

ALZHEIMER'S DISEASE PREDICTION

A TOOL FOR PREDICTING ALZHEIMER IN PATIENTS

A PROJECT REPORT

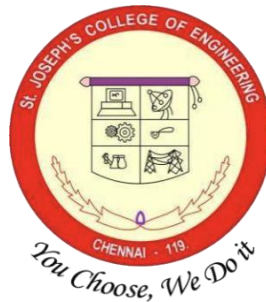
Submitted by

SIVA SANKAR G	312321205156
SURYA R	312321205171

in partial fulfillment for the award of the degree

of

**BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY**



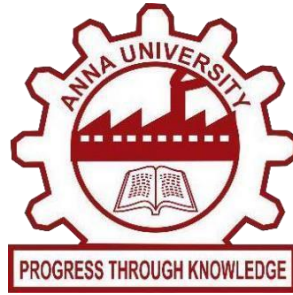
St. JOSEPH'S COLLEGE OF ENGINEERING, CHENNAI 600 119

(An Autonomous Institution)

ANNA UNIVERSITY: CHENNAI 600025

APRIL 2024

ANNA UNIVERSITY: CHENNAI 600 025



BONAFIDE CERTIFICATE

Certified that this project report “ALZHEIMER’S DISEASE PREDICTION” is the bonafide work of **SIVA SANKAR G (312321205156)** and **SURYA R (312321205171)** who carried out the project under my supervision.

SIGNATURE

Project Guide,

Mrs.UTTHIRAKUMARI.R

B.E.,M.E.,

Assistant Professor,

Department of Information Technology,

St. Joseph’s College of Engineering, OMR,

Chennai- 600119.

SIGNATURE

Head of the Department

Ms. G. Latha Selvi M.E., (Ph.D.),

Associate Professor,

Department of Information Technology,

St. Joseph’s College of Engineering, OMR,

Chennai- 600119.

CERTIFICATE OF EVALUATION

College name : St. Joseph's College of Engineering

Branch & Semester : Information Technology (VI)

SL. NO	NAME OF THE STUDENT	TITLE OF THE PROJECT	NAME OF THE SUPERVISOR WITH DESIGNATION
1	SIVA SANKAR G (312321205156)	ALZHEIMER'S DISEASE PREDICTION	Mrs.UTTHIRAKUMARI BE.,ME., ASSISTANT PROFFESSOR St. JOSEPH'S COLLEGE OF ENGINEERING
2	SURYA R (312320205042)		

The report of the project work submitted by the above students in partial fulfillment for the award of Bachelor of Technology degree in Information Technology of Anna University were evaluated and confirmed to be reports of the work done by the above students.

Submitted to Project and Viva Examination held on_____.

(INTERNAL EXAMINER)

(EXTERNAL EXAMINER)

ACKNOWLEDGEMENT

The contentment and elation that accompany the successful completion of any work be incomplete without mentioning the people who made it possible. At the outset we like to express our sincere gratitude in thanking our beloved Chairman, **Dr. B. Babu Manoharan, M.A., M.B.A., Ph.D.**, for his constant guidance and support.

We would like to express our heartfelt thanks to our respected our Managing Director **Mr. B. Shashi Sekar, M.Sc.**, and our Executive Director, **Mrs. B. Jessie Priya, M.Com.**, for providing ample facilities in the institution.

We would like to extend our sincerest and most heartfelt gratitude to our beloved Principal **Dr. Vaddi Seshagiri Rao, M.E., M.B.A., Ph.D., F.I.E.**, for having encouraged us to do our under graduation in Information Technology in this esteemed college.

We express our sincere thanks and most heartfelt sense of gratitude to our Head of the Department-**Ms. G. Lathaselvi, B.E., M.E., (Ph.D.)**, for her guidance and assistance in solving the various intricacies involved in the project.

It is with a deep sense of gratitude that we acknowledge our indebtedness to our supervisor **Mrs.UTTHIRAKUMARI R, B.E.,M.E.**, a perfectionist for his expert guidance and connoisseur suggestion.

Last but not the least, we thank our family members and friends who have been the greatest source of support to us.

ABSTRACT

Alzheimer's disease (AD) is the most common cause of dementia globally. It steadily worsens from mild to severe, impairing one's ability to complete any work without assistance. It begins to outstrip due to the population ages and diagnosis timeline. For classifying cases, existing approaches incorporate medical history, neuropsychological testing, and Magnetic Resonance Imaging (MRI), but efficient procedures remain inconsistent due to lack of sensitivity and precision. The Convolutional Neural Network (CNN) is utilized to create a framework that can be used to detect specific Alzheimer's disease characteristics from MRI images. By considering four stages of dementia and conducting a particular diagnosis, the proposed model generates high-resolution disease probability maps from the local brain structure to a multilayer perceptron and provides accurate, intuitive visualizations of individual Alzheimer's disease risk. To avoid the problem of class imbalance, the samples should be evenly distributed among four types of MRI images Mild Demented, Moderate Demented, Non-Demented, Very Mild Demented the classes DenseNet169 algorithm classification. The obtained MRI image dataset from Kaggle has a major class imbalance problem. A DenseNet169 algorithm classification is proposed to detect the dementia stages from MRI. Which is superior to existing methods, we also used the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset to predict AD classes in order to assess the efficacy of the proposed model. The objective of this work is to bring some useful information in simpler form in front of the users, especially for the medical staff treating the patient. The aim of this work is to define an algorithm that will result in an extracted image of the tumor from the MR brain image.

TABLE OF CONTENTS

	ABSTRACT	i
1	INTRODUCTION	
	1.1 QR CODE PRESERVE DESCRIPTION	1
	1.2 SYSTEM OVERVIEW	2
	1.3 SCOPE	3
2	LITERATURE REVIEW	4
3	SYSTEM DESIGN	
	3.1 UNIFIED MODELLING LANGUAGE	
	3.1.1 Use case diagram for Alzheimer disease	
	3.1.2 Class diagram for Alzheimer disease	
	3.1.3 Sequence Diagram for Alzheimer disease	
	3.1.4 Activity Diagram of Alzheimer disease	
	3.1.5 Component Diagram of Alzheimer disease	
	3.1.6 Deployment Diagram of Alzheimer disease	
	3.1.7 Package diagram for Alzheimer disease	
4	SYSTEM ARCHITECTURE	
	4.1 ARCHITECTURE DESCRIPTION	
5	SYSTEM IMPLEMENTATION	
	5.1 IMPLEMENTATION ALZHEIMER DISEASE PREDICTION	

	5.2 MODULES	
	5.2.1 DATA ACQUISTION	
	5.2.2 DATA PREPROCESSING	
	5.2.3 FEATURE EXTRACTION	
	5.2.4 MAGNETIC RESONANCE IMAGING CLASSIFICATION	
	5.2.5 CNN MODEL	
6	CODING AND SCREENSHOTS	
7	FUTURE ENHANCEMENTS	
8	REFERNCES	

CHAPTER 1

INTRODUCTION

Translational applications of computational neuro scientific approaches have been proven exceptionally beneficial in comprehensive mental health trials. This multidisciplinary field of study can help model the biological processes governing the healthy and diseased states of the human brain and map these processes into observable clinical presentations. In the past decade, the rapid increase in high-volume biomedical datasets (neuroimaging and related biological data), concurrent with the advances in machine learning (ML), has opened new avenues for the diagnosis and prognosis of neurodegenerative and neuropsychiatric disorders. The use of automatic systems capable of differentiating pathological cases from normal cases based on their magnetic resonance imaging (MRI) scans (i.e., no past hypotheses are needed) will contribute immensely to the initial diagnosis of AD. In this study, we review relevant studies that examine AD and use MRI data, ML and Deep Learning (DL) techniques with various AD datasets.

1.1OVERVIEW

Early detection of this disorder is being researched to slow down the abnormal degeneration of the brain, reduce medical care cost reduction, and ensure improved treatment. The recent failures in Alzheimer's disease research studies may suggest that early intervention and diagnosis could be crucial to the effectiveness of treatment. A wide variety of neuroimaging methods are becoming increasingly dependent on the diagnosis of dementia, and this is

reflected in many new diagnostic criteria. Neuroimaging increases diagnosis accuracy for various subtypes of dementia using machine learning. Specific pre-processing steps are needed to implement machine learning algorithms. Extraction and selection of features, reduction of feature dimensionality and classifier algorithm are all phases of the machine learning-based classification process. Such techniques need advanced knowledge and several optimization steps, which can be time consuming.

1.2 PROBLEM STATEMENT

Our study deals with automated Alzheimer disease detection and classification. Normally the anatomy of the brain is analyzed by MRI scans or CT scans. The aim of the paper is Alzheimer disease identification in brain MR images. The main reason for detection of AD is to provide aid to clinical diagnosis. The aim is to provide an algorithm that guarantees the presence of a dementia by combining several procedures to provide a fool proof method of AD detection in MRI brain images. The focus of this project is MR brain images AD extraction and its representation in simpler form such that it is understandable by everyone. The objective of this work is to bring some useful information in simpler form in front of the users, especially for the medical staff treating the patient. The aim of this work is to define an algorithm that will result in an extracted image of the tumor from the MR brain image. The resultant image will be able to provide information like size, dimension, and boundary provides us with information related to the ad that can prove useful for various cases, which will provide a better base for the staff to decide the curing procedure. Finally, we detect whether the given MR brain image has tumor or not using Convolution Neural Network.

1.3 EXISTING SYSTEM

The conventional MRI-based AD diagnosis methods usually partition

the entire MR image into multiple regions with different scales for better feature extraction of local abnormal brain structural changes. Based on the partition with different scales, most of the existing MRI-based studies can be roughly divided into three categories, including 1) voxel-level, 2) region-level, and 3) patch-level. In voxel-level methods, the tissue features (e.g., gray matter densities) extracted from MRI scans composes high-dimensional voxel-wise structural features for AD diagnosis. However, compared with the dimensionality of features, the number of training images for AD classification is too small, which often leads to the curse of dimensionality. To alleviate this problem, region-level methods are proposed to identify the AD patients from normal controls with the handcrafted features (e.g. gray matter, cerebrospinal fluid and cortical thickness) derived from segmented regions of interest (ROIs). However, these methods are resource-intensive for segmenting ROIs. In contrast, patch-level (an intermediate scale between voxel-level and region-level) feature representations are proposed for more effectively characterizing the local structural changes in MR images. Specifically, the centers of patches can be located by certain anatomical landmark detectors or statistics methods. However, how to combine the local patches into a global feature representation for the whole brain structure is still a challenge in patch-level methods.

1.4 PROPOSED SYSTEM

A relation-induced multi-modal shared representation learning framework for AD diagnosis correspond to the training stage, relational regularizes and test stage, respectively. At training stage, the framework first obtains shared representations by learning a bi-directional mapping between original space and shared space. For one thing, it is hope to learn latent discriminative representations from multi-modal data by introducing the projection matrix which conducts original-to-shared transformation. And for another, it is also expected the shared representations can preserve original

information as much as possible, and thus the reconstruction matrix is utilized to achieve shared-to-original conversion. Further the project shared the representations into target space by weight matrix, whose elements stand for the importance of the corresponding feature vectors for type's dataset in deep learning MRI images Mild Demented, Moderate Demented, Non-Demented, Very Mild Demented the class's final DenseNet169 algorithm classification AD diagnosis. Thus, representation learning (from original space to shared space) and classifier modeling (from shared space to label space) are integrated into the unified framework and can be optimized simultaneously. The Advantage of the proposed System the correction of image geometry enhances image information which makes the image more useful for any analysis process, Vanishing Gradient problem is alleviated by Dense-net, Feature Propagation is strengthened, Reusability of feature is enhanced, Number of the parameter is reduced, making use of all these we built this project to achieve high accuracy.

