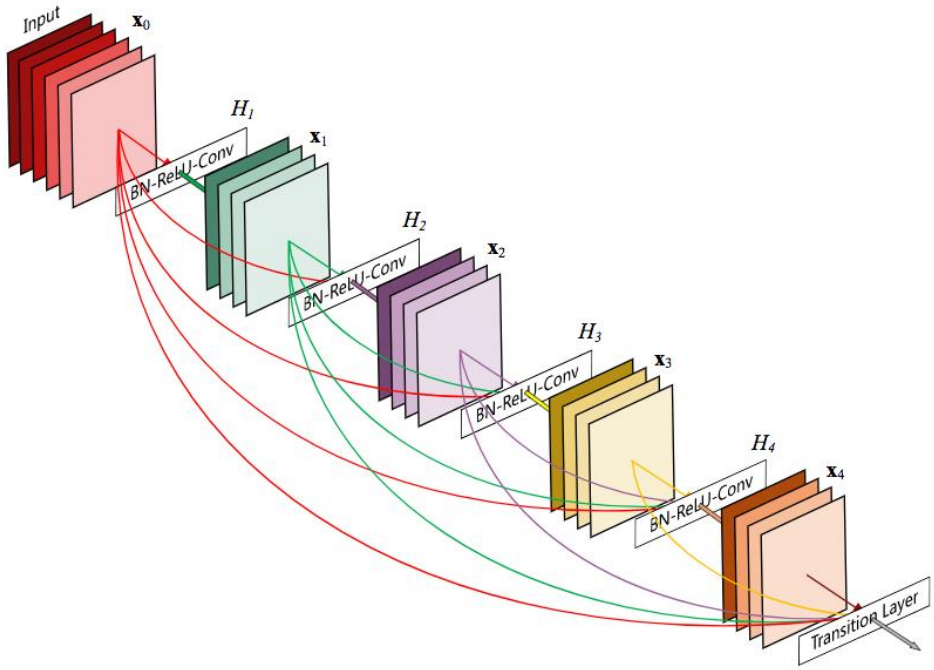

Fig 4.6: DenseNet Architecture [23]

CHAPTER V

Result and Discussion

5.1 Overview

As stated, the Inception v3 model and DenseNet169 model have been used to evaluate this system's performance. Inception v3 is a deep convolutional neural network architecture that was created for image recognition and classification tasks. In 2015, Google researchers created it and released it.

As it is said before, in dataset there are almost 6400 images and it divided into two phases. With Training phase consist of 5121 images and test phase consist of 1279 images. The dataset images were collected from image datasets available in Kaggle [21],

