

ALZHEIMER'S DISEASE CLASSIFICATION USING DEEP LEARNING

A MINOR PROJECT REPORT

Submitted by

**ROHITH REDDY. A
[RA1911003010361]
CHITRALEKHA.CH
[RA1911003010387]**

Under the guidance of

Ms.K.R.JANSI

(ASSISTANT PROFESSOR)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY



S.R.M.Nagar, Kattankulathur, Chengalpattu District

NOVEMBER 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section3 of UGC Act,1956)

BONAFIDE CERTIFICATE

Certified that 18CSP107L minor project report titled “***ALZHEIMER’S DISEASE CLASSIFICATION USING DEEP LEARNING***” is the bonafide work of “**ROHITH REDDY.A [RA1911003010361], CHITRALEKHA.CH [RA1911003010387]**” who carried out the minor project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Ms.K.R.JANSI
GUIDE
Assistant professor

SIGNATURE

Dr.M. Pushpalatha
HEAD OF THE DEPARTMENT
Professor
Dept. of Computing Technologies

Signature of the Panel Head
Dr.K.R.JANSI
Assistant professor

ABSTRACT

When patients are aware of their risk for developing Alzheimer's disease (AD), they are better able to take preventative steps before permanent brain damage occurs, making an accurate diagnosis of AD during inpatient treatment crucial. Among those aged 65 and up, Alzheimer's is so far the most popular form of dementia. There's a significant amount of interest in making use of machine learning to enhance the diagnosis of metabolic illnesses popularly known as Alzheimer's and diabetes. The rates at which they occur are increasing alarmingly every year. Alzheimer's causes brain changes due to neurodegenerative processes. Conditions impairing cognitive abilities are on the rise, burdening patients' loved ones and the healthcare system. The social, economic, and monetary sectors will all feel the effects of these shifts profoundly. Diagnosing Alzheimer's disease in its early stages is challenging. Studies have indicated that if Alzheimer's disease is treated early on, it has a better chance of being cured and has fewer side effects. Parameters used to forecast the onset of Alzheimer's disease can be optimized using a variety of classifiers, including Decision Trees, Support Vector Machines, Gradient Boosting, Random Forests, and Voting classifiers. Recently, deep learning (DL) has seen widespread use in the field of Alzheimer's disease early detection. We explore how DL can help in the early identification of AD and present a brief literature review on the topic.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	ABSTRACT	iii
	TABLE OF CONTENTS	iv
	LIST OF FIGURES	v
	ABBREVIATIONS	vi
1	INTRODUCTION	1
1.1	Overview	1
1.2	Alzheimer's Disease	2
1.3	Motivation	5
1.4	Problem statement	5
1.5	System Requirements	6
2	LITERATURE SURVEY	7
2.1	General	7
2.2	Review of Literature Survey	7
2.3	Existing System	10
2.4	Drawbacks	10
3	SYSTEM ARCHITECTURE AND DESIGN	11
3.1	Deep Learning model	11
3.2	Data flow Diagram	12
3.3	Use case Diagram	13
3.4	Class Diagram	14
3.5	Sequence Diagram	14
3.6	E.R Diagram	15
3.7	Collaboration Diagram	16

	METHODOLOGY	17
4		
4.1	CNN working principle	17
4.2	Keras model	21
4.3	Preparing the data set	22
5	CODING AND TESTING	24
6	RESULTS AND DISCUSSIONS	35
7	CONCLUSION AND FUTURE ENHANCEMENT	36
	REFERENCES	37
	APPENDIX	
A	CONFERENCE PUBLICATION	39
B	JOURNAL PUBLICATION	40
C	PLAGIARISM REPORT	41

ABBREVIATIONS

CNN	Convolutional Neural Network
AD	Alzheimer's Disease
DL	Deep Learning
ML	Machine Learning
MRI	Magnetic Resonance Imaging
ANN	Artificial Neural Network
MCI	Mild Cognitive Impairment
FDG	Fluorodeoxyglucose
ADNI	Alzheimer Disease Neuroimaging Initiative
DFD	Data Flow Diagram

LIST OF FIGURES

Figure No.	Figure Name	Page No.
3.1	Architecture diagram	11
3.2	Data flow diagram	12
3.3	Use case diagram	13
3.4	Class diagram	14
3.5	Sequence diagram	14
3.6	E.R diagram	15
3.7	Collaboration diagram	16

CHAPTER 1

INTRODUCTION

1.1 Overview

Alzheimer's disease (AD) is a nervous system disorder that shows symptoms such as short-term memory loss, psychosis, and delusional thinking. Most people believe that stress or old age are the primary causes of Alzheimer's disease. It is approximated that over 5 million persons in the US are suffering from this ailment. The medical treatment that is now offered for AD is insufficient. To keep Alzheimer's disease under control, constant medication administration is required.

Because it is a chronic condition, AD can continue to affect a person for a significant number of years, and it may even endure for their entire life. It is essential to administer medication at the appropriate time to prevent long-term damage to the brain. This method takes a lot of time and money to discover this ailment in its early stages, we need to collect a large amount of data, utilize complex algorithms for prediction and work with an experienced medical professional. Automated systems are not prone to the errors that are caused by humans, they are capable of being employed in medical decision support systems with a better degree of precision than human review.

Quantitative data from MRI scans, together with other indicators (chemicals, blood flow) have been used to advance our understanding of AD. So that they could identify those who were suffering from mental illness. Eliminating the need for human intervention to arrive at a diagnosis of Alzheimer's is another advantage of automating the procedure. In addition to lowering total expenses, automation also yields more precise results. By studying MRI scans and using prediction tools, we can, for instance, determine if a patient is suffering from dementia. A person with Alzheimer's disease is regarded as mentally impaired in the early stages. More precision will result from this procedure.

In the beginning stages of Alzheimer's, a person normally doesn't need aid from others to perform daily tasks. In some cases, a person may be able to keep working, driving, and participating in social events as usual.

Despite this, the individual may continue to experience anxiety or memory loss (such as forgetting frequently used terms or where they frequently go). Their loved ones have noticed that they have trouble calling them by their full name. The doctor may be able to tell if the patient has attention or memory issues by questioning them in depth. A few of the difficulties often encountered by those with Alzheimer's in the first stages of the disease.

- Challenging to recall the correct term or individual's name.
- Struggling to remember faces after meeting new individuals.
- Putting in a full day's labor in a professional or social context can be taxing.
- Forgetting something you just read or learned.
- Losing or not being able to locate something of importance.
- Making a strategy or organizing your day-to-day tasks and endeavors is becoming more and more challenging.

This study aims to pinpoint areas of research concerning ML frameworks and EHR-produced data.

1.2 Alzheimer's Disease

Alzheimer's disease (AD) is a degenerative brain condition that kills brain cells. The onset of AD is often slow, and the earliest symptoms may be dismissed as simple forgetfulness or the effects of ageing.

The ability to think clearly and make judgments, as well as do routine chores, declines as the disease advances in Alzheimer's patients. There is currently no treatment for the disease beyond a set of recommendations that may slow its progression.

Therefore, an accurate diagnosis will play a crucial role in enhancing their patients' quality of life. The ever-increasing number of people diagnosed with Alzheimer's disease in our society serves as a compelling ethical argument for the development of new tools to aid in the fight against the disease.

Alzheimer's disease accounts for almost two-thirds of all cases of dementia today, which affects approximately 50 million people globally. With a new case occurring every three seconds worldwide, Alzheimer's has surpassed cancer as the leading cause of death in the United States.

When considering how crucial early detection is to successful therapy, one must ask whether or not this is enough. Machine Learning allows for the rapid analysis of big datasets using a variety of methods, allowing for the rapid discovery of patterns and models. Techniques that are unnoticeable to human experience and reasoning are used to greatly improve diagnostic approaches in this way.

Moreover, with the advent of sophisticated deep neural networks, the field of Machine Learning is now more advanced than ever before. To put it plainly, deep neural networks allow the development of systems with the processing ability to accurately represent any finite deterministic mapping between a given set of inputs and an output set.

With the help of these networks, complicated tasks like image recognition and NLP may be performed. In light of the foregoing, this study will investigate if cutting-edge deep learning approaches can help with AD diagnosis.

Early detection of Alzheimer's disease is challenging because of the disease's unpredictability and the human capability for error.