1.4.1 Materials and Methods

a. Dataset:

The datasets we used in this study are open-source and freely available on Kaggle. The data includes all four classes of mild, very mild, moderate and, non-dementia images from multiple domains. The dataset contains MRI images of all four classes. MRI and Clinical information on all the cases of collected dataset in a cross-sectional and longitudinal study design are collected. This dataset provides Morphometric data which gives the volumes of brain areas mostly affected by AD.

b. Libraries and System Configuration:

In order to perform this classification, you need the basic Data Scientist starter pack (sklearn, pandas, NumPy, matplotlib, seaborn) plus some specific libraries like TensorFlow or flow, Tfidf Transformer, feature extraction, linear

model, selection, pre-processing, accuracy score, train_test_split, Pipeline. Hardware system configuration are Processor – i3, i5, i7 Amd Processor, RAM -Above 4 Gb, Hard Disk - 260 GB

c. Data Acquisition:

The first step is to acquire images. To produce a classification model, the computer needs to learn by example. The computer needs to view many images to recognize an object. Other types of data, such as time series data and voice data, can also be used to train deep learning models. In the context of the work surveyed in this project, the relevant data required to detect Alzheimer disease will be images. The output of this step is images that will later be used to train the model.

d. Training

In this project, we are using transfer learning algorithms such as Densenet169, and vgg16.

Densenet169: It predicts the output of a categorical dependent variable.

VGG16: Transfer learning that process the image input and gives the output.

The data preprocessing was done using Jupyter Notebook and Desktop Application was Implemented using python IDLE. The programming language which was used is python and deep learning Sklearn was used to build the model using transfer learning algorithm like Densenet169, VGG16 which gives the results in the next stage.

e. Classification Metrics:

There are many classification approaches in the literature that can be used to extract relevant features with high discriminatory power. However, choosing the appropriate approach is a challenging step, requiring a careful study and good knowledge of the existing techniques. In the present study, popular and widely used transfer learning classification algorithms for

analyzing medical diagnoses in the context of a multiple classification problems, as case-control studies. Thereby, we rely on two main classification approaches from which we select popular transfer learning classification algorithms, appropriate for decision-making problems.

CHAPTER 2

LITERATURE REVIEW

Blennow.K, et.al (2021) [1] proposed Prediction of future Alzheimer's disease dementia using plasma phospho-tau combined with other accessible measures A combination of plasma phospho-tau (P-tau) and other accessible biomarkers might provide accurate prediction about the risk of developing Alzheimer's disease (AD) dementia. We examined this in participants with subjective cognitive decline and mild cognitive impairment from the Bio FINDER (n = 340) and Alzheimer's Disease Neuroimaging Initiative (ADNI) (n = 543) studies. AD-specific magnetic resonance imaging were examined using progression to AD as outcome The clinical predictions by memory clinic physicians had significantly lower accuracy (4-year AUC = 0.71). In summary, plasma P-tau, in combination with brief cognitive tests and APOE genotyping, might greatly improve the diagnostic prediction of AD and facilitate recruitment for AD trials.

Candice Ee Aang, et.al (2021) [2] Application of Artificial Intelligence techniques for the detection of Alzheimer's disease using structural MRI images Alzheimer's disease (AD) is an irreversible, progressive brain disorder that slowly destroys memory and thinking skills. It is one of the leading types of dementia for persons aged above 65 worldwide. In order to achieve accurate and timely diagnosis, and for detection of AD in its early stages, numerous Artificial Intelligence (AI) based Computer-aided Diagnostic (CAD)

approaches have been proposed using data from brain imaging. In this paper, we review the recent application of AI based CAD systems on AD and its stages, with a particular focus on the use of structural MRI. Summarize contributions from different research groups, critically discuss challenges involved and propose directions for future research. Ultimately, it would be ideal for development of a diagnostic framework that could be applicable to not only AD, but to different types of dementia as well in the future.

Chandran Venkatesan, et.al (2021) [3] proposed A Deep Learning Model for Early Diagnosis of Alzheimer Diseases and Dementia from MR Images Alzheimer's disease (AD) is the most common cause of dementia globally. By considering four stages of dementia and conducting a particular diagnosis, the proposed model generates high-resolution disease probability maps from the local brain structure to a multilayer perceptron and provides accurate, intuitive visualizations of individual Alzheimer's disease risk. To avoid the problem of class imbalance, the samples should be evenly distributed among the classes. The obtained MRI image dataset from Kaggle has a major class imbalance problem. A Dementia Network (DEMNET) is proposed to detect the dementia stages from MRI. The DEMNET achieves an accuracy of 95.23%, Area under Curve (AUC) of 97% and Cohen's Kappa value of 0.93 from the Kaggle dataset, which is superior to existing methods. We also used the Alzheimer's disease Neuroimaging Initiative (ADNI) dataset to predict AD classes in order to assess the efficacy of the proposed model.

Jhansi, et.al (2021) [4] proposed Alzheimer Disease Detection using Correlation based Ensemble Feature Selection and Multi Support Vector Machine In recent decades, machine learning techniques have been playing a crucial role in the field of computer aided diagnosis. Initially, an adaptive

histogram equalization and region growing are employed on the collected brain scans for contrast improvement and skull removal. Next, Fuzzy C Means (FCM) clustering algorithm is applied in the enhanced brain scans to segment tissues like White Matter (WM), Cerebral Spinal Fluid (CSF), and Grey Matter (GM). In addition, feature extraction is accomplished in the segmented brain tissues using Gabor and local directional pattern variance features. In order to decrease the dimension of the extracted feature vectors, the correlation based on ensemble feature selection algorithm was proposed. The Alzheimer disease using machine learning algorithms is successfully implemented and gives greater prediction accuracy results.

Yamini T, et.al (2022) [5] proposed Alzheimer's disease is the most common form of dementia affecting the brain's parts. A broad term used to describe illnesses and conditions that causes a deterioration in memory, language, and other cognitive abilities severe enough to interfere with daily life is "dementia". According to estimates, this disease affects 6.2 million Americans and 5 million people in India aged 65 and older. In 2019, the most recent year for which data are available, official death certificates reported 121,499 deaths from AD, making Alzheimer's the "sixth leading cause of death in the country and the fifth leading cause of death for people 65 and older". In this paper, we suggest several machine Learning algorithms like Decision trees, SVM, Logistic regression, and Naive Bayes identify AD at an early stage. The Alzheimer's Disease Neuroimaging Initiative (ADNI) and the Open Access Series of Imaging Investigations (OASIS) provide data sets which are used to detect the disease in its early stage. The datasets consist of longitudinal MRI data (age, gender, mini mental status, CDR) By taking into account many factors in each method, such as precision, F1 Score, Recall, and specificity are calculated. The results obtained 93.7% of maximum accuracy for the Decision Tree Algorithm.

Kavitha C, et.al (2022) [6] proposed Alzheimer's disease (AD) is the leading cause of dementia in older adults. There is currently a lot of interest in applying machine learning to find out metabolic diseases like Alzheimer's and Diabetes that affect a large population of people around the world. Their incidence rates are increasing at an alarming rate every year. In Alzheimer's disease, the brain is affected by neurodegenerative changes. As our aging population increases, more and more individuals, their families, and healthcare will experience diseases that affect memory and functioning. These effects will be profound on the social, financial, and economic fronts. In its early stages, Alzheimer's disease is hard to predict. A treatment given at an early stage of AD is more effective, and it causes fewer minor damage than a treatment done at a later stage. Several techniques such as Decision Tree, Random Forest, Support Vector Machine, Gradient Boosting, and Voting classifiers have been employed to identify the best parameters for Alzheimer's disease prediction. Predictions of Alzheimer's disease are based on Open Access Series of Imaging Studies (OASIS) data, and performance is measured with parameters like Precision, Recall, Accuracy, and F1-score for ML models. The proposed classification scheme can be used by clinicians to make diagnoses of these diseases. It is highly beneficial to lower annual mortality rates of Alzheimer's disease in early diagnosis with these ML algorithms. The proposed work shows better results with the best validation average accuracy of 83% on the test data of AD. This test accuracy score is significantly higher in comparison with existing works.

Vijeeta patil, et.al (2022) [7] proposed a comprehensive review on Alzheimer's disease (AD) is carried out, and an exploration of the two machine learning (ML) methods that help to identify the disease in its initial stages.

Alzheimer's disease is a neurocognitive disorder occurring in people in their early onset. This disease causes the person to suffer from memory loss, unusual behavior, and language problems. Early detection is essential for developing more advanced treatments for AD. Machine learning (ML), a subfield of Artificial Intelligence (AI), uses various probabilistic and optimization techniques to help computers learn from huge and complicated data sets. To diagnose AD in its early stages, researchers generally use machine learning. The survey provides a broad overview of current research in this field and analyses the classification methods used by researchers working with ADNI data sets. It discusses essential research topics such as the data sets used, the evaluation measures employed, and the machine learning methods used. Our presentation suggests a model that helps better understand current work and highlights the challenges and opportunities for innovative and useful research. The study shows which machine learning method holds best for the ADNI data set. Therefore, the focus is given to two methods: the 18-layer convolutional network and the 3D convolutional network. Hence, CNNs with multi-layered fetch more accurate results as compared to 3D CNN. The work also contributes to the use of the ADNI data set, where the classification of training and testing samples is divided with such a number that brings the highest accuracy achieved with 18-layer CNN. The work concentrates on the early prediction of Alzheimer's disease with machine learning methods. Thus, the accuracy achieved is 98% for 18-layer CNN.

Fan Wu, et.al (2022) [8] proposed Alzheimer's disease (AD), the most familiar type of dementia, is a severe concern in modern healthcare. Around 5.5 million people aged 65 and above have AD, and it is the sixth leading cause of mortality in the US. AD is an irreversible, degenerative brain disorder characterized by a loss of cognitive function and has no proven cure. Deep

learning techniques have gained popularity in recent years, particularly in the domains of natural language processing and computer vision. Since 2014, these techniques have begun to achieve substantial consideration in AD diagnosis research, and the number of papers published in this arena is rising drastically. Deep learning techniques have been reported to be more accurate for AD diagnosis in comparison to conventional machine learning models. Motivated to explore the potential of deep learning in AD diagnosis, this study reviews the current state-of-the-art in AD diagnosis using deep learning. We summarize the most recent trends and findings using a thorough literature review. The study also explores the different biomarkers and datasets for AD diagnosis. Even though deep learning has shown promise in AD diagnosis, there are still several challenges that need to be addressed.

Mir Jafkul Alam, et.al (2023) [9] proposed Alzheimer's disease (AD) is one of the leading causes of dementia among older people. In addition, a considerable portion of the world's population suffers from metabolic problems, such as Alzheimer's disease and diabetes. Alzheimer's disease affects the brain in a degenerative manner. As the elderly population grows, this illness can cause more people to become inactive by impairing their memory and physical functionality. This might impact their family members and the financial, economic, and social spheres. Researchers have recently investigated different machine learning and deep learning approaches to detect such diseases at an earlier stage. Early diagnosis and treatment of AD help patients to recover from it successfully and with the least harm. This paper proposes a machine learning model that comprises GaussianNB, Decision Tree, Random Forest, XGBoost, Voting Classifier, and GradientBoost to predict Alzheimer's disease. The model is trained using the open access series of imaging studies (OASIS) dataset to evaluate the performance in terms of accuracy, precision, recall, and

F1 score. Our findings showed that the voting classifier attained the highest validation accuracy of 96% for the AD dataset. Therefore, ML algorithms have the potential to drastically lower Alzheimer's disease annual mortality rates through accurate detection.

Prasun Chakarabart, et.al (2023) [10] proposed Alzheimer's disease (AD) is a brain-related disease in which the condition of the patient gets worse with time. AD is not a curable disease by any medication. It is impossible to halt the death of brain cells, but with the help of medication, the effects of AD can be delayed. As not all MCI patients will suffer from AD, it is required to accurately diagnose whether a mild cognitive impaired (MCI) patient will convert to AD (namely MCI converter MCI-C) or not (namely MCI non-converter MCI-NC), during early diagnosis. There are two modalities, positron emission tomography (PET) and magnetic resonance image (MRI), used by a physician for the diagnosis of Alzheimer's disease. Machine learning and deep learning perform exceptionally well in the field of computer vision where there is a requirement to extract information from high-dimensional data. Researchers use deep learning models in the field of medicine for diagnosis, prognosis, and even to predict the future health of the patient under medication. This study is a systematic review of publications using machine learning and deep learning methods for early classification of normal cognitive (NC) and Alzheimer's disease (AD). This study is an effort to provide the details of the two most commonly used modalities PET and MRI for the identification of AD, and to evaluate the performance of both modalities while working with different classifiers.