The pictures below are some training and testing sample images…

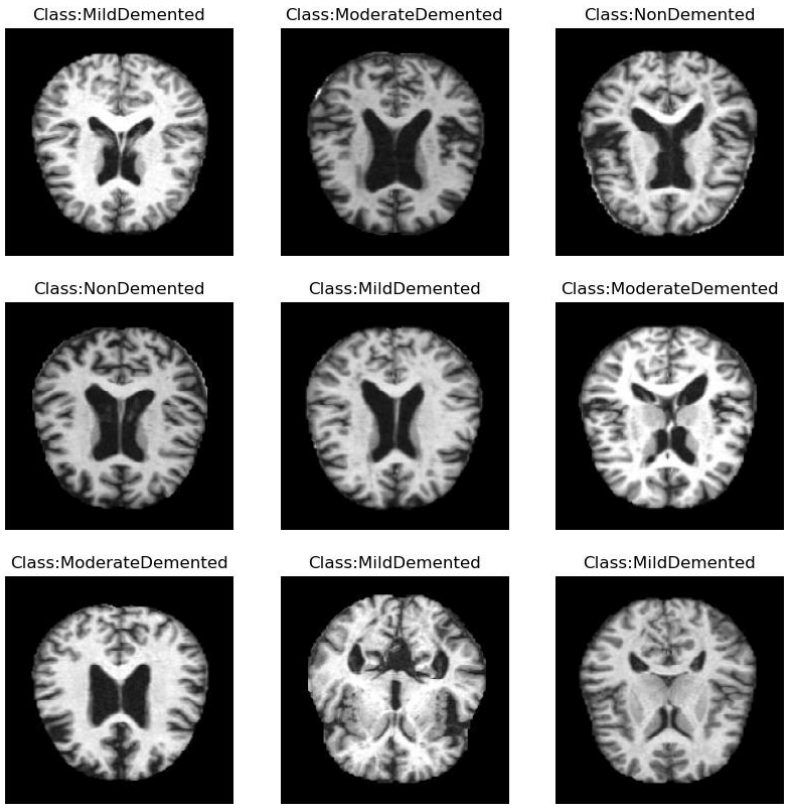

Fig 5.1: Training Data

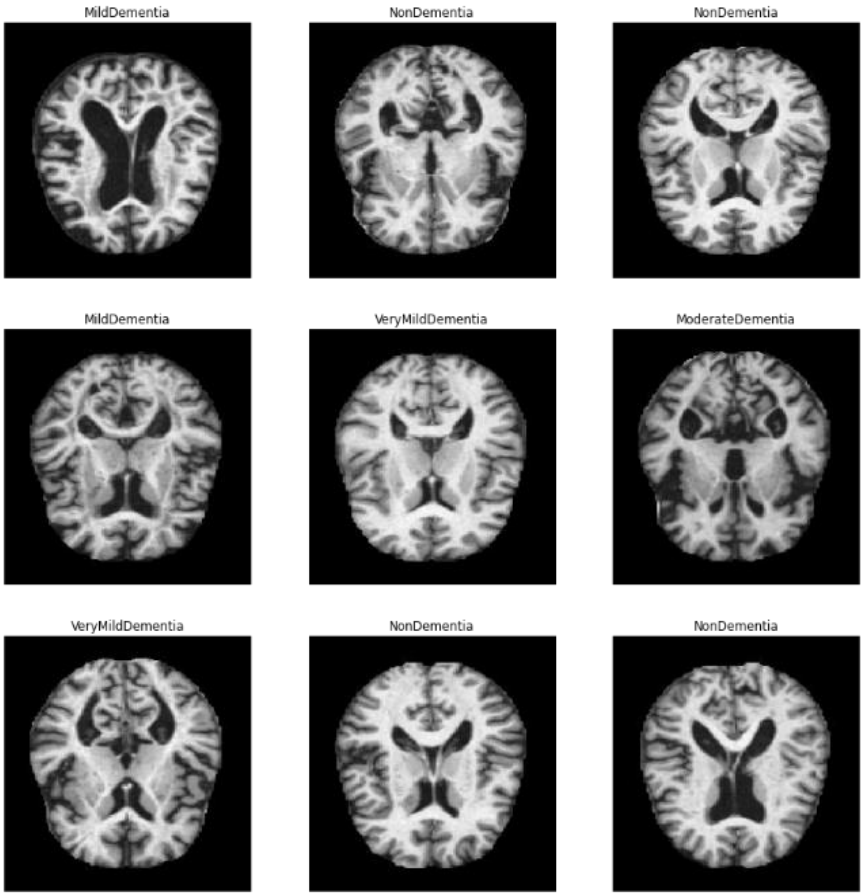

Fig 5.2: Testing Data

5.2 Experimental result

This experimental results in the context of machine learning or artificial intelligence refer to the outcomes obtained when a model or algorithm is tested on a dataset. These results are used to evaluate the performance of the model and its ability to generalize to new data.
Here the Experimental result of two models:

5.2.1 Experimental result (Inception v3)

In Inception v3 model, the most important thing applied is whether or not the brain is affected by Alzheimer's disease. This matter depends entirely on the accuracy of the measurement. We ran this model for different number of epochs. The maximum accuracy was obtained after running 100 epochs. The results shown below,

Epochs	Test Loss	Test Accuracy%
25	0.636	84.21
30	0.557	85.77
40	0.559	85.67
50	0.467	86.17
100	**0.3033**	**88.79**

Table 5.1: Experimental result (Inception v3)

5.2.1.1 Defined hyperparameters

Here, the hyper-parameter constants were the batch size and the initial number of training epochs.

Parameter	Value
Epochs	100
Batch_size	6500
Image size	(176 , 176)

Table 5.2: Hyperparameters Values

The training data will then be loaded and preprocessed, and at this point, all of the imagepaths in the dataset will be grabbed, the data will be initialized, the lists will be labeled, a loop will be created over the imagepaths, and pre-processed pictures will be loaded. Pre-processing processes include scaling the pixel intensities in the input picture to the range [-1, 1], converting to array format, resizing the image to 250*250 pixels, and adding the pre-processed image and associated label to the data and label lists, respectively. Making sure the training data is in Numpy array format is another important step.

5.2.1.2 Model Summary

model. summary ()

The model's summary will then be output after that,

Model: "inception_cnn_model"

Layer (type)	Output Shape	Param #
inception_v3 (Functional)	(None, 4, 4, 2048)	21802784
dropout (Dropout)	(None, 4, 4, 2048)	0
global_average_pooling2d	(None, 2048)	0
flatten (Flatten)	(None, 2048)	0
batch_normalization_94	(None, 2048)	8192
dense (Dense)	(None, 512)	1049088
batch_normalization_95	(None, 512)	2048
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
batch_normalization_96	(None, 256)	1024
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
batch_normalization_97	(None, 128)	512
dropout_3 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
dropout_4 (Dropout)	(None, 64)	0
batch_normalization_98	(None, 64)	256
dense_4 (Dense)	(None, 4)	260