Could it then be possible to examine all these characteristics using various deep learning algorithms and get a result that shows the probability of having such a disease? Maybe technology and medicine may work together again to find new ways to reveal the most important factors in the presence of sickness.

What causes Alzheimer's disease?

It is believed that the development of abnormal proteins within and around brain cells is the root cause of AD. Amyloid is the protein that accumulates around brain cells to create plaques. Getting older is the leading risk factor for Alzheimer's disease. Those older than 65 are typically the hardest hit.

After this age, the risk of developing Alzheimer's disease increases approximately every five years. Approximately 1 in 6 people above the age of 79 face a problem with dementia or memory loss. Age raises the likelihood of acquiring dementia, but there are things you may take to reduce this risk.

Examples include regular walks, proper nutrition, and mental stimulation.

The chance of developing Alzheimer's disease is increased by a number of lifestyle factors, including smoking, heavy alcohol consumption, and the presence of chronic diseases like cardiovascular disease, diabetes, hypertension, and obesity.

Some risk factors, including becoming older and having certain genes, can't be changed.

Alzheimer's disease progresses at widely varying rates.

Some patients with Alzheimer's disease live for 20 years or more following diagnosis, while the average lifespan is closer to 11 years.

When does one know they have AD?

One of the earliest symptoms of the disease is memory loss. Alzheimer's disease can also be indicated by a decline in cognitive abilities that are not directly related to memory. These abilities include the ability to find the right word, the ability to understand visual images and spatial relationships, and the ability to reason and judge.

In the same way as with many other diseases, Alzheimer's-related brain changes start long before the disease manifests itself clinically. This period, which is sometimes referred to as "preclinical Alzheimer's disease," likely begins around 10 to 15 years before people show symptoms. They have no clue that they have any sort of mental impairment. Due to brain damage caused by the illness, they are completely unaware of what is going on around them.

Problems with memory that interfere with day to day operations:

- Problems that are difficult to plan or solve
- Problems completing even routine tasks
- Confusion about time and place
- Trouble grasping pictures and interpreting spatial relationships
- Speak and write differently
- Lose your ability to remember where you put things,
- Forget how you got anywhere.
- Lack or bad judgment
- Delays or postponements in participating in work or social events
- Switches in attitude and personality

Alzheimer's disease treatment

There are three regularly given cholinesterase inhibitors:

Donepezil (Aricept) is authorised to treat all stages of dementia.

Once a day, a pill is taken.

It's common practise to prescribe one of three cholinesterase inhibitors:

The Alzheimer's drug donepezil (Aricept) has been approved for use at every senescence.

It comes in pill form and is taken daily.

The Alzheimer's drug galantamine (Razadyne) has been approved to treat mild to moderate cases of the disease

Mild to severe AD may be treated with rivastigmine .

1.3 Motivation

Alzheimer disease is a common type of dementia, but its etiology and pathophysiology remain a mystery. Preclinical Alzheimer's disease diagnosis has significant clinical implications for patient care. Short-term memory loss and paranoid suspicion are two of the earliest symptoms of Alzheimer's disease, but they are often attributed to normal aging and stress, or mistaken for the effects of other brain disorders, making it difficult to predict the disease until it reaches a more advanced stage with more recognizable symptoms.

Our goal in developing this project was to make Alzheimer's disease detection and categorization more effective, so that sufferers could devise their own treatments.

1.4 Problem statement

In particular, the deep learning algorithm has emerged as the method of choice for the rapid analysis of medical images.

Although Alzheimer's disease (AD) is widely recognised as the leading cause of dementia, experts estimate that only one-quarter of those who suffer from the condition receive a timely diagnosis. However, mild cognitive impairment (MCI) or earlier stages can be managed, and treatment is most effective when begun before significant downstream damage. There is no refractory treatment available. This classification uses deep learning to address the pressing need for improved diagnostics. In this project, we focus on Alzheimer's disease and explore the various diagnostic approaches currently in use.

Analyzed the various resources for studying Alzheimer's disease data and compared relevant studies. On average, a proper diagnosis of Alzheimer's is made within the recommended time frame.

1.5 System requirements

1. Software requirements:

1. Operating Systems: Linux and Windows
2. Anaconda simulation tool with Jupyter notebook

2. Hardware specifications:

1. Pentium IV/III processor
2. Hard disc: at least 80 GB
3. RAM should be at least 2 GB.

CHAPTER 2

LITERATURE SURVEY

2.1 General:

The purpose of a literature review is to examine the major findings and research methods that have been applied to a specific issue. Secondary sources are those that describe previously published information on a topic or, more specifically, about that topic within a defined time frame.

Its ultimate purpose is to bring the reader up to speed on the state of the art in the literature on a particular topic; it lays the groundwork for another objective, such as potential future research needs in the area; it comes before a research proposal, and it may be as simple as a synopsis of sources.

It often follows a certain structure and incorporates aspects of both summary and synthesis. Synthesis involves rearranging and rearranging material, while summary just restates key points from the source.

It could offer a fresh perspective on previously established ideas, synthesize new and established ways of thinking, or chart the development of thought in the area from its inception to the present day, including its most heated conflicts.

In some cases, a literature review will provide an assessment of the available sources and recommendations for those that are most useful to the reader.

2.2 Review of Literature Survey:

[1]The Alzheimer's disease datasets on OASIS and Kaggle are used to train a wide variety of patient data with machine learning algorithms like SVM, a Decision tree classifier, Random Forest classifier,xg boost leading to quicker and more accurate identification of those with the disease.

Pros: We used various machine learning techniques to categorize cases of Alzheimer's and compared their performance across a range of parameters.

Cons: Didn't rely heavily on CNN to determine Alzheimer's disease diagnosis.

[2]The purpose of this study is to train and test a deep learning system that can distinguish the period between aging and dementia to that of Alzheimer's disease based on a PET scan of the brain fluorodeoxyglucose (FDG) and compare its results to those of human radiologists. Data containing the prospective Alzheimer Disease Neuroimaging Initiative (ADNI) dataset imaging studies from 2005-2017 and the retrospective independent test set were combined to create a database of 18F-FDG PET brain pictures (40 imaging studies from 2006-2016, 40 patients).

The clinical findings were recorded after the end of the follow-up time. Following training on 90% of the ADNI data set using the InceptionV3 architecture, the findings were compared to those of human radiologist readers on the remaining ten percent and a test set. Multiple metrics, including sensitivity, specificity, ROC, and ROC area, as well as a saliency map and a t-distributed stochastic neighbor embedding, were used to evaluate the model.

[3]Although many different methods and ML algorithms have been studied to extract patterns from the brain scan data to help in the clinical and research diagnosis of Alzheimer's disease, it has proven difficult to distinguish between dementia and healthy brain data in old people (age > 75) due to similar patterns of brain shrinkage and image intensities. These pipelines, run on a graphics processing unit (GPU) based high-performance computing platform, performed extensive and rigorous preprocessing of the data. Following this, a large number of training images were input into a convolutional neural network (CNN) to extract low- to high-level features that are robust to scale and shift changes. In this study, for the first time, deep learning was applied to fMRI data to assess medical imaging and diagnose Alzheimer's.

The fMRI and MRI pipelines, respectively, achieved good, repeatable accuracy of 99.9%, and 98.84% after adopting the planned pipelines, which suggests an improvement in the differentiation of output in comparison to earlier investigations. In addition, subject-level categorization for therapeutic purposes was carried out, with approximate accuracy rates of 95.33% for the fMRI pipeline and 97.88% for the MRI pipeline.

The decision-making system at the subject level enabled the fMRI rate to increase to 97.7 percent, while simultaneously enabling the MRI pipeline rate to achieve 100 percent.

[4]There has been a huge rise in the senior population as the effects of China's rapid aging become more apparent. Simultaneously, there has been a rise in the number of people diagnosed with Alzheimer's disease (AD). The current gold standard for diagnosing Alzheimer's disease requires radiologists with extensive experience to examine brain structural nuclear magnetic resonance (MRI) images and make a subjective diagnosis. Potential for an incorrect diagnosis.