Javed Rahebi, et.al (2023) proposed Alzheimer's is a neurodegenerative disorder affecting the central nervous system and cognitive processes, explicitly impairing detailed mental analysis. Throughout this condition, the

affected individual's cognitive abilities to process and analyze information gradually deteriorate, resulting in mental decline. In recent years, there has been a notable increase in endeavors aimed at identifying Alzheimer's disease and addressing its progression. Research studies have demonstrated the significant involvement of genetic factors, stress, and nutrition in developing this condition. The utilization of computer-aided analysis models based on machine learning and artificial intelligence has the potential to significantly enhance the exploration of various neuroimaging methods and non-image biomarkers. This study conducts a comparative assessment of more than 80 publications that have been published since 2017. Alzheimer's disease detection is facilitated by utilizing fundamental machine learning architectures such as support vector machines, decision trees, and ensemble models. Furthermore, around 50 papers that utilized a specific architectural or design approach concerning Alzheimer's disease were examined. The body of literature under consideration has been categorized and elucidated through the utilization of data-related, methodology-related, and medical-fostering components to illustrate the underlying challenges. The conclusion section of our study encompasses a discussion of prospective avenues for further investigation and furnishes recommendations for future research activities on the diagnosis of Alzheimer's disease.

CHAPTER 3

SYSTEM DESIGN

In this chapter, the various UML diagrams for the Alzheimer Disease Prediction using CNN Model is represented and the various functionalities are explained.

3.1 UNIFIED MODELING LANGUAGE

Unified Modelling language (UML) is a standardized modelling language enabling developers to specify, visualize, construct and document artifacts of a software system. Thus, UML makes these artifacts scalable, secure and robust in execution. It uses graphic notation to create visual models of software systems. UML is designed to enable users to develop an expressive, ready to use visual modelling language. In addition, it supports high level development concepts such as frameworks, patterns and collaborations. Some of the UML diagrams are discussed. UML helps software engineers, businessmen and system architects with modelling, design and analysis. UML is linked with object-oriented design and analysis. It makes use of the elements and forms associations between them to form diagrams. It is not a programming language; it is rather a visual language.

3.1.1 Use Case Diagram of Alzheimer disease prediction

Use case diagrams are considered for high level requirement analysis of a system. So, when the requirements of a system are analyzed, the functionalities are captured in use cases. So, it can be said that use cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors.

Actors can be defined as something that interacts with the system. The actors can be human user, some internal applications or may be some external applications. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly

design requirements.

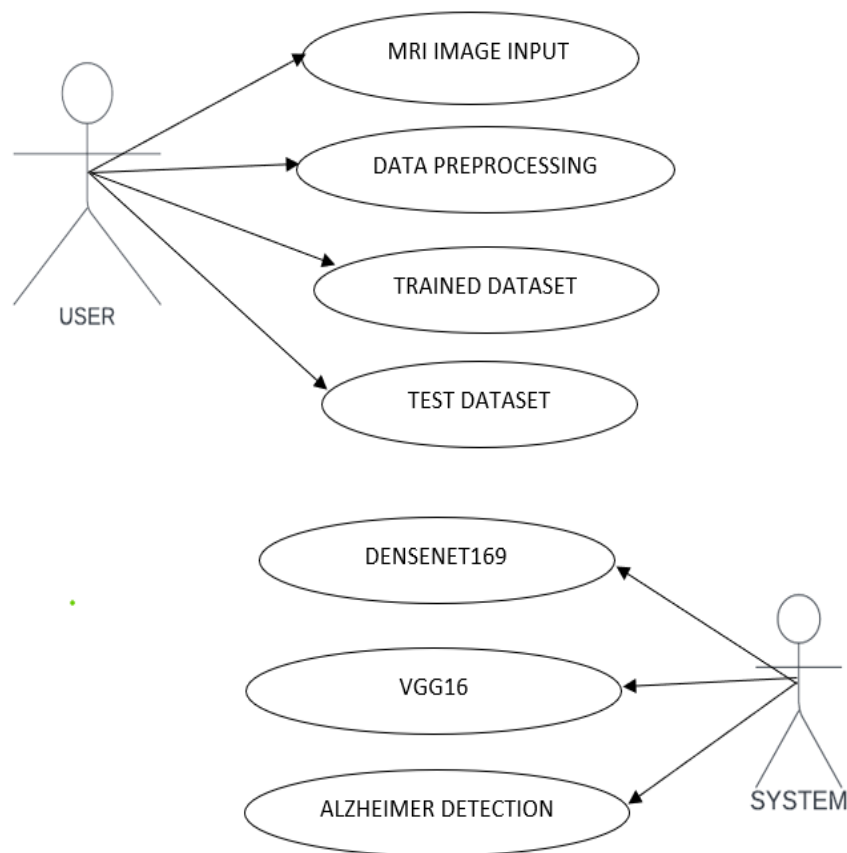


Figure 3.1 Use case diagram of Alzheimer Disease Prediction

Figure 3.1 shows that the functionalities are to be represented as a use case in the representation. Each and every use case is a function in which the user or the server can have the access on it. The names of the use cases are given in such a way that the functionalities are preformed, because the main purpose of the functionalities is to identify the requirements. To add some extra notes that should be clarified to the user, the notes kind of structure is added to the use case diagram. Only the main relationships between the actors and the functionalities are shown because all the representation may collapse the diagram.

3.1.2 Class Diagram of Alzheimer Disease Prediction

Figure 3.2 shows that class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So, a collection of class diagrams represent the whole system. The name of the class diagram should be meaningful to describe the aspect of the system.

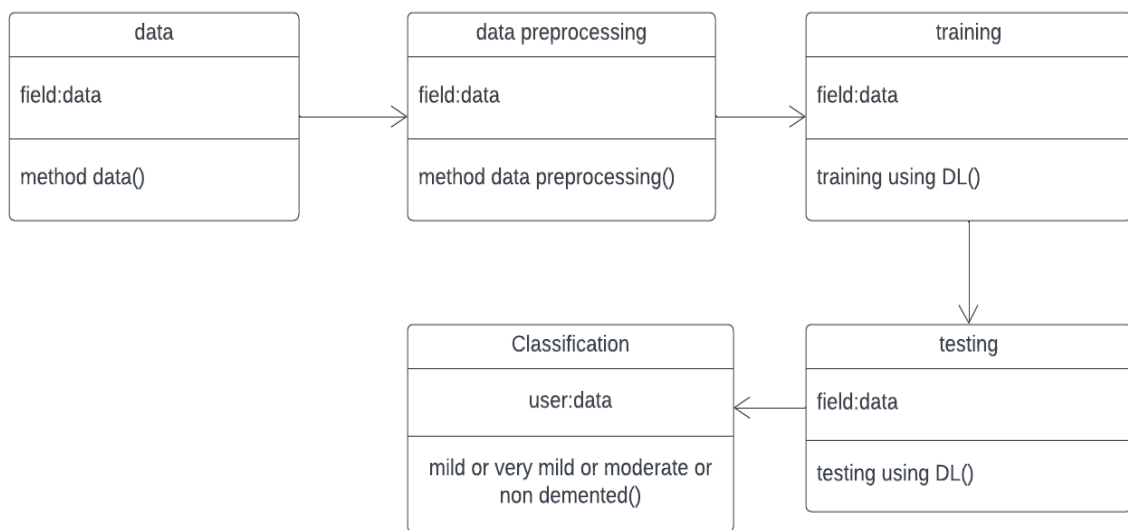


Figure 3.2 Class Diagram of Alzheimer Disease Prediction

Figure 3.2 shows the attributes, classes, functions, and relationships to give an overview of the software system. They are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system. In a class diagram, the classes are arranged in groups that share common characteristics. A class diagram resembles a flowchart in which classes are portrayed as boxes, each box having three rectangles inside which includes the name, attributes and methods used in each class.

3.1.3 Sequence Diagram of Alzheimer Disease Prediction

A sequence diagram represents the flow of messages in a system. It helps in envisioning several dynamic scenarios. It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality. It portrays the communication between any two lifelines as a time-oriented sequence of events, such that these lifelines took part at the runtime.

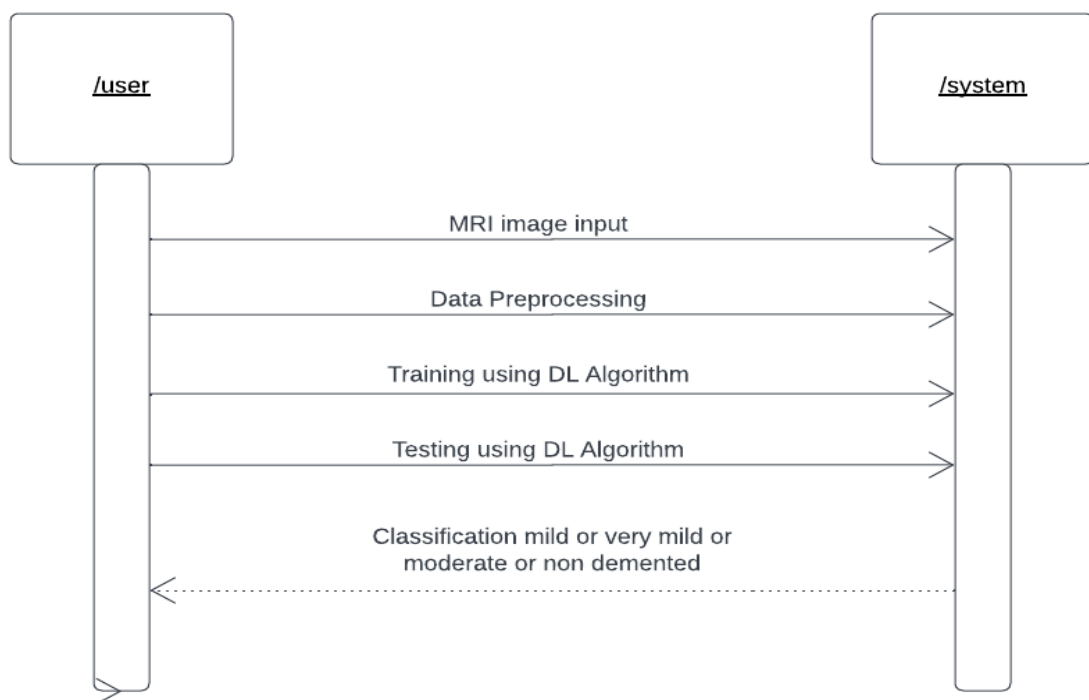


Figure 3.3 Sequence diagram of Alzheimer Disease Prediction

Figure 3.3 represents the sequence diagram of a Alzheimer Disease Prediction. Here, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line. It incorporates the iterations as well as branching. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. A sequence diagram shows the sequence of

messages passed between objects. Sequence diagrams can also show the control structures between objects. The communication between the user and the bank server are represented by messages passed between them. The sequence diagram shows the objects and the messages between the objects.

3.1.4 Activity Diagram of Alzheimer Disease Prediction

An activity diagram is a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

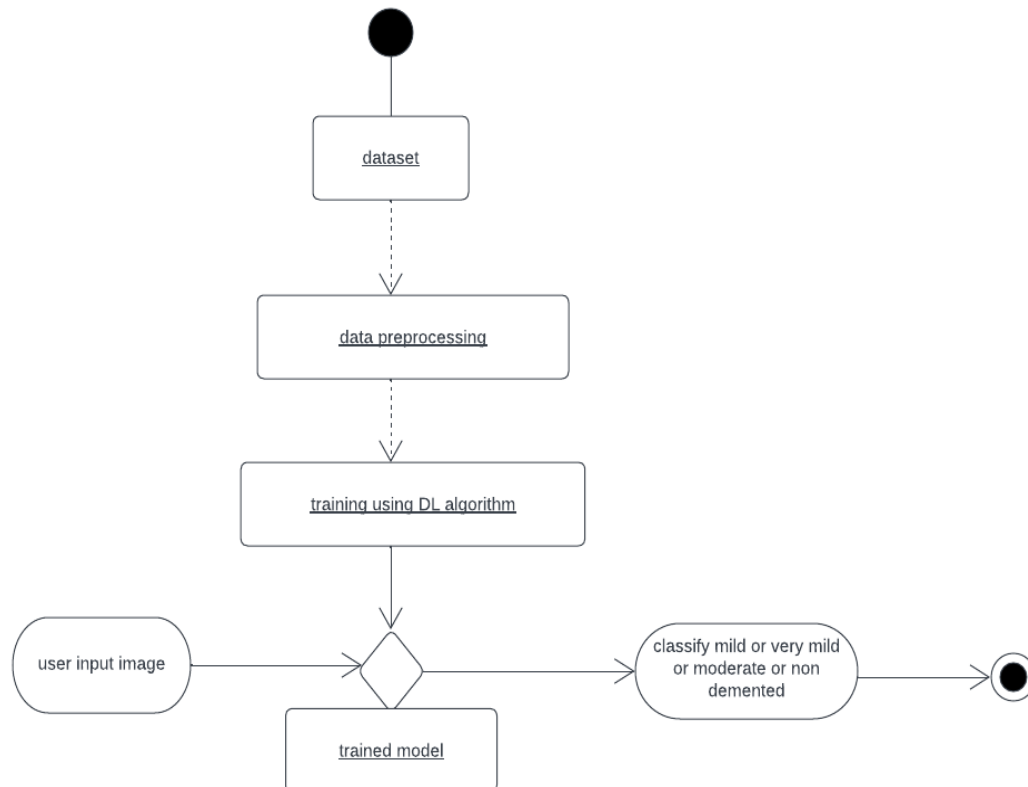


Figure 3.4 Activity Diagram of Alzheimer Disease Prediction

Figure 3.4 represents the activity diagram of the Alzheimer Disease Prediction system. The figure represents five activities that take place sequentially. Activities are a network of nodes connected by edges. There can

be action nodes, control nodes, or object nodes. Action nodes represent some action. Control nodes represent the control flow of an activity. Object nodes are used to describe objects used inside an activity. Edges are used to show a path or a flow of execution. Activities start at an initial node and terminate at a final node.