Total params: 23,036,644

Trainable params: 1,227,844

Non-trainable params: 21,808,800

5.2.1.3 Training model History

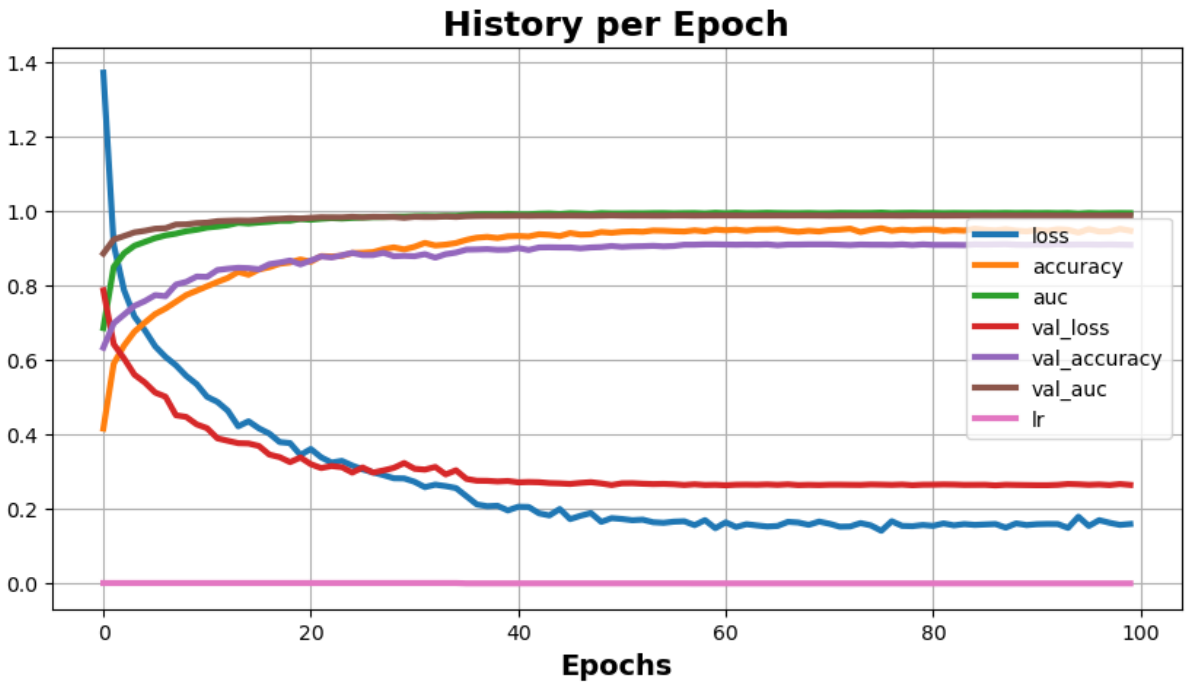

Fig 5.3: Model History (Inception v3)

5.2.1.4 Training Accuracy

Accuracy is the measurement of how accurate this model's prediction is compared to the true data. Here the training accuracy is 94.68%.

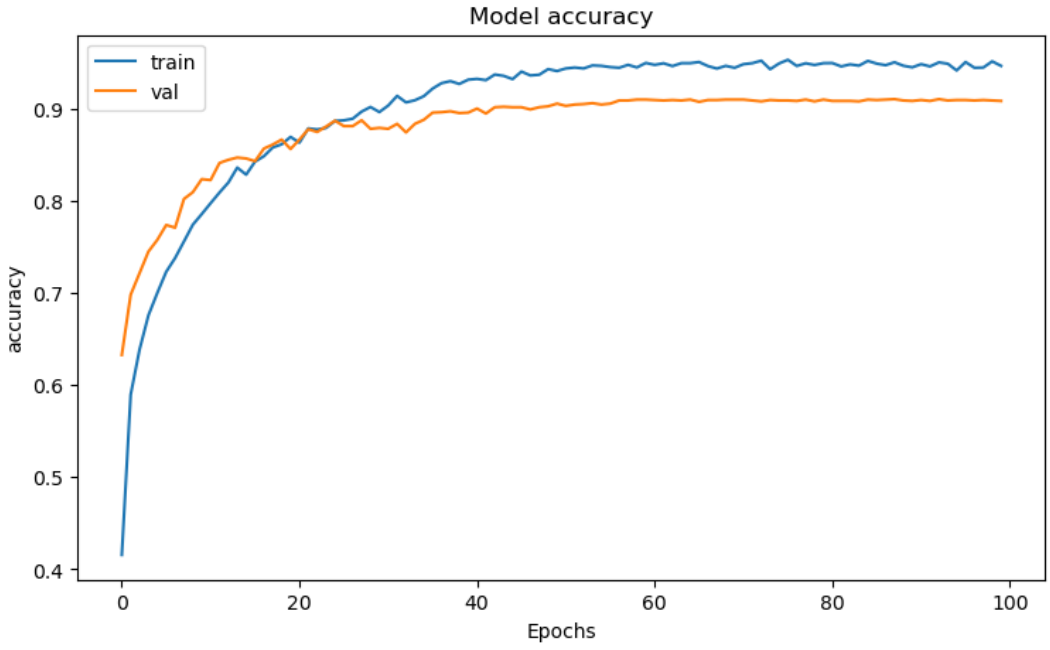

Fig 5.4: Training and Validation accuracy

5.2.1.5 Training Loss

In machine learning, training loss is a measure of how well a machine learning algorithm is performing on the training data. This model updates its parameters based on the loss value computed for each training example. The goal of the training process is to minimize the loss function by adjusting the model's parameters. Here the training loss is 0.159