Image classification using a deep learning Convolutional Neural Network (CNN) shows excellent performance and accurate classification when applied to MRI scans of Alzheimer's patients and healthy controls (NC). By leveraging deep learning and transfer learning, we evaluate and contrast the VGG 16 network model of the convolutional neural network with the MobileNet network model. When comparing classification accuracy, we discover that the MobileNet network model outperforms the VGG 16, network model.

[5]Researchers have been able to gain a deeper understanding of the problem at hand and discover novel solutions to previously intractable medical dilemmas thanks to the proliferation of machine learning techniques in the biomedical sciences over the past decade. These techniques have been used in everything from drug delivery systems [7] to medical imaging.

Deep learning is a machine learning technique that excels in categorization and feature extraction at a high level. To distinguish Alzheimer's disease brains from healthy brains, we trained a convolutional neural network in this study. This form of medical data classification is important because it may be used to create a prediction model or system that can distinguish between disease types and healthy patients, or estimate the severity of a disease. Selecting the most discriminative features has traditionally been the most difficult component of clinical data classification, especially for diseases like Alzheimer's. Using a Convolutional Neural Network (CNN) with the popular architecture LeNet-5, we successfully segmented functional MRI data from Alzheimer's patients and healthy controls with an accuracy of 96.85% on test data based on trained data. This research proves that extracting shift- and scale-invariant features with CNN and subsequently classifying the data with deep learning is the most efficient method for distinguishing

clinical data from healthy data in fMRI. Taking this approach allows us to extrapolate our methods to predict behavior in systems with greater complexity.

[6]Alzheimer's disease is a devastating kind of dementia. It kills off brain cells, leaving victims unable to remember things, thinks clearly, or carry on with their regular lives. Although there is currently no cure for Alzheimer's disease, early diagnosis can greatly improve quality of life. The accuracy of Alzheimer's disease diagnoses can be greatly enhanced by using machine learning techniques. Recent years have seen tremendous advancements in the application of deep learning to medical image processing. It has received scant attention that deep learning methods can be used to diagnose and classify Alzheimer's disease. Based on Brain MRI data, we introduce a deep-learning model for identifying and categorizing multi-class Alzheimer's disease. We show the Opening of Access Series of Imaging Studies database can benefit from an extremely deep convolutional network that we've developed.

2.3 Existing system

Here, we introduce a subject action object (SAO) analyses, a method for comparing medical technologies that makes use of the semantics and the feature weights of specialised vocabularies, to better gauge their degree of similarity. The first step is the isolation and tidying of SAO's semantic structures. This is accomplished with the help of the single medical language system's semantic network (UMLS). Second, the SAO semantic frameworks are evaluated using the UMLS Metathesaurus's semantic information. It is helpful to gain perspective on the industry as a whole by contrasting the technologies employed by competitors and possible business partners.

This research improves upon a previously established method for quantifying the degree to which various medical technologies are analogous. Our proposed indicator for the importance of technological data is the feature weighting of the data analysis of dependencies. It is possible to evaluate the degree of semantic structure similarity between SAOs by employing Unified Modeling Language (UML).

2.4 Draw backs:

- 1)The use of CNN as a classifier has been overlooked.
- 2)Efforts to increase the frequency of diagnoses and precision of classification have not primarily focused on the disease's severity.
- 3)Unfortunately, the planned deployment of the model did not take place.

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN

3.1 DEEP LEARNING MODEL

- Alzheimer's diagnosis follows the same classification pattern. With cutting-edge technology and deep learning, neurologists may soon be able to diagnose Alzheimer's disease noninvasively. In the realm of image categorization, deep learning's convolutional neural network has proven to be highly effective (CNN).
- A database of T1-weighted contrast-enhanced MRI images including or not containing Alzheimer's disease was used for categorization. Since we used unaltered photos as input, no preparation of Alzheimer's data was required. Larger numbers of photos from various categories, including "normal" and "abnormal," are collected as samples. Each input image classification results in a different number of photographs being collected.
- Our method involves using Deep Learning (DL) to predict Alzheimer's disease outcomes. A DL method called Convolutional Neural Networks (CNN) was used in this study. If Alzheimer's is successfully classified via CNN and additional feature extraction techniques, the results may be improved.

Advantages:

- To identify Alzheimer's disease easily and reduce the workload of doctors in the medical field.
- In terms of deep learning techniques, it is the best model to identify Alzheimer's disease quickly.

Scope:

This work aims to implement and explore the effects of several CNN algorithms on prediction. The goal of our study is to categorize these pictures into those with Alzheimer's disease and those without. There are two distinct categories here. Additionally, one of the most fruitful areas of this research has been the extraction of hand-crafted features from raw photos following a variety of processing steps. The approach is used to extract features that are solely responsible for Alzheimer's disease prediction.

ARCHITECTURE DIAGRAM:

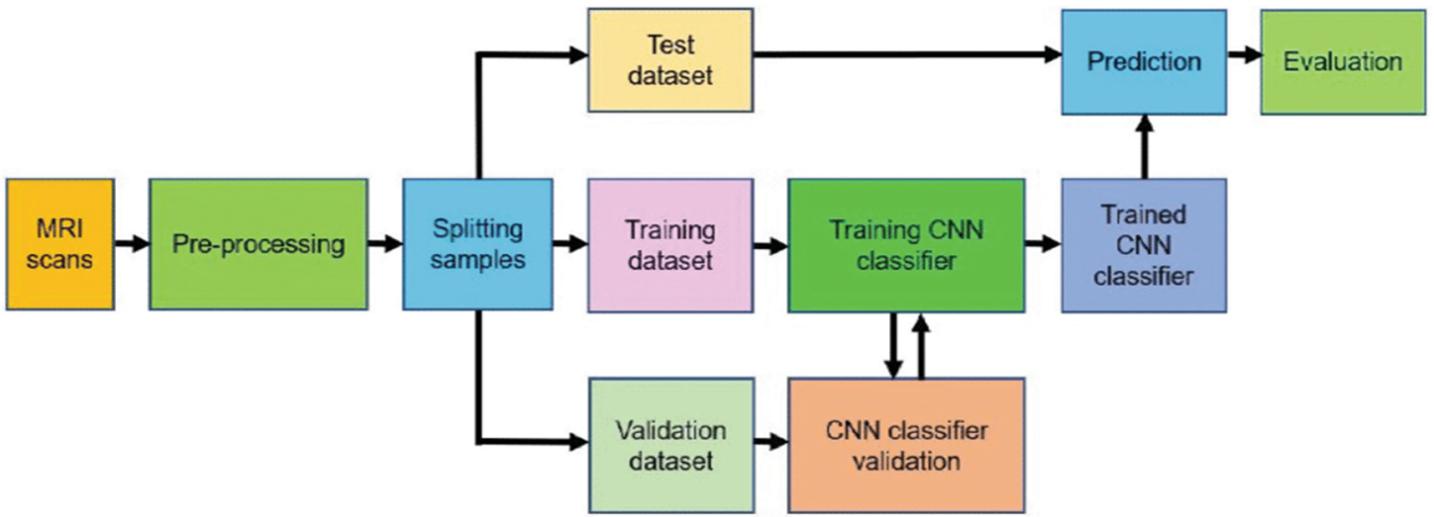


Fig 3.1 architecture diagram

The MRI scans are pre-processed and the samples are split into three parts naming them as a test, train, and validation. This splitting is done in different percentages i.e 78% to train the dataset in which 69% is divided into training set and 9% into validation set , and finally the remaining 12% is for testing set.

3.2 Data flow diagram:

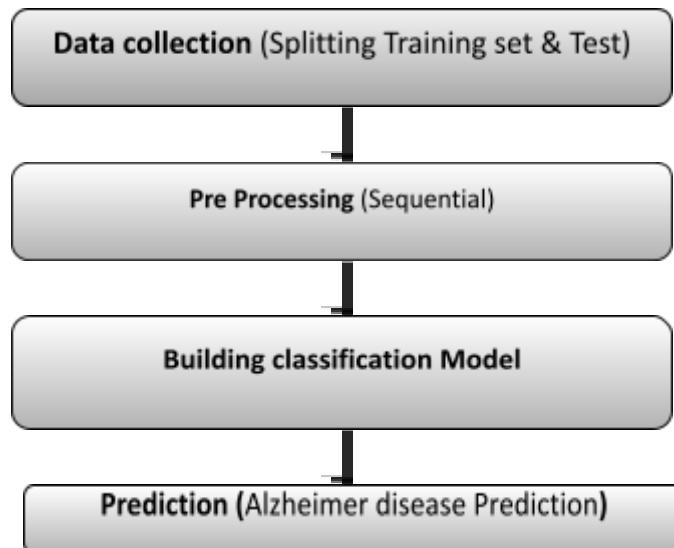


Fig 3.2 data flow diagram

Data flow diagram is a graphical representation of the “flow” of data. In the above mentioned Fig. 3.3 is the DFD of the Alzheimer's disease prediction which includes data collection then it has data pre-processing and the building of a classification model, and finally prediction.