3.1.5 Component Diagram of Alzheimer Disease Prediction

Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that are often used to model the static implementation view of a system.

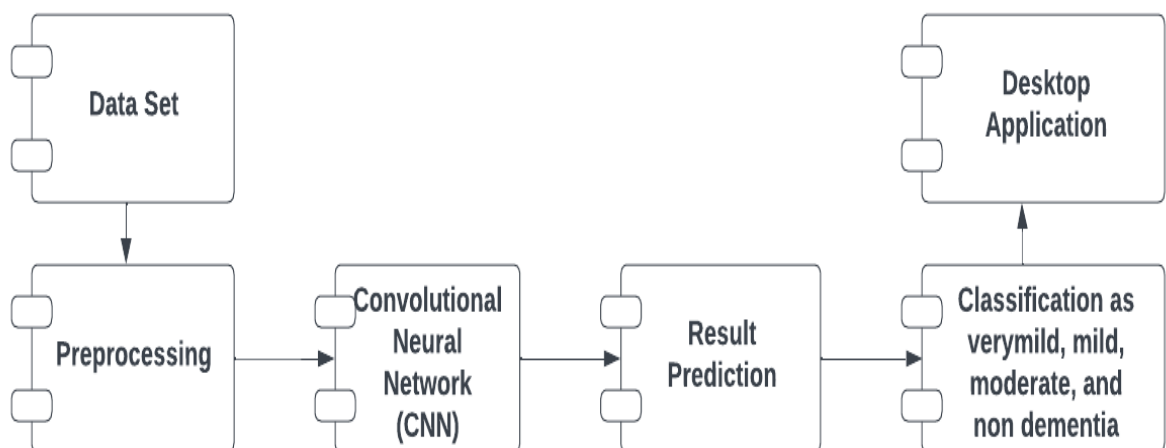


Figure 3.5 Component Diagram of Alzheimer Disease Prediction

Figure 3.5 depicts the component diagram of the Alzheimer Disease Prediction system. A component diagram breaks down the actual system under development into various high levels of functionality. Each component is responsible for one clear aim within the entire system and only interacts with

other essential elements on a need-to-know basis. Since it is a special kind of a UML diagram, it holds distinct purposes. It describes all the individual components that are used to make the functionalities, but not the functionalities of the system. It visualizes the physical components inside the system. The components can be a library, packages, files, etc. The component diagram also describes the static view of a system, which includes the organization of components at a particular instant. The collection of component diagrams represents a whole system. The component diagram also describes the static view of a system, which includes the organization of components at a particular instant. The collection of component diagrams represents a whole system.

3.1.6 Deployment Diagram of Alzheimer Disease Prediction

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships. It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

Deployment diagram represents the deployment view of a system. It is related to the component diagram because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application. Deployment diagrams are useful for system engineers. An efficient deployment diagram is very important as it controls parameters such as performance, scalability, maintainability, portability.

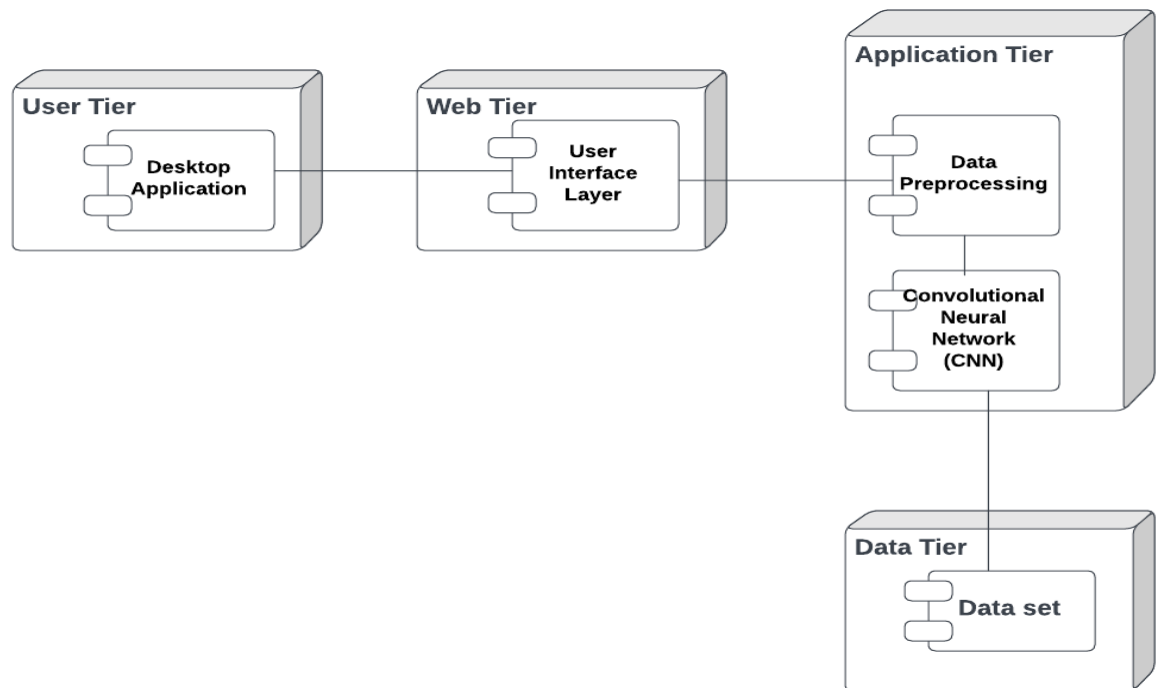


Figure 3.6 Component Diagram of Alzheimer Disease Prediction

Figure 3.6 depicts the deployment diagram of the Alzheimer Disease Prediction system. The main purpose of the deployment diagram is to represent how software is installed on the hardware component. It depicts in what manner a software interacts with hardware to perform its execution.

3.1.7 Package Diagram of Alzheimer Disease Prediction

The package diagram is a kind of structural diagram that shows the arrangement and organization of model elements in a middle to large scale project. Package diagram can show both structure and dependencies between subsystems or modules, showing different views of a system. A package is a collection of logically related UML elements. Packages can be built to represent either physical or logical relationships. When choosing to include classes in specific packages, it is useful to assign the classes with the same inheritance hierarchy to the same package

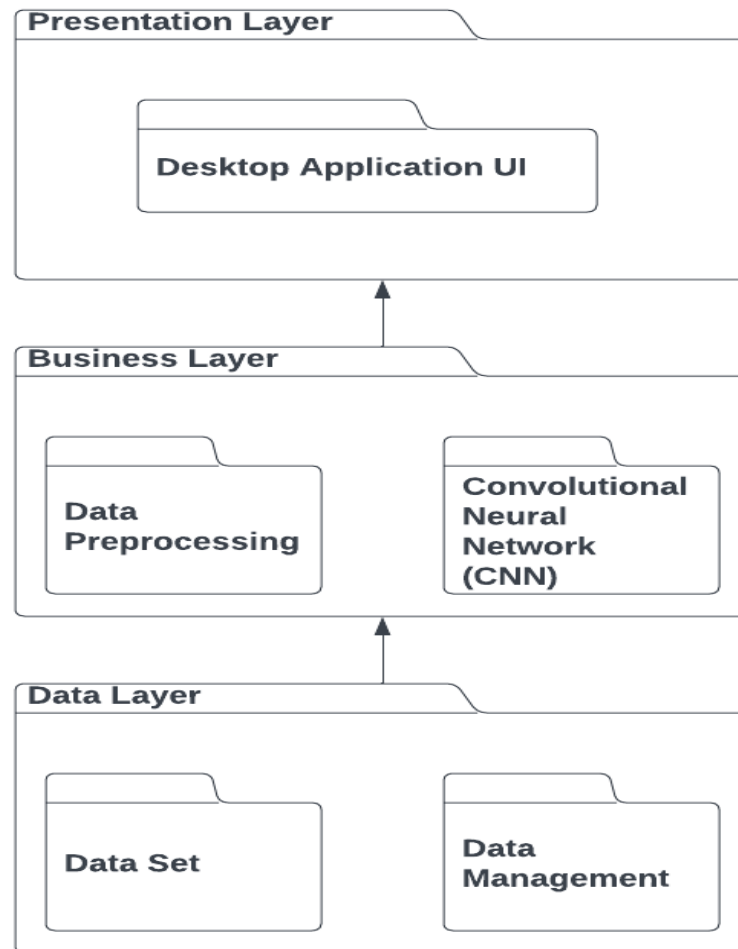


Figure 3.6 Package Diagram of Alzheimer Disease Prediction

Figure 3.6 represent the package diagram of the Alzheimer Disease Prediction system. Package diagram follows the hierarchical structure of nested packages. It is used to simplify complex class diagrams, allowing grouping of classes into packages. A package is a collection of logically related UML elements. Packages can be built to represent either physical or logical relationships. When choosing to include classes in specific packages, it is useful to assign the classes with the same inheritance hierarchy to the same package. Package diagram can show both structure and dependencies between subsystems or modules, showing different views of a system.

CHAPTER 4

SYSTEM ARCHITECTURE

In this chapter, the System Architecture for the Alzheimer Disease Prediction using CNN Model is represented and the modules are explained.