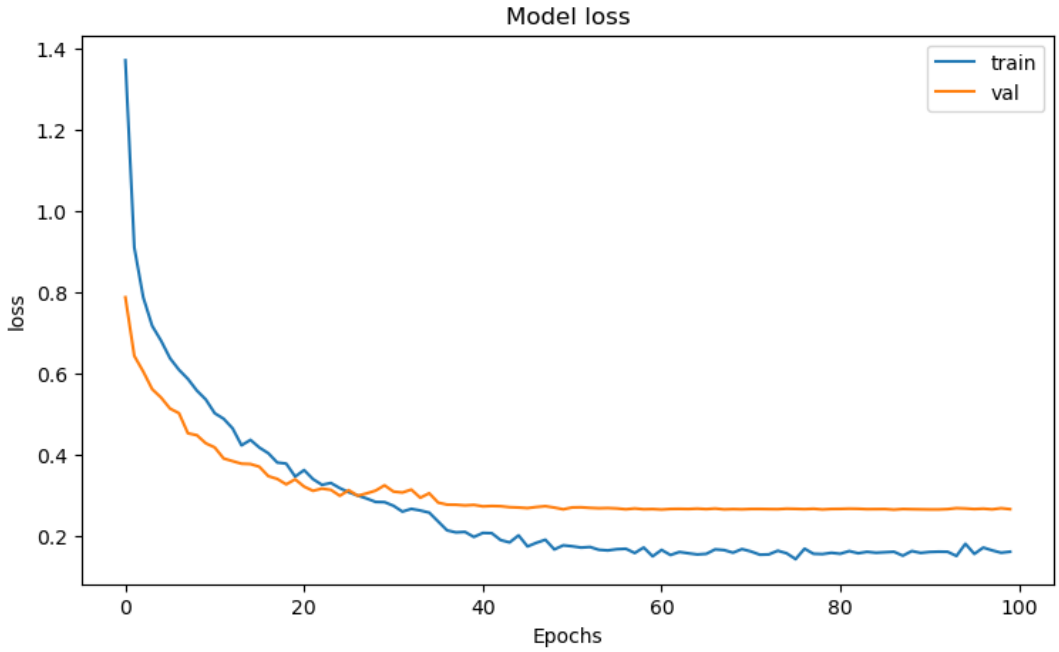

Fig 5.5: Training and Validation Loss

The accuracy, recall, or fl-score average is known as the macro average,and the weighted avg is just the weighted average of precision or recall or fl-score.

	precision	**recall**	**f1- score**	**support**
NonDemented	0.93	0.94	0.93	639
VeryMildDemented	1.00	1.00	1.00	635
MildDemented	0.85	0.79	0.82	662
ModerateDemented	0.78	0.83	0.80	624
Micro avg	0.89	0.89	0.89	2560
Macro avg	0.89	0.89	0.89	2560
Weighted avg	0.89	0.89	0.89	2560
Samples avg	0.89	0.89	0.89	2560

Table 5.3: Evaluation matrix

So now, in those curves there are four metrics: training loss, validation loss, training accuracy, and validation accuracy. An algorithm for machine learning is optimized using a loss function. Loss is computed using training and validation data. A model's loss value indicates how well or poorly it performs after each optimization cycle. If the model has been appropriately trained, it may be assessed by looking at the training loss, validation loss, training accuracy, and validation accuracy across a number of epochs.

5.2.1.6 Evaluation model

Evaluation is the process of measuring an image processing model's efficiency. The purpose of assessment is to determine how accurately the model is performing out the desired image processing task**.** A short part of the code section is given below:

After run that section the output is given below:

40/40 [==========] - 143s 4s/step - loss: 1.3686 - accuracy: 0.8621
Accuracy: % 88.79
Loss: 0.3033

5.2.1.7 Confusion Matrix

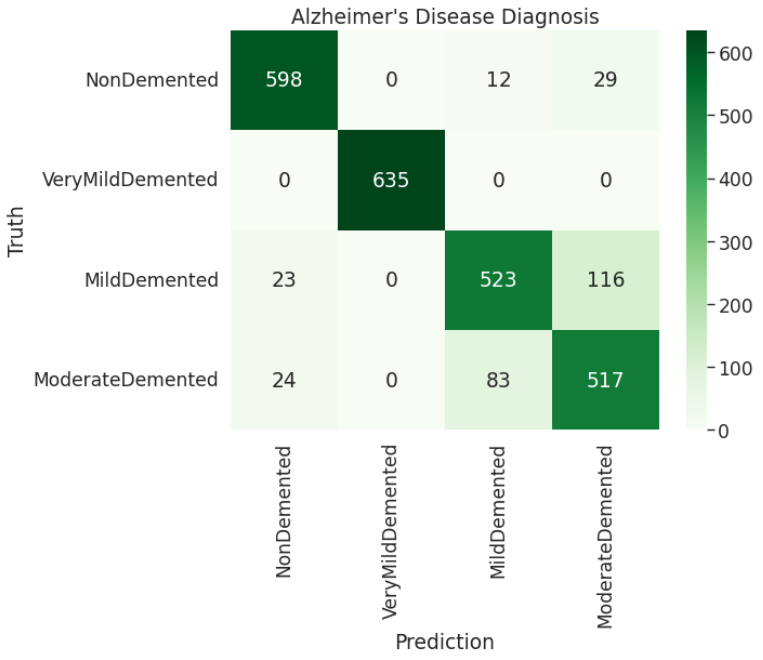

Fig 5.6: Confusion matrix

This confusion matrix provides a way to evaluate the performance of an image classification model by computing various metrics such as accuracy, precision, recall, and F1 score.