3.3 USE CASE DIAGRAM

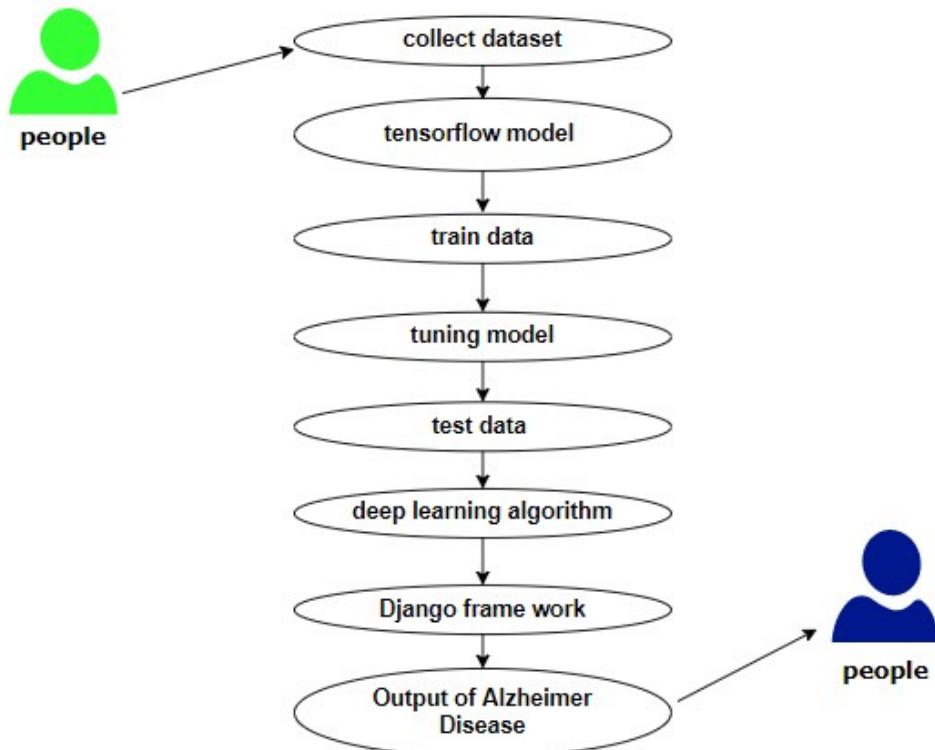


Fig 3.3 Use Case Diagram

Use case diagrams can be determined as the organized manner of system functionalities. In fig 3.4 the functionalities required are the collection of the dataset then after the model is used, training the data using the model, testing the data after the training, the deep learning algo used finally the output for the user with the help of the interface.

3.4 CLASS DIAGRAM

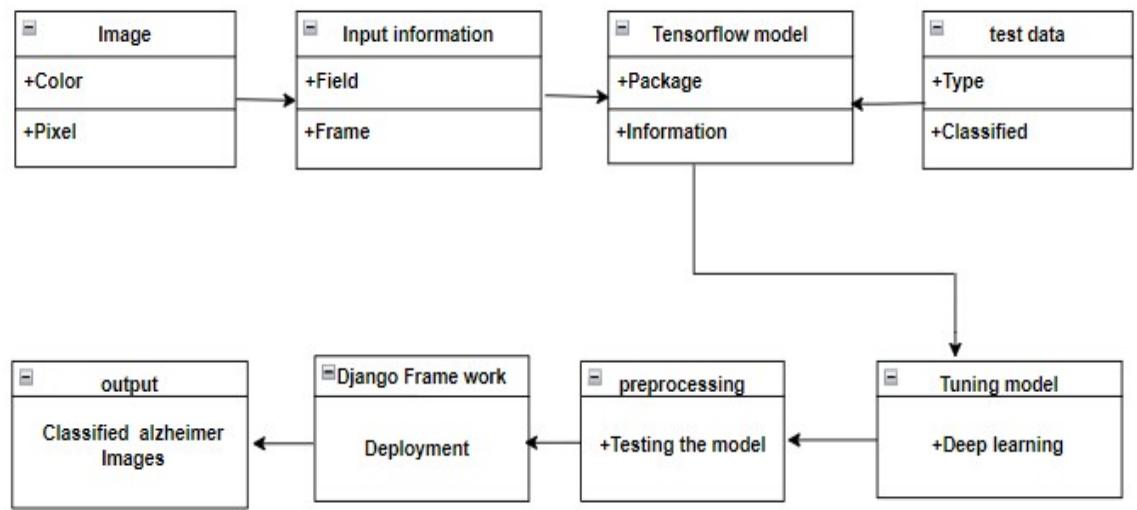


Fig 3.4 Class Diagram

Class diagrams generally represent the static view of the system and represent different aspects of the application. Using notes whenever required to describe some aspects of the diagram and at the end of the drawing, it should be understandable to the end user. Finally, before making the final version, the diagram should be made on plain paper and rechecked as many times as possible to make it correct.

3.5 SEQUENCE DIAGRAM

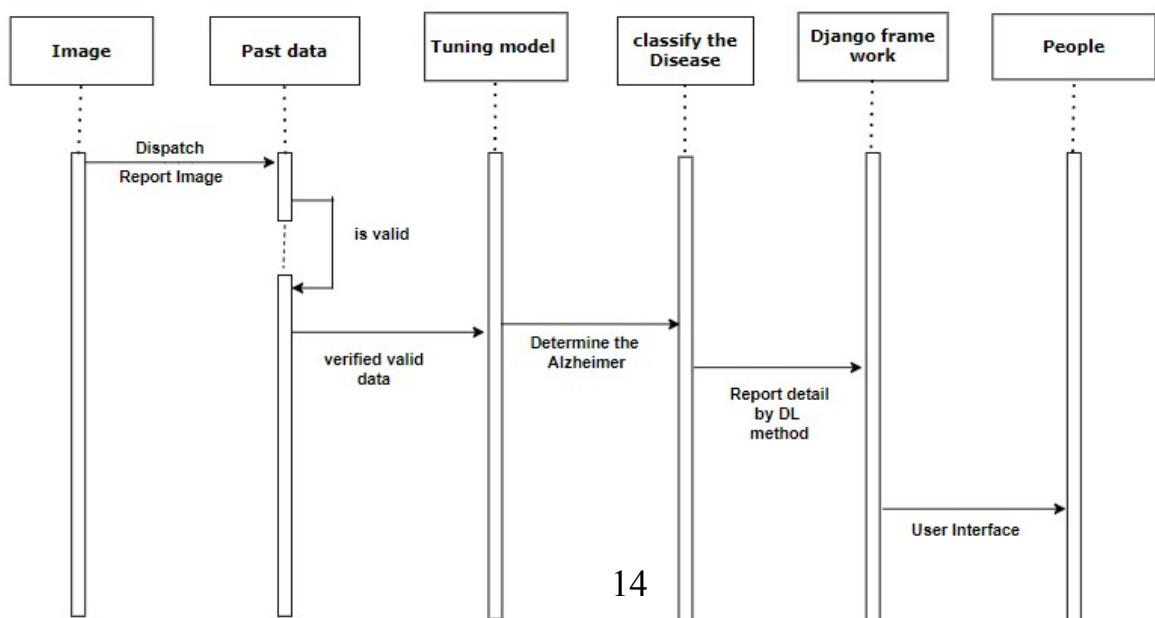


Fig 3.5 Sequence Diagram

The flow of logic within the system in a visualized manner is known as a sequence diagram. In the above-mentioned fig 3.5 image is dispatched as a report and then the past data is validated with the validated data. Tuning of the model is done after that disease classification, then the detailed report is generated by the deep learning model and then displayed to the user.