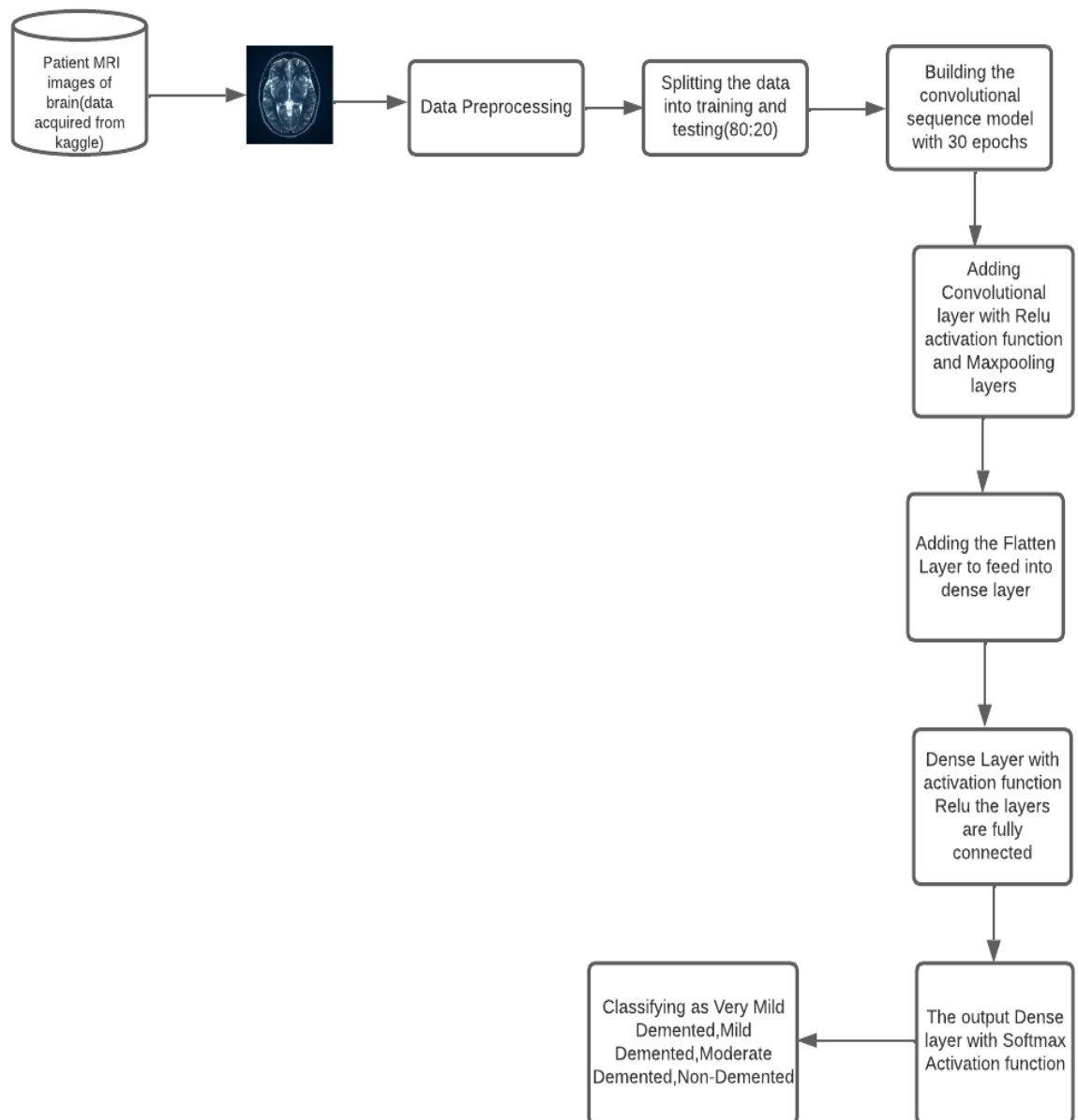


Fig 4.1 System Architecture Diagram

4.1 ARCHITECTURE DESCRIPTION

Here we explore DL Algorithms to identify Alzheimer disease in the Medical Industry in this model. It uses two deep learning Algorithms for detecting the convolutional channel feature (CCF). However, in this model we choose the convolutional neural network (CNN) model and its layers to determine the mild, very mild, moderate and non demented stage of the disease. We collect datasets from an open-source website called Kaggle. The dataset holds MRI images of the AD affected person. There were 400 images in total where 200 images of each category is taken and analyzed. Resizing the MRI image into 200X200.

As per the Figure 4.1, after the dataset collection and pre-processing methods, we apply the following machine and ensemble learning algorithm. The Dense Net and VGG16 are the transfer learning method is a neural network for classification and feature learning. It can be used with one or a multilayer of unseen nodes. Parameters of unseen nodes are tuned. We are building a convolutional sequence model of 30 epochs. Where Convolutional sequence is approach to sequence-to-sequence learning that maps an input sequence to a variable length output sequence via recurrent neural networks. Then subsequently convolutional layer, Flatten layer, and Dense Layer. The convolutional layer is a set of filters, Parameters of which are to be Learned throughout the training. Flatten Layer collapses the facial dimension of the input into the channel dimension. Dense layer helps in changing the dimensionality of the output from the preceding layer. In this process relu and SoftMax Activation function are used. Relu activation function is most commonly used activation function in neural networks, especially in Convolutional Neural Networks (CNNs) & Multilayer perceptron

CHAPTER 5

SYSTEM IMPLEMENTATION

In this chapter, the System Implementation for the Alzheimer Disease Prediction using Deep Learning is explained in detail.

5.1 IMPLEMENTATION OF ALZHEIMER DISEASE PREDICTION

The project is implemented in Jupyter Notebook. Here, the various functionalities required for the application are implemented by coding them in Python.

-Dataset source – Kaggle

-Data-(Image)

5.2 MODULES

5.2.1 DATA ACQUISITION:

The first step is to acquire images. To produce a classification model, the computer needs to learn by example. The computer needs to view many images to recognize an object. Other types of data, such as time series data and voice data, can also be used to train deep learning models. In the context of the work surveyed in this paper, the relevant data required to detect Alzheimer disease will be images. The output of this step is images that will later be used to train the model.

5.2.2 DATA PREPROCESSING:

An image classification task determines the category of a given input MRI image. It is a basic task in high-level image understanding and can be divided into binary- and multi classification tasks. After multiple convolution-and-pooling operations via a CNN, an image is classified in the output layer following the requirements. Activation function of the output layer is the only difference between binary and multi classification tasks. An image classification task for MRI image analysis easily identified and then necessary

actions can be taken to which type of dementia, is a high performance in natural image classification, including Convolution neural network (CNNs) can be used in JPG/PNG image classification.

5.2.3 FEATURE EXTRACTION

In this module, we are performing some more operation on segmented image. In this module we will perform feature extraction operation to get all detailed information about brain image. Feature Extraction and reduction has been playing a vital role for tumor region into their relevant categories in the field of computer vision and machine learning. The major issue behind feature extraction is to compute the most active or robust features for classification, which produced an efficient performance. The Feature extraction is used related to dimensionality reduction.

5.2.4 MAGNETIC RESONANCE IMAGING CLASSIFICATION:

This imaging technique utilizes radio waves and magnetic fields to generate high-quality and high-resolution 2D and 3D images of brain structures. No harmful radiations from X-rays or radioactive tracers are generated. The most commonly used MRI for AD cases is the structural MRI, which measures brain volumes in vivo to detect brain degeneration (loss of tissue, cells, neurons, etc.). Brain degeneration is an inevitable progressive component of AD. A structural MRI used to detect brain atrophy. Alternatively, Functional Magnetic Resonance Imaging (fMRI), a widely used method to measure human primary visual cortex and detect brain topography. fMRI provides useful information and data about the human brain's activity, i.e., how the brain functions. fMRI methods, such as brain imaging based on arterial Blood Oxygenation Level Dependent (BOLD) contrasts and spin-labelling, are sensitive to the cerebral metabolic rate of oxygen consumption and cerebral blood flow (CBF).

5.2.5 CNN MODEL

Image classification involves the extraction of features from the image to observe some patterns in the dataset. Using an ANN for the purpose of image classification would end up being very costly in terms of computation since the trainable parameters become extremely large. For example, if we have a 50 X 50 image of a cat, and we want to train our traditional ANN on that image to classify it into a dog or a cat the trainable parameters become – $(50*50) * 100$ image pixels multiplied by hidden layer + 100 bias + $2 * 100$ output neurons + 2 bias = 2,50,30.

Examples of different filters and their effects Filters help us exploit the spatial locality of a particular image by enforcing a local connectivity pattern between neurons. Convolution basically means a pointwise multiplication of two functions to produce a third function. Here one function is our image pixels matrix and another is our filter. We slide the filter over the image and get the dot product of the two matrices. The resulting matrix is called an “Activation Map” or “Feature Map”.

Step 1: Choose a Dataset

Choose a dataset of your interest or you can also create your own image dataset for solving your own image classification problem. An easy place to choose a dataset is on [kaggle.com](https://www.kaggle.com). The dataset I’m going with can be found [here](#). This dataset contains 12,500 augmented images of blood cells (JPEG) with accompanying cell type labels (CSV). There are approximately 3,000 images for each of 4 different cell types grouped into 4 different folders (according to cell type). The cell types are Eosinophil, Lymphocyte, Monocyte, and Neutrophil. Here are all the libraries that we would require and the code for importing them.

Step 2: Prepare Dataset for Training

Preparing our dataset for training will involve assigning paths and creating categories(labels), resizing our images. Resizing images into 200 X

200.

Step 3: Create Training Data

Training is an array that will contain image pixel values and the index at which the image in the CATEGORIES list.

Step 4: Shuffle the Dataset

Step 5: Assigning Labels and Features

This shape of both the lists will be used in Classification using the NEURAL NETWORKS.

Step 6: Normalizing X and converting labels to categorical data

Step 7: Split X and Y for use in CNN

Step 8: Define, compile and train the CNN Model

Step 9: Accuracy and Score of models

```
function XCOMPRESSCU(*pCurCU)
M ← FastCUMope (PO, QP)
if M 4 SPLIT, then
C2n ←CHECKINTRA (pCurCU)
else
C2n ← ∞
end if
if M! = HOMO and Dcur < Dmax then
Cn ← 0
for i = 0 to 3 do
pSubCUi ← pointer to SubCUi
CN ← CN+ XCompressCU(pSubCUi) end for
else
CN ← ∞
end if
CHECKBESTMODE (C2N, CN)
end function
```

a. Dense Networks (Dense Net)

Very deep neural networks are hard to train as they are more prone to vanishing or exploding gradients. To solve this problem, the activation unit from a layer could be fed directly to a deeper layer of the network, which is termed as a skip connection. This forms the basis of Dense networks or Dense

Net This post will introduce the basics the residual networks before implementing one in Keras.

Dense Block

A building block of a Dense Net is called a Dense block or identity block. A residual block is simply when the activation of a layer is fast-forwarded to a deeper layer in the neural network.

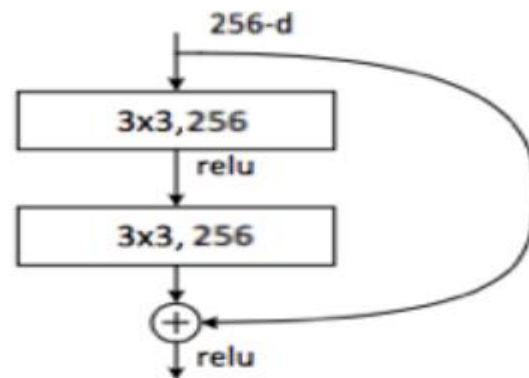


Figure 5.1 Dense Block Diagram

Figure 5.1 represents the activation from a previous layer is being added to the activation of a deeper layer in the network. This simple tweak allows training much deeper neural networks. In theory, the training error should monotonically decrease as more layers are added to a neural network. In practice however, for a traditional neural network, it will reach a point where the training error will start increasing. Dense Nets do not suffer from this problem. The training error will keep decreasing as more layers are added to the network. In fact, Dense Nets have made it possible to train networks with more than 100 layers, even reaching 1000 layers. Building a Dense Net for image classification. Now, let's build a Dense Net with 50 layers for image classification using Keras. Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. In this case, we will use TensorFlow as the backend.

- Step 1: Define the identity block
- Step 2: Convolution block
- Step 3: Build the model
- Step 4: Training
- Step 5: Print the model summary

Algorithm 2. Pseudo code of the used preprocessing method.

Input: The raw 1D sensor signal (S) with size of 5625
Output: Graylevel image (Im) with size of 125 x 45
<pre> 1: <i>count</i> = 1; 2: for $i=1$ to 125 do 3: for $j=1$ to 45 do 4: $Im(i,j) = S(count)$; 5: $count = count + 1$; 6: end for j 7: end for i 8: Normalize Im by using min-max normalization.</pre>

Figure 5.2 Pseudocode of the preprocessing model

b. VGG16 Implementation

VGG16 is a convolution neural net (CNN) architecture which was used to win ILSVR(ImageNet) competition in 2014. It is considered to be one of the excellent vision model architectures till date. Most unique thing about VGG16 is that instead of having a large number of hyper-parameters they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC (fully connected layers) followed by a SoftMax for output. The 16 in VGG16 refers to it has 16 layers that have

weights. This network is a pretty large network and it has about 138 million (approx.) parameters.