5.2.2 Experimental result (Densenet-169)

In Densenet-169 model, the most important thing applied is whether or not the brain is affected by Alzheimer's disease. This matter depends entirely on the accuracy of the measurement. We ran this model for different number of epochs. The maximum accuracy was obtained after running 25 epochs. The results shown below,

Epochs	**Loss**	**Accuracy%**
10	0.9708	79.05
25	**0.9506**	**80.0083**
50	1.0749	78.89
100	1.0689	77.67

Table 5.4: Experimental result (Densenet-169)

Here, every cycle a model goes through the whole training dataset is referred to as an epoch. The outcome should improve in terms of predicting the provided unknown data, which is the test data, for instance, if the neural network is fed training data over a longer period of time (more than one epoch). In this approach, 25 epochs were applied to achieve the best accuracy results of predicting the test data.

In this Paper, The Inception-v3 model has been trained on large datasets like ImageNet and achieved state-of-the-art performance on various image classification tasks. This paper reported an overall lowest accuracy of 82.72% and a highest accuracy of 86.21% using this Alzheimer's dataset which was got from Kaggle.

5.2.2.1 Defined hyperparameters

Here, the hyper-parameter constants were the batch size and the initial number of training epochs.

Parameter	**Value**
Epochs	25
Batch_size	32
Image size	(224 , 224)

Table 5.5: Hyperparameters Values

The training data will then be loaded and preprocessed, and at this point, all of the image paths in the dataset will be grabbed, the data will be initialized, the lists will be labeled, a loop will be created over the image paths, and pre-processed pictures will be loaded. Pre-processing processes include scaling the pixel intensities in the input picture to the range [-1, 1], converting to array format, resizing the image to 224*224 pixels, and adding the pre-processed image and associated label to the data and label lists, respectively. Making sure the training data is in NumPy array format is another important step.

5.2.2.2 Model Summary

model. summary ()

The model's summary will then be output after that,

Model: "sequential"

Layer (type)	Output Shape	Param #
densenet169 (Functional)	(None, 7, 7, 1664)	12642880
dropout (Dropout)	(None, 7, 7, 1664)	0
global_average_pooling2d	(None, 1664)	0
flatten (Flatten)	(None, 1664)	0

batch_normalization	(None, 1664)	6656
dense (Dense)	(None, 64)	106560
batch_normalization_1	(None, 64)	256
dense_1 (Dense)	(None, 64)	4160
batch_normalization_2	(None, 64)	256
dense_2 (Dense)	(None, 64)	4160
batch_normalization_3	(None, 64)	256
activation_2 (Activation)	(None, 64)	0
dropout_3 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2080
batch_normalization_5	(None, 32)	128
dense_5 (Dense)	(None, 4)	132

==========

Total params: 12,768,708

Trainable params: 121,988

Non-trainable params: 12,646,720

5.2.2.3 Training model History

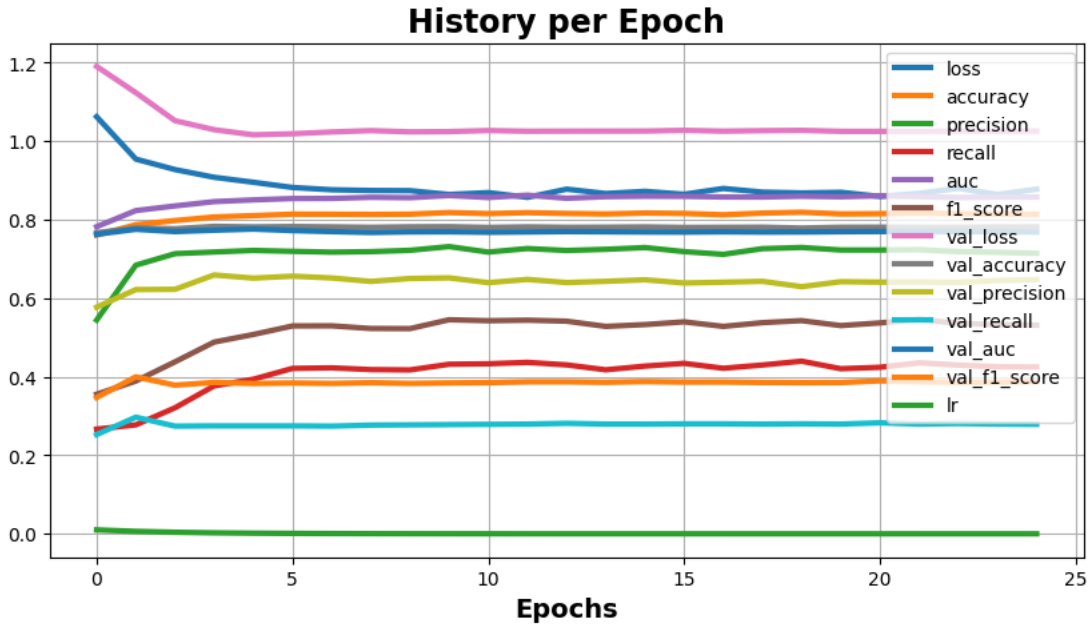

Fig 5.7: Model History (Densenet-169)

5.2.2.4 Training Accuracy

Accuracy is the measurement of how accurate this model's prediction is compared to the true data. Here the training accuracy is 82.29%.