3.6 ER DIAGRAM

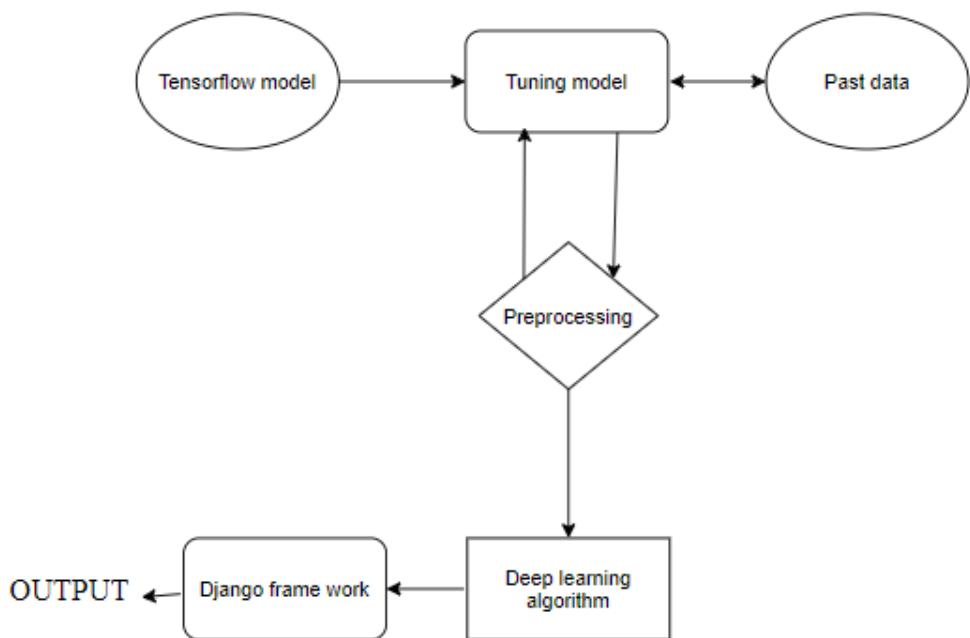


Fig 3.6 E.R Diagram

Relationships between entities (such as people, places, things, concepts, or events) in an information system can be represented graphically using an entity relationship diagram (ERD), also called an entity-relationship model. In figure 3.6 tensor flow is used for best practice to train the data then we tune the model with the past data and preprocessing is done after a deep learning algo is applied to predict the output.

3.7 COLLABORATION DIAGRAM

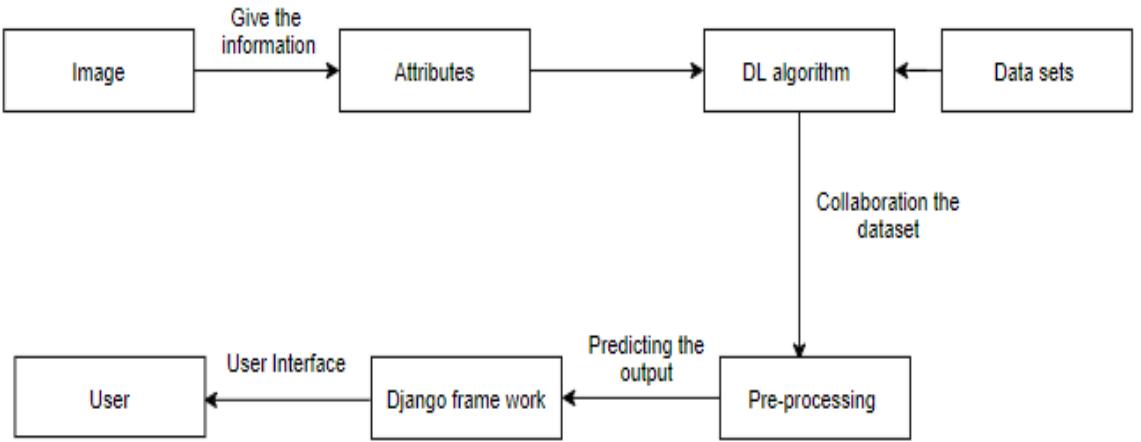


Fig 3.7 Collaboration Diagram

In figure 3.7 the collaboration diagram generally displays relationships between objects involved in interaction, and the sequence of messages exchanged among them during the interaction. The collaboration diagram is basically a decomposition of a class diagram. The collaboration diagram shows messages being sent between classes and objects. Here the image gives the information to the attributes which inturn goes in to the DL algorithm for preprocessing along with the datasets which helps in predicting the final output.

CHAPTER 4

METHODOLOGY

Training the model (Convolutional Neural Network): Images are resized, cropped, and converted to array format as part of the preprocessing of the dataset. Even the test image undergoes the same treatment. A collection of Alzheimer's disease images numbering in the hundreds; each one of these pictures can serve as a test case for your program. As a result of being trained on the training dataset, the model (CNN) can correctly detect the disease in the test image. Dense, Dropout, Activation, Flatten, Convolution2D, and MaxPooling2D are some of the layers that make up a CNN network. Following successful model training, the software is able to correctly identify Alzheimer's disease from images of the affliction in the training dataset. The test image is compared to the learned model once training and preprocessing have been completed successfully.

4.1 CNN Working Process:

One kind of ANN is a Convolutional ANN (or simply CNN). Image processing, classification, and segmentation are just a few applications of convolutional neural networks, which are neural networks with one or more convolutional layers.

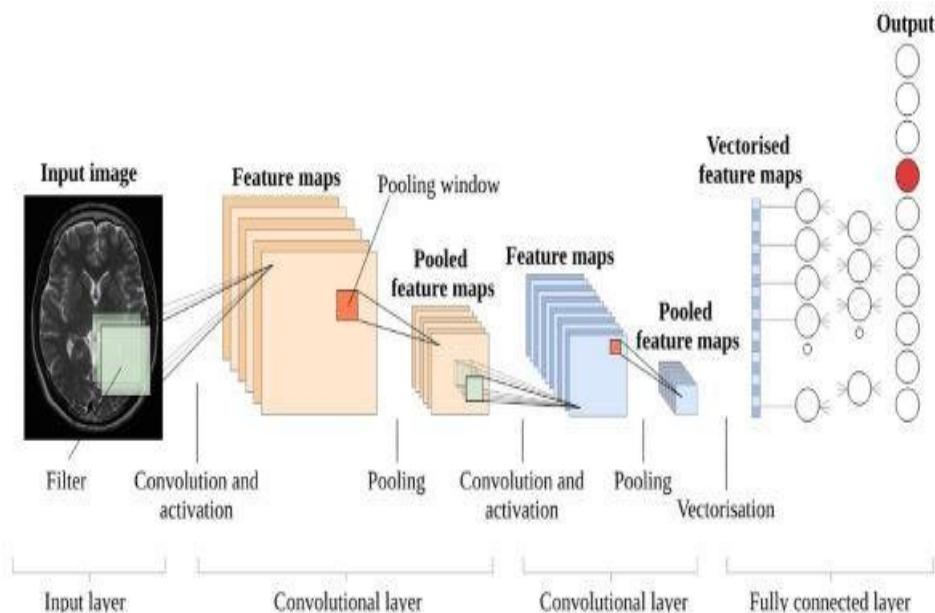


Fig 4.1 working of CNN

- **Convolution** : We feed images here, apply convolutions and perform feature extractions which are called feature maps.
- **Pooling** : After feature maps are extracted, we try to reduce the size of feature maps, that is compress them, which is downsampling.
- **Flattening** : After all the feature maps are pooled, we flatten them. Here, we consider the 2D pixels and convert into 1D and then feed them to our dense fully connected artificial neural network and train the model
- **Padding:**

There are a few ways to deal with the pixels around the edges:

- Inadequate resolution due to missing edge pixels
- Inflating by using pixels with no value
- Protective cushioning against reflections

By far the most effective method is reflection padding, which involves copying pixels from the image's edges and pasting them onto the image's periphery so that the convolutional kernel has enough room to analyse the edge pixels. With a 3x3 kernel, an extra pixel should be placed around the perimeter. The dimension has been halved and adjusted to the number of pixels added to each side. Many research articles follow a common practise of simply ignoring the edge pixels, which results in a loss of data (which is exacerbated by the presence of multiple deep convolutional layers). Due to this, I was unable to locate any suitable graphics that could effectively express some of the arguments made here without being deceptive or conflating stride 1 and stride 2 cvds.