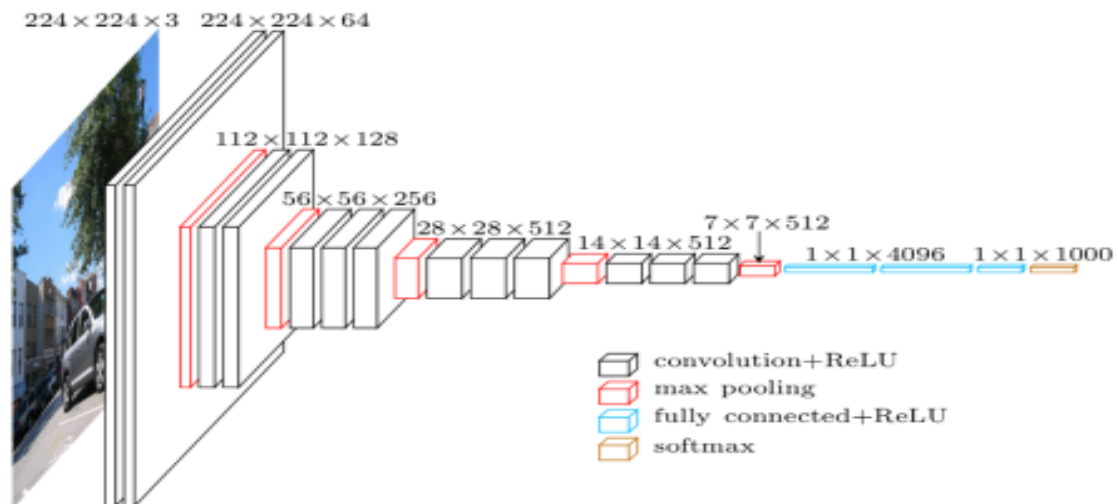


Figure 5.3 VGG16 Implementation Diagram

Here we first import all the libraries which will be need to implement VGG16. We will be using Sequential method as we are creating a sequential model. Sequential model means that all the layers of the model will be arranged in sequence. Here we have imported Image Data Generator from Keras preprocessing. The objective of Image Data Generator is to import data with labels easily into the model. It is a very useful class as it has many functions to rescale, rotate, zoom, flip etc. The most useful thing about this class is that it does not affect the data stored on the disk. This class alters the data on the go while passing it to the model.

CHAPTER 6

CODING AND SCREENSHOTS

6.1 Sample Code

training.ipynb

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import skimage.io
import os
import tqdm
import glob
import tensorflow
from tqdm import tqdm
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
from skimage.io import imread, imshow
from skimage.transform import resize
#from skimage.color import grey2rgb
from skimage.color import rgb2gray
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import InputLayer, BatchNormalization,
Dropout, Flatten, Dense, Activation, MaxPool2D, Conv2D
from keras.models import model_from_json
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras.applications.densenet import DenseNet169
from tensorflow.keras.preprocessing.image import load_img, img_to_array
batch_size = 32
# All images will be rescaled by 1./255
train_datagen = ImageDataGenerator(rescale=1/255)
```

```

# Flow training images in batches of 128 using train_datagen generator
train_generator = train_datagen.flow_from_directory(
    'alzheimer/training', # This is the source directory for training images
    target_size=(200, 200), # All images will be resized to 200 x 200
    batch_size=batch_size,
    # Specify the classes explicitly
    classes
    ['MildDemented','NonDemented','VeryMildDementedd','ModerateDemented']
    ,
    # Since we use categorical_crossentropy loss, we need categorical labels
    class_mode='categorical')
valid_datagen = ImageDataGenerator(rescale = 1./255,
                                    validation_split = 0.2)
test_datagen = ImageDataGenerator(rescale = 1./255)
train_dataset=train_datagen.flow_from_directory(directory=r'C:/Users/admin/
CODE-ALZHEIMERDISEASEPREDICTION/alzheimer/training',
                                                target_size = (224,224),
                                                class_mode = 'categorical',
                                                subset = 'training',
                                                batch_size = 128)
valid_dataset=valid_datagen.flow_from_directory(directory=r'C:/Users/admin
/CODE-ALZHEIMERDISEASE PREDICTION/alzheimer/training',
                                                target_size = (224,224),
                                                class_mode = 'categorical',
                                                subset = 'validation',
                                                batch_size = 128)
fig, ax = plt.subplots(nrows = 1, ncols = 5, figsize=(20,20))
for i in tqdm(range(0,5)):

```



```

rand1 = np.random.randint(len(train_dataset))
rand2 = np.random.randint(50)
ax[i].imshow(train_dataset[rand1][0][rand2])
ax[i].axis('off')
a = train_dataset[rand1][1][rand2]
if a[0] == 1:
    ax[i].set_title('Mild Dementia')
elif a[1] == 1:
    ax[i].set_title('Moderate Dementia')
elif a[2] == 1:
    ax[i].set_title('Non Demetia')
elif a[3] == 1:
    ax[i].set_title('Very Mild Dementia')
import tensorflow as tf
model = tf.keras.models.Sequential([
    # Note the input shape is the desired size of the image 200x 200 with 3 bytes
    color
    # The first convolution
    tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(200, 200,
3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    # The second convolution
    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The third convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The fourth convolution

```

```

tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
tf.keras.layers.MaxPooling2D(2,2),
# The fifth convolution
tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
tf.keras.layers.MaxPooling2D(2,2),
# Flatten the results to feed into a dense layer
tf.keras.layers.Flatten(),
# 128 neuron in the fully-connected layer
tf.keras.layers.Dense(128, activation='relu'),
# 5 output neurons for 5 classes with the softmax activation
tf.keras.layers.Dense(4, activation='softmax'))])
model.summary()
from tensorflow.keras.optimizers import RMSprop
model.compile(loss='categorical_crossentropy',
              optimizer=RMSprop(lr=0.001),
              metrics=['acc'])
total_sample=train_generator.n
n_epochs = 30
history = model.fit_generator(
    train_generator,
    steps_per_epoch=int(total_sample/batch_size),
    epochs=n_epochs, verbose=1)
model.save('model2.h5')
# Test Case 1: Non-Dementia
import numpy as np
from keras_preprocessing import image
import easygui
dic = test_dataset.class_indices

```

```

idc = {k:v for v, k in dic.items()}
img=load_img(r'C:/Users/admin/MultiClass-Image-Classification-
master/alzheimer/training/NonDemented/nonDem
18.jpg', target_size = (224,224,3))
img = img_to_array(img)
img = img/255
imshow(img)
plt.axis('off')
img = np.expand_dims(img,axis=0)
test_image=image.load_img(r'C:/Users/admin/MultiClass-Image-
Classification-
master/alzheimer/training/NonDemented/nonDem18.jpg',target_size=(200,20
0))
test_image = np.expand_dims(test_image, axis=0)
result = model.predict(test_image)
if result[0][1] == 1:
    prediction = "NonDemented"
elif result[0][0] == 1:
    prediction = "MildDemented"
elif result[0][2] == 1:
    prediction = "VeryMildDementedd"
elif result[0][3] == 1:
    prediction = "ModerateDemented"
print(prediction)

```

predict.py

```

import tensorflow as tf

```

```

from keras_preprocessing.image import ImageDataGenerator
from keras_preprocessing import image
import numpy as np
import easygui
from keras.models import load_model
import os
import tkinter as tk
from tkinter import *
from tkinter import filedialog
from tkinter.filedialog import askopenfile
from PIL import Image, ImageTk
my_w = tk.Tk()
my_w.geometry('1244x829+0+10')
my_w.title('Alizheimer disease Prediction')
my_font1=('times', 18, 'bold')
        bg = ImageTk.PhotoImage(file='brain-tumorr.png')
bgLabel = Label(my_w, image=bg)
bgLabel.place(x=0, y=0)
l1=tk.Label(my_w,text='UploadFiles&getresults',width=30,font=my_font1,bg
        = '#000080',fg='red',)
l1.place(x=550, y=190, width=300)
b1=tk.Button(my_w,text='UploadImages',width=20,command
        = lambda:result(), activebackground='#000080', bg='green')
b1.place(x=590,y=500, width=230, height=40)
print(tf.__version__)
def close():
my_w.destroy()
titleLabel = Label(my_w, text=' ALIZHEIMER DISEASE PREDICTION',

```

```

font=('italic', 22, 'bold '), bg='black', fg='white', )
titleLabel.place(x=0, y=40, width=1350, height=50)
endbtn=Button(my_w,text="Exit",font='italic14
bold',bg='black',fg='white',command=close)
endbtn.place(x=670,y=600,width=50)
classifierLoad = tf.keras.models.load_model('model2.h5')
def result():
    filename =upload_file()
    test_image2 = image.load_img(filename, target_size = (200,200))
    test_image2 = image.img_to_array(test_image2)
    test_image2 = np.expand_dims(test_image2, axis = 0)
    # cnn prediction on the test image
    result = classifierLoad.predict(test_image2)
    print(result)
    if result[0][1] == 1:
        prediction2="NonDemented"
    elif result[0][0] == 1:
        prediction2="MildDemented"
    elif result[0][2] == 1:
        prediction2="VeryMildDementedd"
    elif result[0][3] == 1:
        prediction2="ModerateDemented"
    print(prediction2)
    prediction=prediction2
    l2=tk.Label(my_w,text="Result
"+prediction,width=50,font=my_font1,bg='pink', fg='black',)
    l2.place(x=560, y=550, width=400)
    return filename

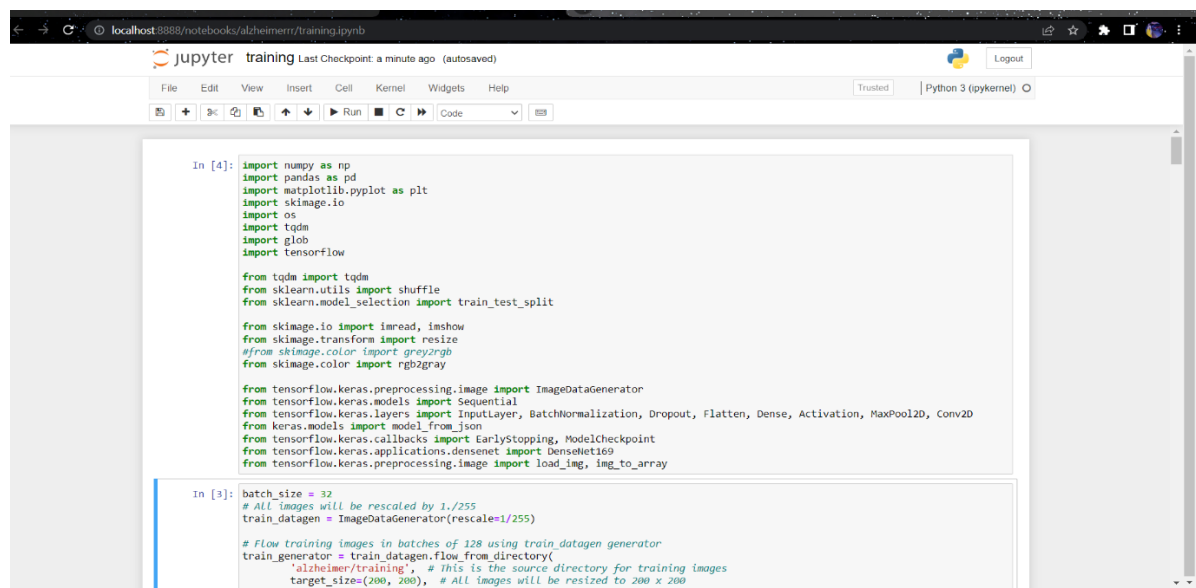
```

```

def upload_file():
    filename = easygui.fileopenbox()
    img=Image.open(filename) # read the image file
    img=img.resize((200,140)) # new width & height
    img=ImageTk.PhotoImage(img)
    e1=tk.Label(my_w)
    e1.place(x=590, y=240, width=240, height=250)
    e1.image = img
    e1['image']=img
    return filename
    my_w.mainloop()

```

6.2 SAMPLE SCREENSHOT



```

In [4]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import skimage.io
import os
import tqdm
import glob
import tensorflow

from tqdm import tqdm
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split

from skimage.io import imread, imshow
from skimage.transform import resize
from skimage.color import grey2rgb
from skimage.color import rgb2gray

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import InputLayer, BatchNormalization, Dropout, Flatten, Dense, Activation, MaxPool2D, Conv2D
from keras.models import model_from_json
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras.applications.densenet import DenseNet169
from tensorflow.keras.preprocessing.image import load_img, img_to_array

In [3]: batch_size = 32
# All images will be rescaled by 1./255
train_datagen = ImageDataGenerator(rescale=1./255)