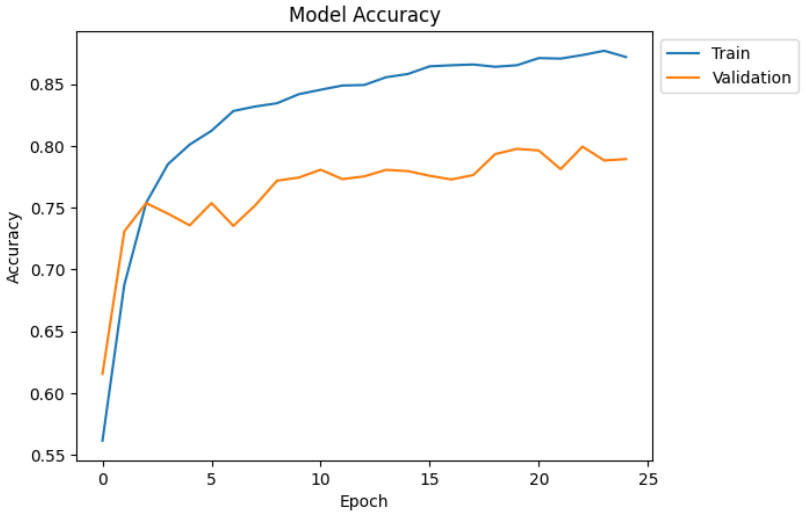

Fig 5.8 : Training Accuracy

5.2.1.5 Training Loss

In machine learning, training loss is a measure of how well a machine learning algorithm is performing on the training data. This model updates its parameters based on the loss value computed for each training example. The goal of the training process is to minimize the loss function by adjusting the model's parameters. Here the training loss is 0.9506

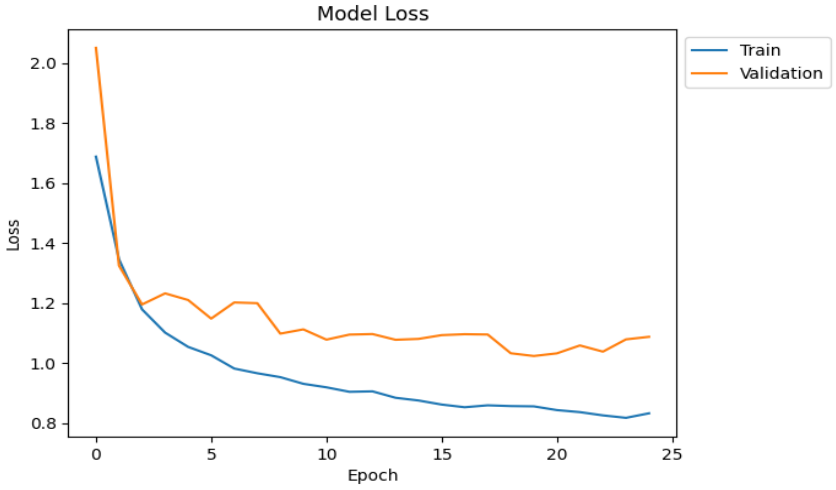

Fig 5.9: Training Loss

The accuracy, recall, or fl-score average is known as the macro average and the weighted avg is just the weighted average of precision or recall or fl-score.