- **Strides:**

If the input has width w and height h and the kernel iterates over each pixel in turn, the resulting output would also have width w & height h . The output channel/feature map size can be reduced by half by using a stride two convolution instead of a stride one convolution, in which the convolutional kernel takes strides of one pixel. Since the kernel only generates a single, aggregated output for each stride, the maximum value that could be produced input with width w , height h , and depth 3 with padding would be width $w/2$, height $h/2$, and depth 1.

- **Rectified Linear Unit:**

A Rectified Linear Unit is used as a non-linear activation function. A ReLU says if the value is less than zero, round it up to zero.

- **Normalization:**

The term "normalization" refers to the process of removing the mean and subtracting the standard deviation. In a process known as Min-Max scaling, the range of the dataset is transformed to be between -1 and 1, standardizing the data on the same scale.

Normalization of input features is a typical technique for achieving data standardization by means of the mean being subtracted & scale to random values. It is often crucial that the input features have the same order and variance and are centered around zero. Images, for example, have their data scaled so that the values range from 0 to 1 by dividing each pixel's value by 255.

- **Batch Normalisation:**

Training times can be cut in half and accuracy can be improved by a magnitude when using batch normalization to stabilize the predictions made by a network.

By removing the average batch's activations and dividing the standard variation of the batch's activations. However, even after normalizing the input, some activations may still be larger than average, rendering the network less stable. By applying a modification called batch normalization, we can keep the return confidence interval around 1 and the mean output close to 0. It's important to note that the way batch normalization functions varies between the training and inference phases.

- **Input shape:** An N-dimensional tensor of the form (batch size, input dimensions). The most typical scenario involves a 2 dim as input of shape (batch size, input dimensions).

- **Output shape:** Tensor in N dimensions, of the form (size of batch, units). The shape of the output would be (batch size, input dim) for a 2D input, for instance.

- **Image Data Generator:**

It is that which does a resizing transformation, a shearing transformation across a specific range, a zoom transformation, and a horizontal flipping transformation on the image. All conceivable image orientations are represented in this image data generator.

- **Training Process:**

The train dataset directory's data can be prepared for use using the train datagen. flow from the directory function. The image's intended size can be set with the target size parameter. All of the aforementioned is also true for the model-testing utility known as Test catagen. flow from a directory. The steps per epoch variable indicate the number of times the model will be run on the training data, while the fit generator is used to fit the data into the model built in the preceding section.

- **Epochs:**

It tells us how many iterations of forward and reverse training the model will undergo.

- **Validation process:**

The validation and test data are taken from the validation data. The quantity of validation/test samples is denoted by validation steps.

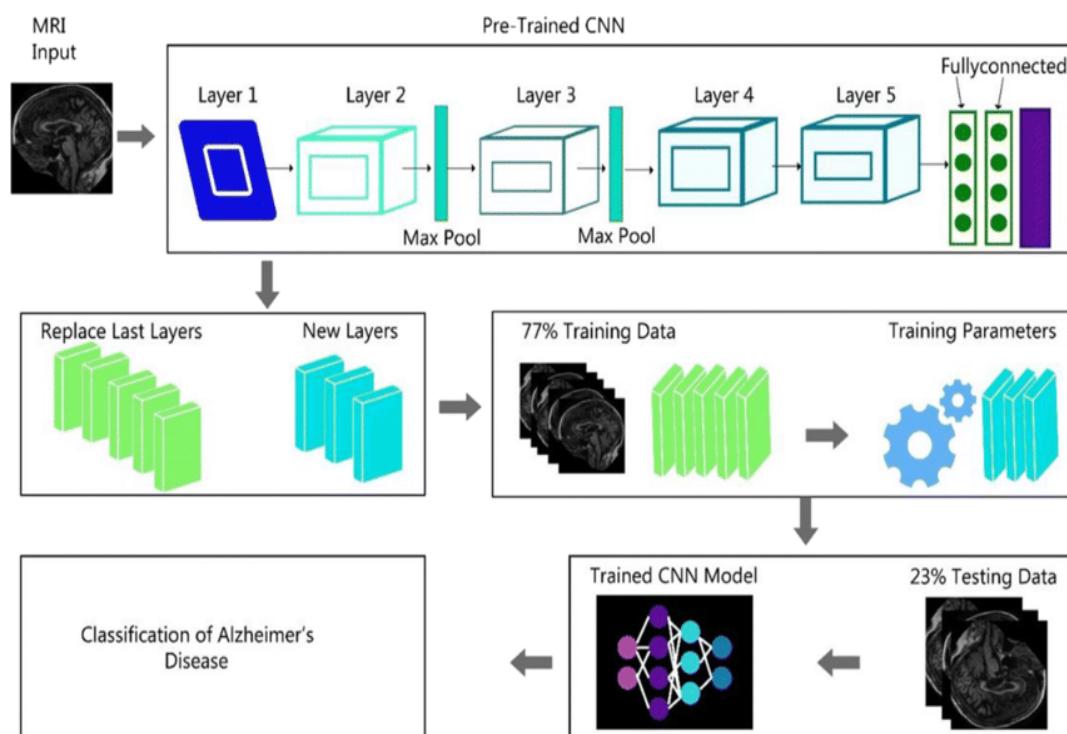


Fig 4.2 layers of CNN model

4.2 KERAS MODEL:

Despite being the simplest model the sequential model is limited to stacks of layers with a single input and output. The Functional API is a simple, feature-rich API that may be used with any kind of model design. You should be employing this method for the vast majority of users and uses. The Keras model is considered "industry strong." This is achieved through model subclassing, in which you create your implementation of the model.

Types of Keras Models

- Keras Sequential Model
- Keras Functional API

1. Sequential Model in Keras

Through this method, we can build up models in a methodical fashion. But we can't build models with multiple inputs and/or outputs because of this limitation. It works well with one-input, one-output layers in a simple stack. The presence of many inputs or outputs in any given tier of the stack renders this paradigm inappropriate. It is inappropriate even if non-linear topology is what we seek.

2. Functional API in Keras

To build a model as well as add layers in keras, it gives you greater freedom. Using a multi-input/multi-output paradigm is made possible by the API's functional programming interface. In addition, this facilitates the dissemination of these levels. Using Keras's functional API, we can construct layer graphs. Since a functional API is just a data model, it can be stored as a single file and used to recreate the same model even if you lose the source code. The graph can be modeled with little effort, and its nodes can be accessed without any hassle.

Model Subclassing in Keras

Developing models using a sequential approach is restrictive. The flexibility of a fully functional API is limited. However, you may make your own complete model in Keras. To

do this, a call method must be subclassed from the Model class. A Keras tensor is created with the help of the input() function.

ernels:

Inside each convolutional layer is a filter known as a convolutional kernel. The integer matrix that is of the same size as that the kernel is used as the filter, and it is applied to a subset of a input image pixels. For lucidity's sake, in the final channel/feature map, we multiply each pixel by its level equivalent within the kernel and sum the products. Each convolution is a special case of an affine function, making these changes linear in nature. A common form of input for computer vision is an RGB image, which has three colour channels. To keep things simple, let's use a grayscale image with a single channel (a two-dimensional matrix) and a convolutional kernel of size three . The kernel scans the first few rows of the input matrix, which contains the image's pixel values, by moving horizontally over the columns of integers. The kernel then descends in a vertical fashion to the succeeding rows. Please take note that the filter may skip over a single pixel or multiple pixels at once, as will be explained in further depth below.

4.3 PREPARING THE DATASET:

This dataset contains approximately 400 train and 100 test images, which were then classified into 2 classes:

- DEMENTED - The person is having alzheimer disease
- NON DEMENTED-The person is not having alzheimer disease

DATA SET:

Our dataset is in format of .jpg in which we divided them into demented and non-demented form.

Training : 387 images.

Testing : 104 images.