# Flow training images in batches of 32 using train_datagen generator
train_generator = train_datagen.flow_from_directory(
    'alzheimer/training', # This is the source directory for training images
    target_size=(200, 200), # All images will be resized to 200 x 200

```

Figure 6.1: importing the necessary packages

Figure 6.1 depicts importing of all the necessary packages like pandas, NumPy, sklearn, TensorFlow, Keras the predefined functions are used.

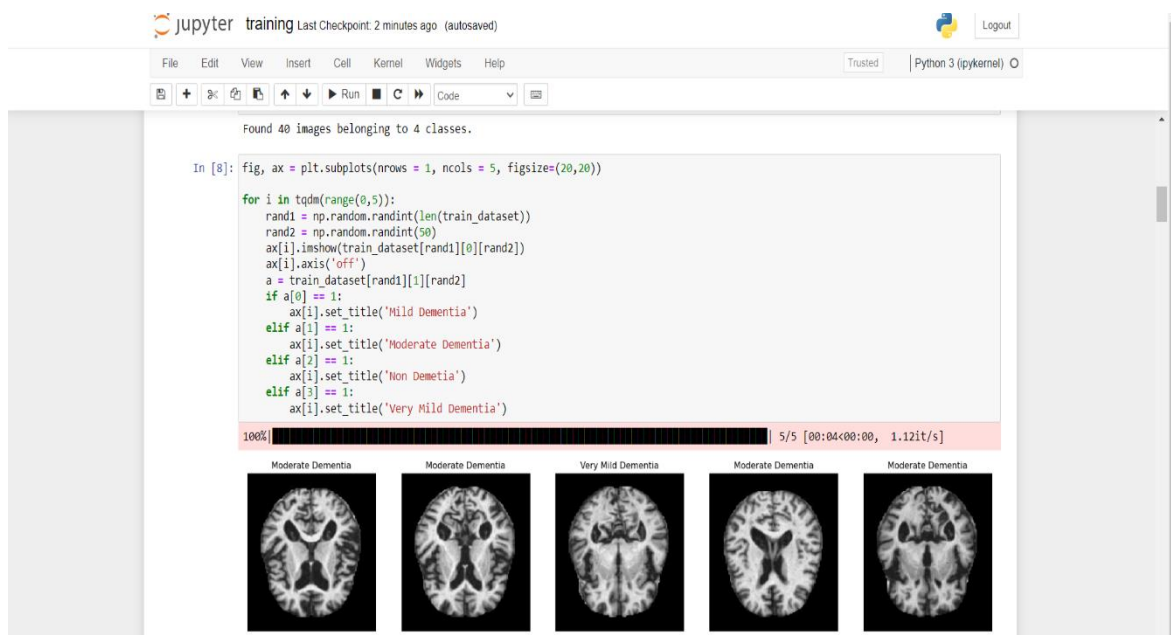


Figure 6.2: Training the dataset

Figure 6.2 depicts training the dataset where it uses the trained dataset of AD images and then assigned a random integer to the images. Specifying the looping conditions according to it.



Figure 6.3: Representation of Dense Net Code

Figure 6.3 Represents the coding part of the algorithm dense Net. where Convolutional, maxpooling at initial and for the 4 dense blocks and finally layers are flattened ,relu and SoftMax function is used.

```

In [21]: # Test Case 1: Non-Dementia
import numpy as np
from keras_preprocessing import image
import easygui

dic = test_dataset.class_indices
idc = {k:v for v, k in dic.items()}

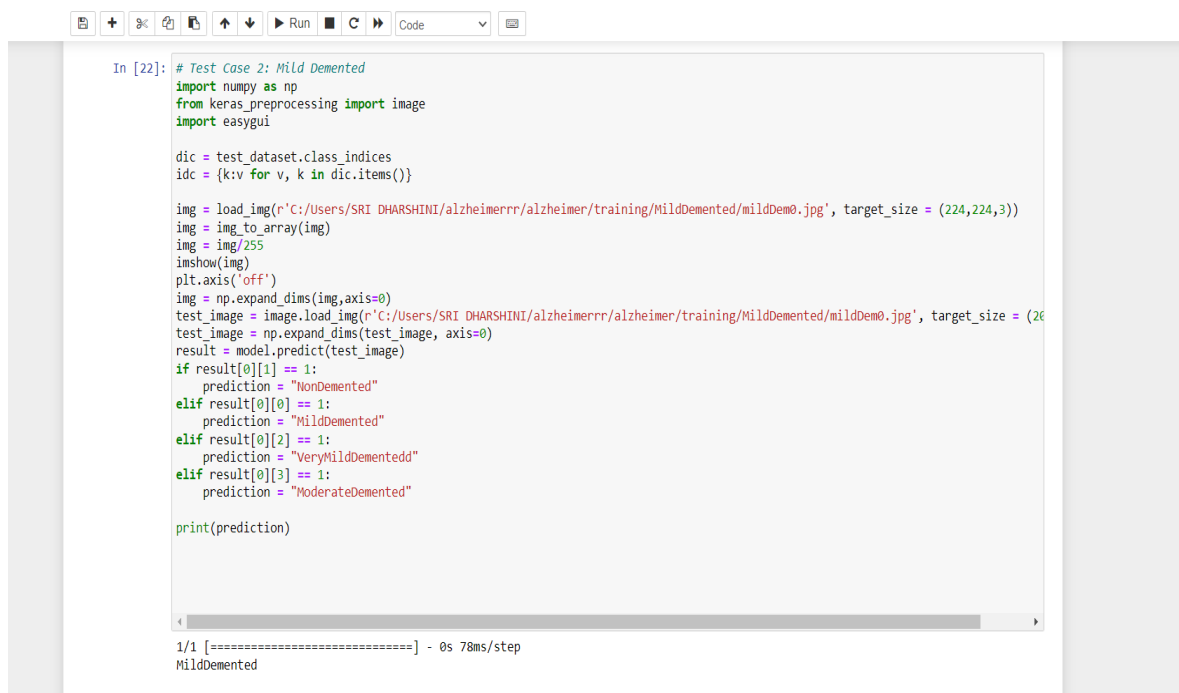
img = load_img(r'C:/Users/SRI DHARSHINI/alzheimerrr/alzheimer/training/NonDemented/nonDem0.jpg', target_size = (224,224,3))
img = img_to_array(img)
img = img/255
imshow(img)
plt.axis('off')
img = np.expand_dims(img,axis=0)
test_image = image.load_img(r'C:/Users/SRI DHARSHINI/alzheimerrr/alzheimer/training/NonDemented/nonDem0.jpg', target_size = (200,
test_image = np.expand_dims(test_image, axis=0)
result = model.predict(test_image)
if result[0][1] == 1:
    prediction = "NonDemented"
elif result[0][0] == 1:
    prediction = "MildDemented"
elif result[0][2] == 1:
    prediction = "VeryMildDementedd"
elif result[0][3] == 1:
    prediction = "ModerateDemented"

print(prediction)

```

Figure 6.4: Case 1

Figure 6.4 Represents the coding logic of the non-Dementia stage of the Alzheimer disease affected person.



```

In [22]: # Test Case 2: Mild Demented
import numpy as np
from keras_preprocessing import image
import easygui

dic = test_dataset.class_indices
idc = {k:v for v, k in dic.items()}

img = load_img(r'C:/Users/SRI DHARSHINI/alzheimerrr/alzheimer/training/MildDemented/mildDem0.jpg', target_size = (224,224,3))
img = img_to_array(img)
img = img/255
imshow(img)
plt.axis('off')
img = np.expand_dims(img,axis=0)
test_image = image.load_img(r'C:/Users/SRI DHARSHINI/alzheimerrr/alzheimer/training/MildDemented/mildDem0.jpg', target_size = (200,
test_image = np.expand_dims(test_image, axis=0)
result = model.predict(test_image)
if result[0][1] == 1:
    prediction = "NonDemented"
elif result[0][0] == 1:
    prediction = "MildDemented"
elif result[0][2] == 1:
    prediction = "VeryMildDementedd"
elif result[0][3] == 1:
    prediction = "ModerateDemented"

print(prediction)

```

1/1 [=====] - 0s 78ms/step
MildDemented

Figure 6.5: Case 2

Figure 6.5 Represents the coding logic of the Mild Dementia stage of the Alzheimer disease affected person.


```

In [24]: # Test Case 3: Moderate Demented
import numpy as np
from keras_preprocessing import image
import easygui

dic = test_dataset.class_indices
idc = {k:v for v, k in dic.items()}

img = load_img(r'C:/Users/SRI DHARSHINI/alzheimerrr/alzheimer/training/ModerateDemented/moderateDem26.jpg', target_size = (224,224))
img = img_to_array(img)
img = img/255
imshow(img)
plt.axis('off')
img = np.expand_dims(img,axis=0)
test_image = image.load_img(r'C:/Users/SRI DHARSHINI/alzheimerrr/alzheimer/training/ModerateDemented/moderateDem26.jpg', target_size = (224,224))
test_image = np.expand_dims(test_image, axis=0)
result = model.predict(test_image)
if result[0][1] == 1:
    prediction = "NonDemented"
elif result[0][0] == 1:
    prediction = "MildDemented"
elif result[0][2] == 1:
    prediction = "VeryMildDementedd"
elif result[0][3] == 1:
    prediction = "ModerateDemented"

print(prediction)

```

Figure 6.6: Case 3

Figure 6.6 Represents the coding logic of the Moderate stage of the Alzheimer disease affected person.

```

# Test Case 4: Very Mild Demented
# Test Case 3: Moderate Demented
import numpy as np
from keras_preprocessing import image
import easygui

dic = test_dataset.class_indices
idc = {k:v for v, k in dic.items()}

img = load_img(r'C:/Users/SRI DHARSHINI/alzheimerrr/alzheimer/training/VeryMildDementedd/verymildDem1.jpg', target_size = (224,224))
img = img_to_array(img)
img = img/255
imshow(img)
plt.axis('off')
img = np.expand_dims(img,axis=0)
test_image = image.load_img(r'C:/Users/SRI DHARSHINI/alzheimerrr/alzheimer/training/VeryMildDementedd/verymildDem1.jpg', target_size = (224,224))
test_image = np.expand_dims(test_image, axis=0)
result = model.predict(test_image)
if result[0][1] == 1:
    prediction = "NonDemented"
elif result[0][0] == 1:
    prediction = "MildDemented"
elif result[0][2] == 1:
    prediction = "VeryMildDementedd"
elif result[0][3] == 1:
    prediction = "ModerateDemented"