Classification report	Value
precision	0.67
recall	0.38
AUC	0.82
f1- score	4.88

Table 5.6: Evaluation matrix

5.2.2.6 Confusion Matrix

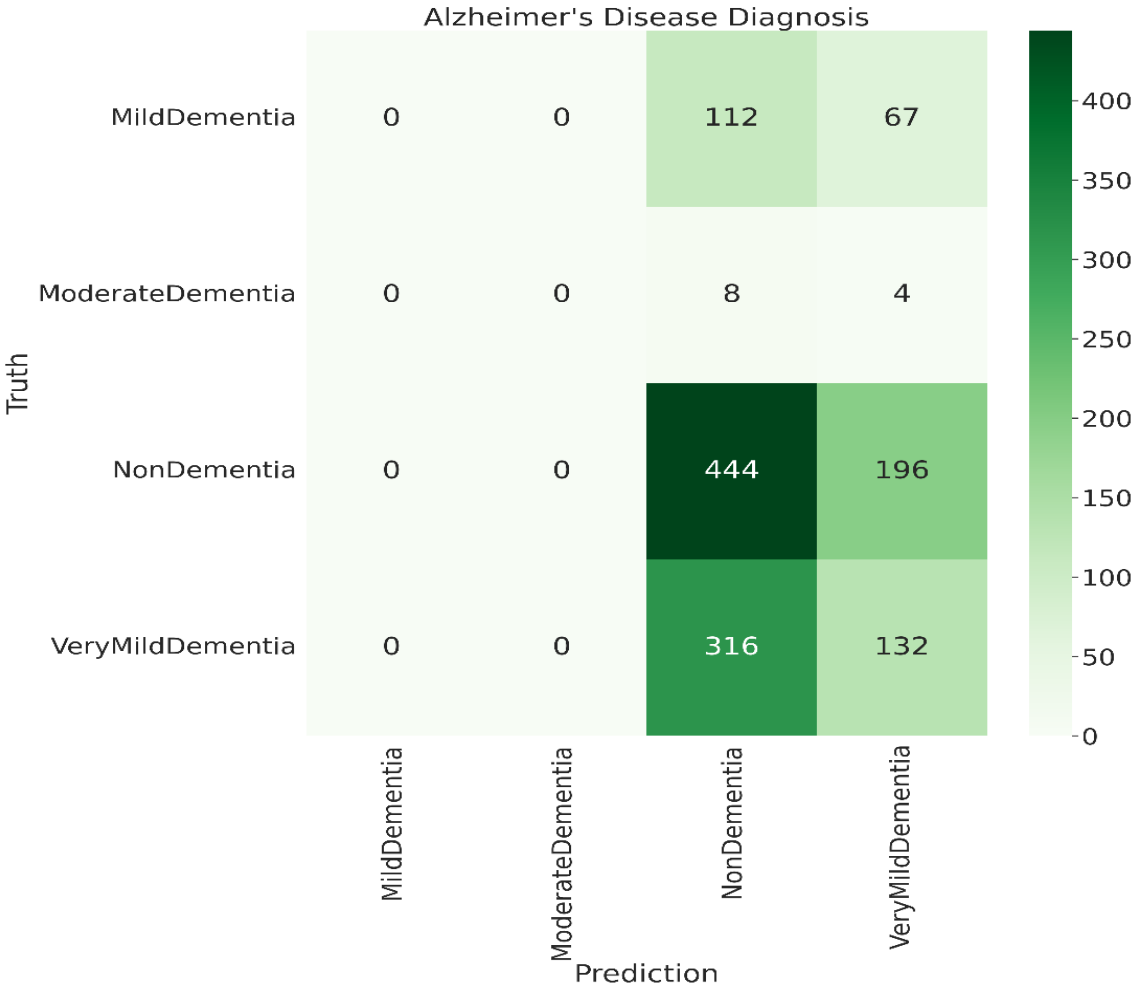

Fig 5.10: Confusion matrix

5.3 Observation

Alzheimer's disease is the leading cause of dementia. This paper decides a imminent arrangement for identifying the illness at an early arrange. The models utilized in this paper have effectively classified the pictures into the fitting four classes and undoubtedly given us with promising comes about. Here Inception v3 performs better than DenseNet-169. Advance research is required to guarantee that this specific demonstrate can be implemented in clinical settings, increasing the health care rate against this specific disease.

5.4 Grad-Cam view

Grad-CAM (Gradient-weighted Class Activation Mapping) is a technique used in image processing models to visualize and understand which parts of an input image are important for a specific classification decision made by the model.We used the Grad-cam technique in this model to help improve the interpretability of image processing models by providing insight into which parts of the input image are important.

After applying the Grad-Cam procedure, the sample data of classes:

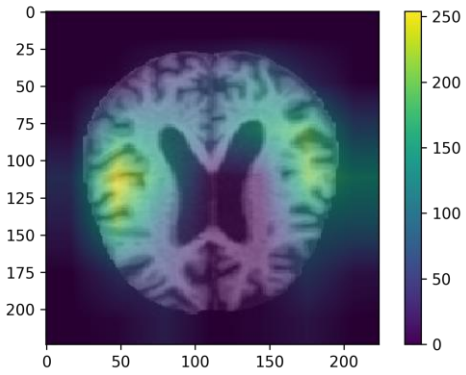

Fig 5.11: MildDemented _Grad_view

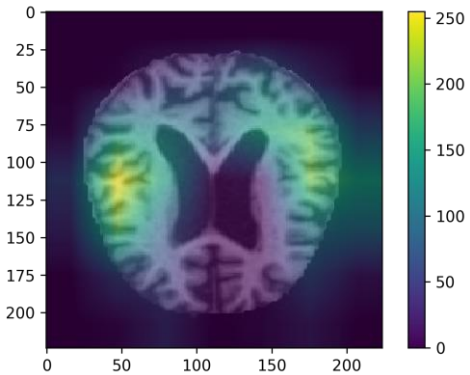

Fig 5.12: ModerateDemented _Grad_view

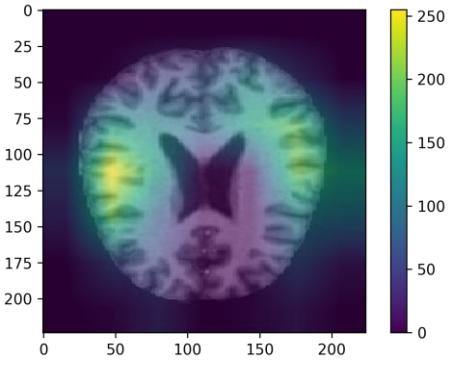

Fig 5.12: NonDemented _Grad_view

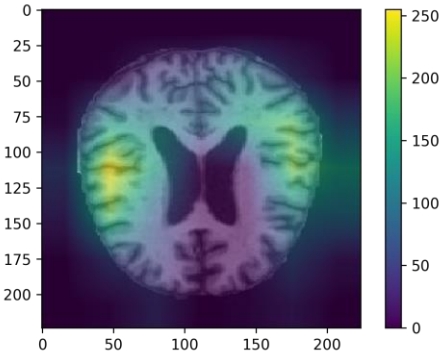

Fig 5.12: VeryMildDemented _Grad_view

This Grad-CAM heatmap can be overlaid on top of the original input image to provide a visual representation of the regions the model is focusing on to make its classification decisions.

Grad-CAM generates a heatmap that highlights the regions of the image that are most important for the prediction made by a particular neuron in the neural network. The heatmap is generated by computing the gradient of the target class with respect to the feature maps produced by the final convolutional layer of the network. The resulting gradient is then averaged over each feature map to produce a heatmap that indicates the regions of the input image that are most relevant to the target class.