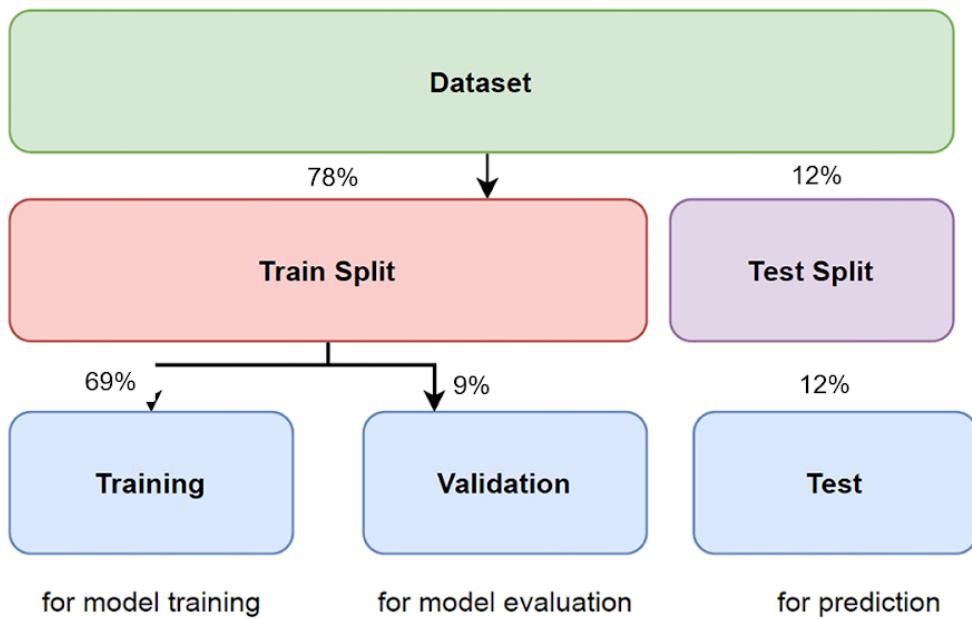


Fig 4.3 Splitting of datasets

The MRI scans are pre-processed and the samples are split into three parts naming them as a test, train, and validation. This splitting is done in different percentages i.e 78% to train the dataset in which 69% is divided into training set and 9% into validation set , and finally the remaining 12% is for testing set.

CHAPTER 5

CODING & TESTING

MODULE 1:

Using the keras preprocessing image data generator function, we must first import our data set before we can generate its dimensions, rescale it, determine its range, zoom in and out, and flip it horizontally. Next, we use the data generator to load our image dataset from a local folder. In this section, we specify the conditions under which the network should be trained, including the types of data to be used, the number of training iterations, the size of each training batch, and the training mode. The use of a classifier and a fit generating function, in addition to the number of training steps per epoch, the total number of epochs, validation data, and validation steps, allows us to train our dataset.

```
[5] # Dl framework - tensorflow, keras a backend
import tensorflow as tf
import tensorflow.keras.backend as K
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.layers import Input, Dense, Flatten, Dropout, BatchNormalization
from tensorflow.keras.layers import Conv2D, SeparableConv2D, MaxPool2D, LeakyReLU, Activation
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, EarlyStopping
from IPython.display import display
from os import listdir
from os.path import isfile, join
from PIL import Image
import glob
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense

import warnings
warnings.filterwarnings('ignore')

[7] dir_name_train_Demented = '/content/drive/MyDrive/Dataset-20221013T025357Z-001/Dataset/train/Demented'
dir_name_train_NonDemented = '/content/drive/MyDrive/Dataset-20221013T025357Z-001/Dataset/train/NonDemented'
```

```
[1] #!pip install tensorflow-gpu

[3] from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[4] import os
import numpy as np # linear algebra
import matplotlib.pyplot as plt
```

```
✓ [4] def plot_images(item_dir, n=6): #to represent images graphically
    all_item_dir = os.listdir(item_dir)
    item_files = [os.path.join(item_dir, file) for file in all_item_dir][:n]

    plt.figure(figsize=(35, 10))
    for idx, img_path in enumerate(item_files):
        plt.subplot(2, n, idx+1)
        img = plt.imread(img_path)
        plt.imshow(img, cmap='gray')
        plt.axis('off')

    plt.tight_layout()

[9] def Images_details_Print_data(data, path): #printing the details about images
    print(" ===== Images in: ", path)
    for k, v in data.items():
        print("%s:\t%s" % (k, v))

def Images_details(path):
    files = [f for f in glob.glob(path + "**/*.*", recursive=True)]
    data = {}
    data['images_count'] = len(files)
    data['min_width'] = 10**100 # No image will be bigger than that
    data['max_width'] = 0
    data['min_height'] = 10**100 # No image will be bigger than that
    data['max_height'] = 0
```

```

print("")
print("Trained data for Demented:")
print("")
Images_details(dir_name_train_Demented)
print("")
plot_images(dir_name_train_Demented, 9)

Trained data for Demented:

===== Images in: /content/drive/MyDrive/Dataset-20221013T025357Z-001/Dataset/train/Demented
images_count: 200
min_width: 176
max_width: 208
min_height: 208
max_height: 208



```

```

print("")
print("Trained data for NonDemented:")
print("")
Images_details(dir_name_train_NonDemented)
print("")
plot_images(dir_name_train_NonDemented, 9)

Trained data for NonDemented:

===== Images in: /content/drive/MyDrive/Dataset-20221013T025357Z-001/Dataset/train/NonDemented
images_count: 187
min_width: 176
max_width: 208
min_height: 208
max_height: 208



```

```

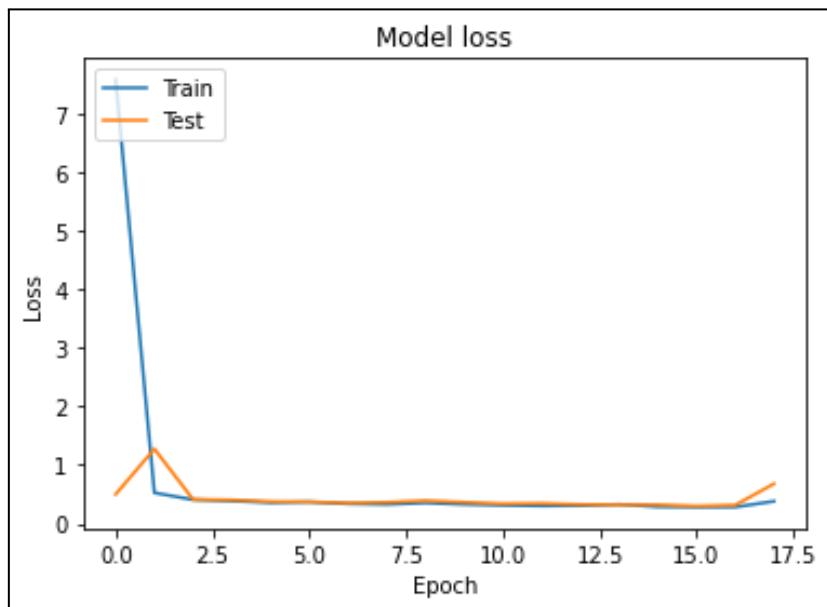
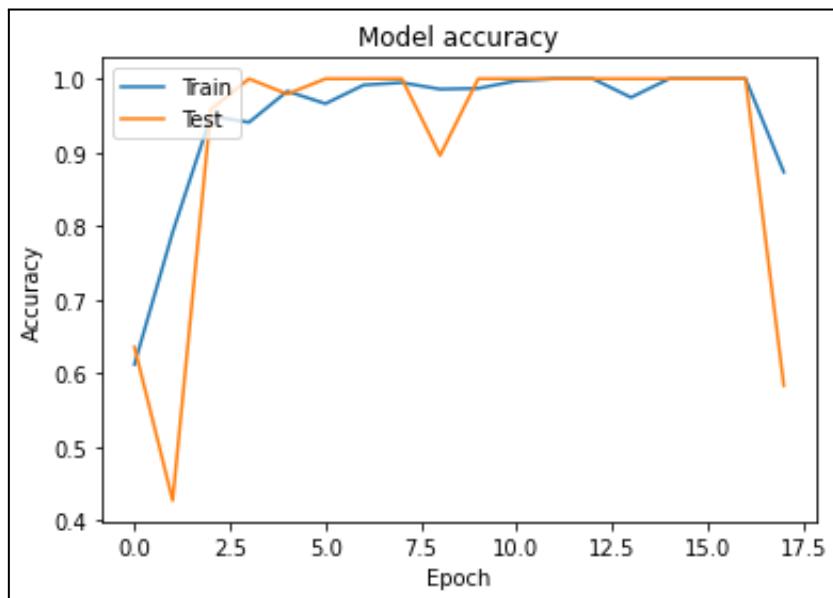
[20] ##### Fitting the model
history = Classifier.fit_generator(
    training_set, steps_per_epoch=training_set.samples // batch_size,
    epochs=epochs,
    validation_data=test_set, validation_steps=test_set.samples // batch_size)

Epoch 1/5
12/12 [=====] - 63s 6s/step - loss: 0.5949 - accuracy: 0.9549 - val_loss: 4.7427e-05 - val_accuracy: 1.0000
Epoch 2/5
12/12 [=====] - 4s 319ms/step - loss: 0.0022 - accuracy: 1.0000 - val_loss: 6.4431e-05 - val_accuracy: 1.0000
Epoch 3/5
12/12 [=====] - 4s 319ms/step - loss: 0.0017 - accuracy: 1.0000 - val_loss: 4.7950e-05 - val_accuracy: 1.0000
Epoch 4/5
12/12 [=====] - 4s 341ms/step - loss: 0.0012 - accuracy: 1.0000 - val_loss: 5.0926e-05 - val_accuracy: 1.0000
Epoch 5/5
12/12 [=====] - 4s 318ms/step - loss: 3.5875e-04 - accuracy: 1.0000 - val_loss: 7.3551e-05 - val_accuracy: 1.0000

def graph():
    #Plot training & validation accuracy values
    plt.plot(history.history['accuracy'])
    plt.plot(history.history['val_accuracy'])
    plt.title('Model accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.show()