print(prediction)

```

Figure 6.7: Case 4

Figure 6.7 Represents the coding logic of the Very Mild Dementia stage of the Alzheimer disease affected person.

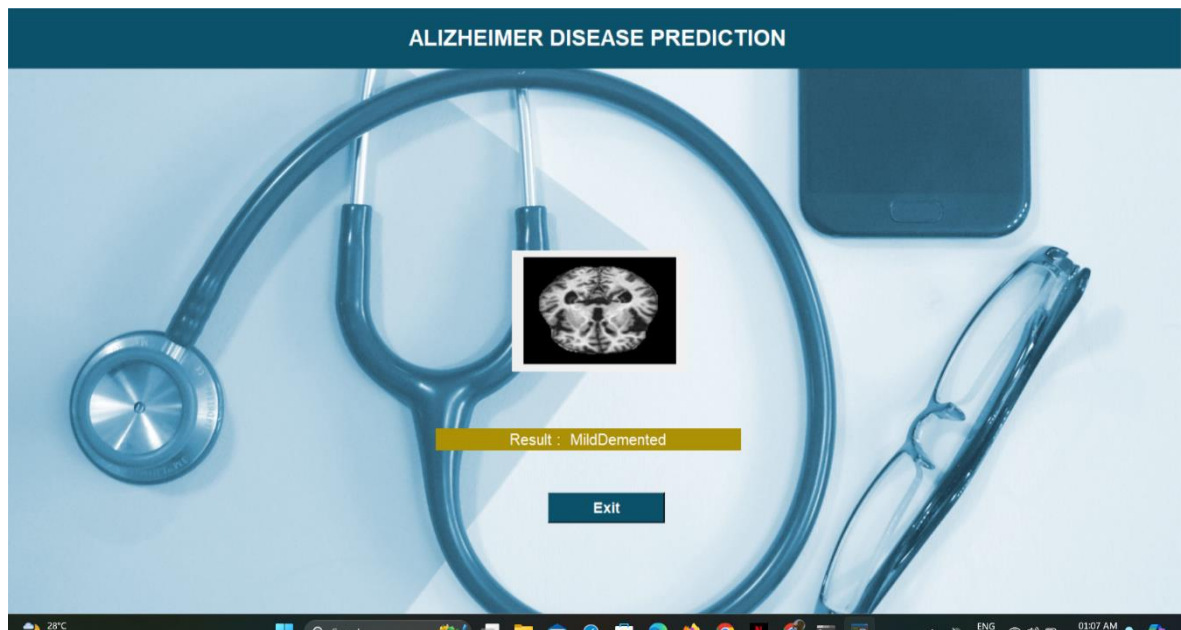


Figure 6.8: Representation of Mild Dementia

Figure 6.8 Represents the output image of Mild dementia. The code analyses and recognizes the given image falls under the mild demented stage of the disease.

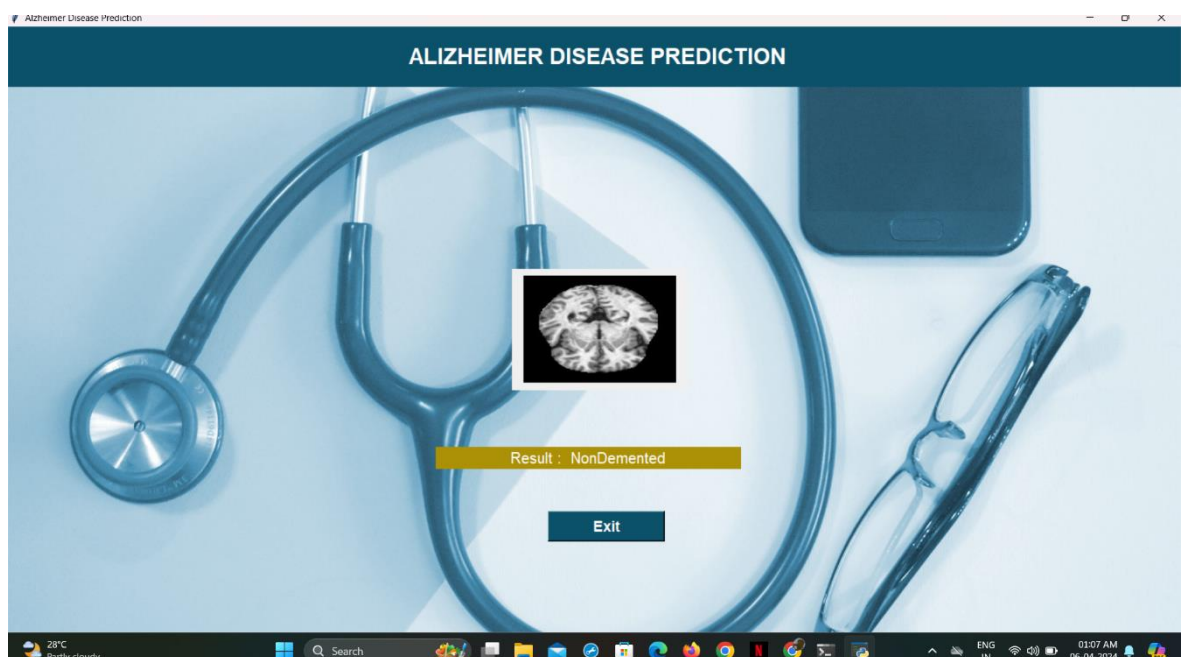


Figure 6.9: Representation of non-dementia

Figure 6.9 Represents the output image of Non dementia. The code analyses and recognizes the given image falls under the Non demented stage of the disease.

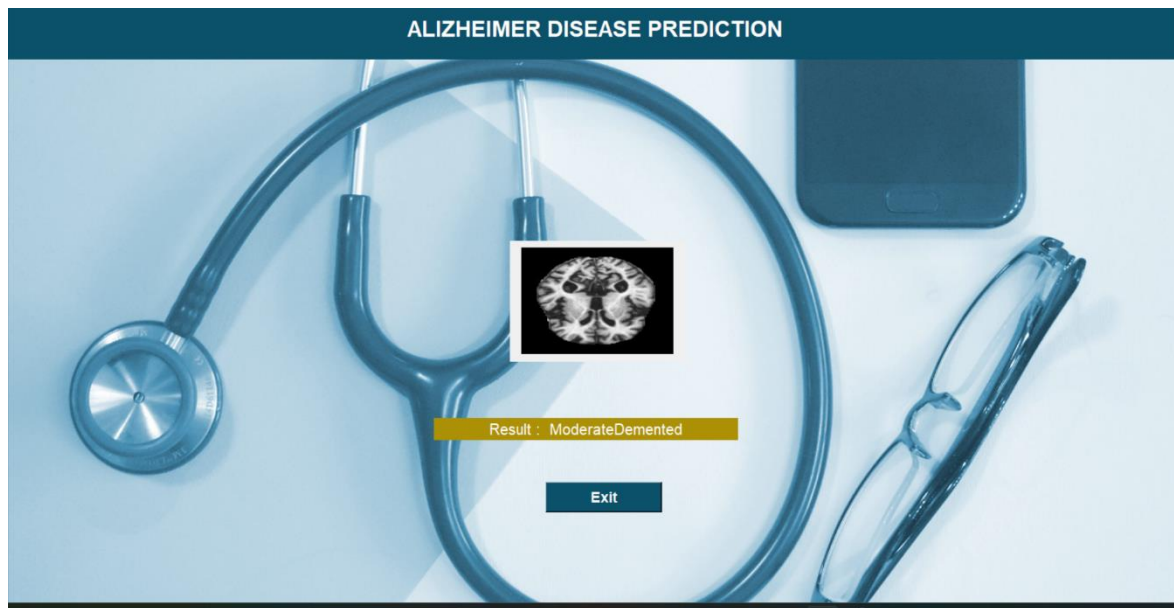


Figure 6.10: Representation of Moderate Dementia

Figure 6.10 Represents the output image of Moderate dementia. The code analyses and recognizes the given image falls under the Moderate demented stage of the disease.

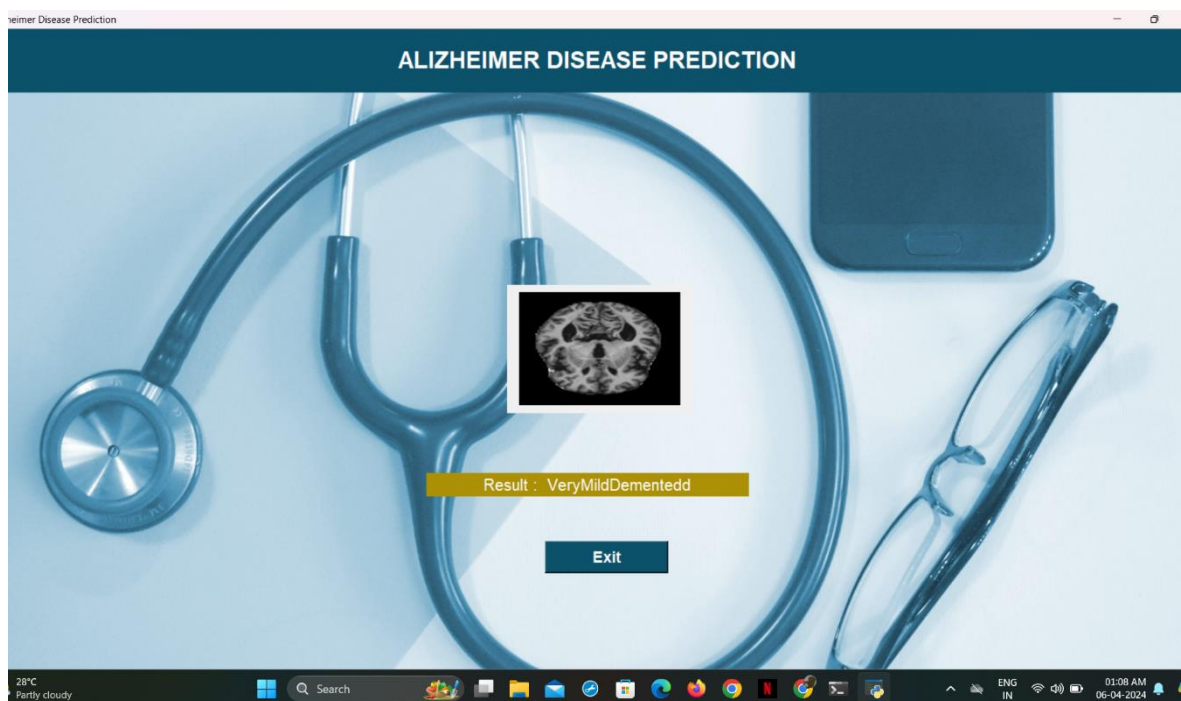


Figure 6.11: Representation of Very Mild Dementia

Figure 6.11 Represents the output image of Very Mild dementia. The code analyses and recognizes the given image falls under the Very Mild.

FUTURE WORK

Far greater expert optimism exists about breakthroughs in AD in the next 20 years than in the prior 20 years. In our assessment, 10 breakthroughs were judged as being at least 70% likely to occur by 2037, whereas in our 2001 study no breakthrough was judged as being even 50% likely by 2021.

This optimism is reflected in the clinical pipeline for novel therapies, with a wide range of possibly disease-modifying biologics and small molecules now in Phase II and III clinical trials.

However, challenges remain in delivering the predicted new AD therapies to patients, ranging from the use of appropriate cognitive screening tools to the preparedness of national healthcare systems to diagnose and treat large numbers of potentially eligible patients.

CONCLUSION

In this paper we proposed a simple and robust classification approach of MRI scans for Alzheimer's disease diagnosis. The approach is based on visual content description of anatomical structure of a brain region involved in AD (hippocampal area). We proposed a late fusion of classification results on two biomarkers: hippocampus and CSF.

The experiments showed that combining hippocampus features and CSF amount classification gave better accuracy especially when discriminating between AD and MCI than when using either visual features or CSF volume separately for discriminating between AD and MCI than using either visual features extraction or CSF volume computation separately.

We also demonstrated that the proposed method provides better classification accuracy compared to other volumetric methods. In the perspective of this work, we plan to use multiple ROIs, but also multiple MRI modalities in the established classification framework.

REFERENCE

- [1] AbolBosher, Kunholee, “Volumetric FeatureBased Alzheimer’s Disease Diagnosis From sMRI Data Using Convolutional Neural Networks and a Deep Network”, in IEEE Access, vol. 9, pp. 29870- 29882, 2021.
- [2] Jiang Wang, Kai Li, Shanshan Li, “Feature Extraction and Identification of Alzheimer’s disease based on Latent Factor of Multi-Channel EEG”, in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 29, pp. 1557- 1567, 2021.
- [3] Baoqiang Ma, Yan Zhao, “Prediction of Alzheimer’s Disease Progression with MultiInformation Generative Adversarial Network”, in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 3, pp. 711-719, March 2021.
- [4] Qing Xiao, Zhenyuan Ning, “Relation-induced Multi-modal Shared Representation Learning for Alzheimer’s disease Diagnosis”, in IEEE Transactions on Medical Imaging, vol. 40, no. 6, pp. 1632-1645, June 2021.
- [5] Bumshik Lee, Jae Young Choi, “Combining of Multiple Deep Networks via Ensemble Generalization Loss, Based on MRI Images, for Alzheimer’s disease Classification”, in IEEE Signal Processing Letters, vol. 27, pp. 206-210, 2021.
- [6] M.S Geetha Devasana J.Neelaveni, “Alzheimer Disease Prediction using Machine Learning Algorithms:further evidence of its accuracy via a transfer learning approach”, Int Psychometrist, 2022.
- 7] Amira Rezk,Nora Shoaip, “ A Comprehensive Fuzzy Ontology-Based Decision Support System for Alzheimer’s disease Diagnosis”, in IEEE Access, vol. 9, pp. 31350-31372, 2022.
- [8] Chandran Venkatesan,Suriya Murugan, “A Deep Learning Model for Early Diagnosis of Alzheimer Diseases and Dementia from MR Images”, in IEEE Access, vol. 9, pp. 90319-90329, 2023 .

[9] Chongfeng Cao,Tian Zhu, “Anatomical Landmarks and DAG Network Learning for Alzheimer’s disease Diagnosis”, in IEEE Access, vol. 8, pp. 206063-206073, 2023.