The color bar is a helpful tool for interpreting the heatmap and understanding which parts of the image are most important for the model's prediction. By examining the color bar, one can determine the threshold values for the heatmap that correspond to the areas of the image that the model is paying attention to. In general, the color bar for a Grad-CAM heatmap is a horizontal bar that is placed underneath the image, with ticks on either end to represent the minimum and maximum values of the heatmap, and labels or color swatches in between to indicate the range of colors and values.

CHAPTER VI

Conclusion

The most common cause of dementia is Alzheimer's disease. This study develops a potential method for identifying the illness at an early stage. The models utilized in this work were successful in classifying the photos into the proper four groups, and they did in fact provide us promising results. To ensure that this particular model can be used in clinical settings and improve the rate of treatment for this particular condition, more study is needed. People should be made aware of this condition and advised to get themselves evaluated. In the current dataset, we only had 5121 and 1279 images for training and testing.

5.1 Limitations and Future Work

In this model has shown promise in detecting Alzheimer's disease, it also has some limitations, including:

i. Small sample size: Many studies using deep learning models to detect Alzheimer's disease have been conducted on relatively small sample sizes, which can limit the generalizability of the results.

ii. Dependence on high-quality images: The accuracy of the Inception v3 and Densenet-169 model in detecting Alzheimer's disease is highly dependent on the quality of the MRI images used. Poor quality images can lead to misdiagnosis or inaccurate results.

iii. Limited applicability: The Inception v3 and Densenet-169 model is designed specifically for detecting Alzheimer's disease from brain MRI images and may not be applicable to other types of data or for detecting other diseases.

When learning to identify the pictures, this image recognition system does not take into account the various kinds of brain matter, such as white matter and gray matter. This method may be enhanced by giving users the ability to distinguish between different types of brain matter.

REFERENCES

[1] The Alzheimer's Association, © 2007 Alzheimer's Association.org, https://www.nia.nih.gov/health/alzheimers-disease-fact-sheet.

[2] "Machine Learning Image Processing". AI & Machine Learning Blog, 2021. [Online] Available: https://nanonets.com/blog/machine-learning-image-processing/

[3] J. Brownlee, "How to use Data Scaling Improve Deep Learning Model Stability and Performance", Machine Learning Mastery, 2021. [Online] Available: https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/

[4] J. Brownlee, "A Gentle Introduction to Statistical Data Distributions", Machine Learning Mastery, 2022. [Online].Available:https://machinelearningmastery.com/statistical-data-distributions/. [Accessed: 25-Jun-2022].

[5] "Confusion Matrix for Machine Learning",Available: https://www.turing.com/kb/how-to-plot-confusion-matrix

[6] "Confusion Matrix for Machine Learning", Analytics Vidhya, 2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix- machine-learning/. [Accessed: 25-Jun-2022].

[7] A. Kharwal, "Classification Report in Machine Learning", Data Science Machine Learning | Python | C++ | Coding | Programming | JavaScript, 2022. [Online]. Available:https://thecleverprogrammer.com/2021/07/07/classification report-in-machine-learning/. [Accessed: 25-Jun-2022]

[8] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.

[9] "TensorFlow", TensorFlow, 2021. [Online]. Available: https://www.tensorflow.org/

[10] "Core - Keras Documentation". keras.io. Retrieved 2018-11-14.

[11] "The Most Intuitive and Easiest Guide for CNN", Medium, 2021. [Online]. Available: https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480.

[12] "Introduction to Convolutional Neural Networks (CNN) with TensorFlow", Medium, 2021.[Online].Available:https://towardsdatascience.com/introduction-to-convolutional- neural-networks-cnn-with-tensorflow-57e2f4837e18.

[13] T. A. M. a. S. A.-Z. S. Albawi, "Understanding of a convolutional neural network, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8308186.

[14] "Intuition of Adam Optimizer GeeksforGeeks", GeeksforGeeks, 2021. [Online]. Available: https://www.geeksforgeeks.org/intuition-of-adam-optimizer.

[15] Aston Zhang et al. (2020). Dive into Deep Learning Release 0.7.1. Chapter 11, equation (11.9.1).

[16] D.P. Devanand et al. (2012). MRI hippocampal and entorhinal
cortex mapping in predicting conversion to Alzheimer's disease.
NeuroImage 60 (2012) 1622–1629.

[17] Bryan, R. Nick. "Machine learning applied to Alzheimer disease." ,
2016, pp. 665-668

[18] John R et al. (2018). Detection of Alzheimer’s Disease Using Fractional Edge Detection. Global J Technol Optim 9: 230. doi:10.4172/2229-8711.1000230

[19] Andrea Chincarini et al. (2011). Local MRI analysis approach in the diagnosis of early and prodromal Alzheimer's disease. NeuroImage 58 (2011) 469–480.

[20] Escudero, Javier, Emmanuel Ifeachor, John P. Zajicek, Colin Green, James Shearer, Stephen Pearson, and Alzheimer's Disease Neuroimaging Initiative. "Machine learning-based method for personalized and costeffective detection of Alzheimer's disease." IEEE transactions on biomedical engineering 60, no. 1, 2012, pp. 164-168

[21] Kaggle Dataset Available:"https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images"

[22] https://medium.com/@AnasBrital98/inception-v3-cnn-architecture-explained-691cfb7bba08

[23] Densenet architecture, Available:"https://arthurdouillard.com/post/densenet/"