```

TRAINING DATA:



MODULE-2:

After a machine learning algorithm has been trained on a training data set, it is put through its paces using a secondary (or tertiary) data set known as a test set.

Models are tested against the validation set for accuracy before being narrowed down for use in model selection. The test set, on the other hand, is used to check how effectively the final model (chosen in the previous stage) generalises to novel, unseen data.

Using the same data for both training and testing helps reduce the impact of data discrepancies and provides a clearer picture of the model's properties. As soon as a model has been processed using the training set, it may be put to the test by making predictions against the test set.

```
[2] # Dl framework - tensorflow, keras a backend
import tensorflow as tf

[3] import tensorflow.keras.backend as K

[4] from tensorflow.keras.models import Model

[5] from tensorflow.keras.models import Sequential

[6] from tensorflow.keras.layers import Input

[7] from tensorflow.keras.layers import Dense

[8] from tensorflow.keras.layers import Flatten

[9] from tensorflow.keras.layers import Conv2D

[10] from tensorflow.keras.layers import MaxPooling2D

[11] from tensorflow.keras.layers import Dropout

[12] from tensorflow.keras.layers import LeakyReLU

[13] from tensorflow.keras.layers import Activation

[14] from tensorflow.keras.optimizers import Adam
```

✓ 0s completed at 15:52

```
+ Code + Text
[1] from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[20] model = Sequential()
# 1st Convolutional Layer
model.add(Conv2D(filters=96, input_shape=(208,208,3), kernel_size=(11,11), strides=(4,4), padding='valid'))
model.add(Activation('relu'))
# Max Pooling
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='valid'))
# 2nd Convolutional Layer
#CNNs use 2D convolutional kernels to predict the segmentation map for a single slice.
model.add(Conv2D(filters=256, kernel_size=(11,11), strides=(1,1), padding='valid'))
model.add(Activation('relu'))
# Max Pooling
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='valid'))
# 3rd Convolutional Layer
model.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), padding='valid'))
model.add(Activation('relu'))
# Passing it to a Fully Connected layer
model.add(Flatten())
# 1st Fully Connected Layer
model.add(Dense(4096, input_shape=(300*300*3,)))
model.add(Activation('relu'))
# Add Dropout to prevent overfitting
model.add(Dropout(0.4))
# 2nd Fully Connected Layer
model.add(Dense(4096))
model.add(Activation('relu'))
# Add Dropout
model.add(Dropout(0.4))
# 3rd Fully Connected Layer
model.add(Dense(1000))
model.add(Activation('relu'))

```

```
+ Code + Text
model.add(Dense(4096))
model.add(Activation('relu'))
# Add Dropout
model.add(Dropout(0.4))
# 3rd Fully Connected Layer
model.add(Dense(1000))
model.add(Activation('relu'))
# Add Dropout
model.add(Dropout(0.4))
# Output Layer
model.add(Dense(2))
model.add(Activation('softmax'))
model.summary()

# Compile the model
model.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

D+ Model: "sequential"
Layer (type)          Output Shape         Param #
================================================================
conv2d (Conv2D)        (None, 50, 50, 96)    34944
activation (Activation) (None, 50, 50, 96)    0
max_pooling2d (MaxPooling2D) (None, 25, 25, 96) 0
conv2d_1 (Conv2D)       (None, 15, 15, 256)   2973952
activation_1 (Activation) (None, 15, 15, 256)   0
max_pooling2d_1 (MaxPooling2D) (None, 7, 7, 256) 0
conv2d_2 (Conv2D)       (None, 5, 5, 384)     885120
activation_2 (Activation) (None, 5, 5, 384)     0
```

```
+ Code + Text
[20]
activation_1 (Activation) (None, 15, 15, 256) 0
max_pooling2d_1 (MaxPooling2D) (None, 7, 7, 256) 0
conv2d_2 (Conv2D) (None, 5, 5, 384) 885120
activation_2 (Activation) (None, 5, 5, 384) 0
flatten (Flatten) (None, 9600) 0
dense (Dense) (None, 4096) 39325696
activation_3 (Activation) (None, 4096) 0
dropout (Dropout) (None, 4096) 0
dense_1 (Dense) (None, 4096) 16781312
activation_4 (Activation) (None, 4096) 0
dropout_1 (Dropout) (None, 4096) 0
dense_2 (Dense) (None, 1000) 4097000
activation_5 (Activation) (None, 1000) 0
dropout_2 (Dropout) (None, 1000) 0
dense_3 (Dense) (None, 2) 2002
activation_6 (Activation) (None, 2) 0
=====
Total params: 64,100,026
Trainable params: 64,100,026
Non-trainable params: 0

```

✓ 0s completed at 15:52

```
+ Code + Text
[21] train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)

[22] training_set=train_datagen.flow_from_directory('/content/drive/MyDrive/Dataset-20221013T025357Z-001/Dataset/train',target_size=(208,208),batch_size=32,class_mode='categorical')
test_set=test_datagen.flow_from_directory('/content/drive/MyDrive/Dataset-20221013T025357Z-001/Dataset/test',target_size=(208,208),batch_size=32,class_mode='categorical')

Found 387 images belonging to 2 classes.
Found 104 images belonging to 2 classes.

[23] img_dims = 150
epochs = 5
batch_size = 32

[24] #### Fitting the model
history = model.fit(
    training_set,
    steps_per_epoch=training_set.samples // batch_size,
    epochs=epochs,
    validation_data=test_set,validation_steps=test_set.samples // batch_size)

Epoch 1/19
12/12 [=====] - 49s 3s/step - loss: 4.2883 - accuracy: 0.5078 - val_loss: 0.7890 - val_accuracy: 0.4167
Epoch 2/19
12/12 [=====] - 4s 356ms/step - loss: 0.6695 - accuracy: 0.6094 - val_loss: 0.4339 - val_accuracy: 0.5833
Epoch 3/19
12/12 [=====] - 5s 413ms/step - loss: 0.0882 - accuracy: 0.9803 - val_loss: 5.9426e-06 - val_accuracy: 1.0000
Epoch 4/19
12/12 [=====] - 4s 331ms/step - loss: 8.0719e-07 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 5/19
12/12 [=====] - 4s 335ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 6/19
12/12 [=====] - 4s 342ms/step - loss: 4.5526e-05 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 7/19
12/12 [=====] - 4s 335ms/step - loss: 1.3738 - accuracy: 0.9324 - val_loss: 0.3615 - val_accuracy: 0.7188
Epoch 8/19
12/12 [=====] - 4s 338ms/step - loss: 0.0659 - accuracy: 0.9718 - val_loss: 1.2417e-06 - val_accuracy: 1.0000

```

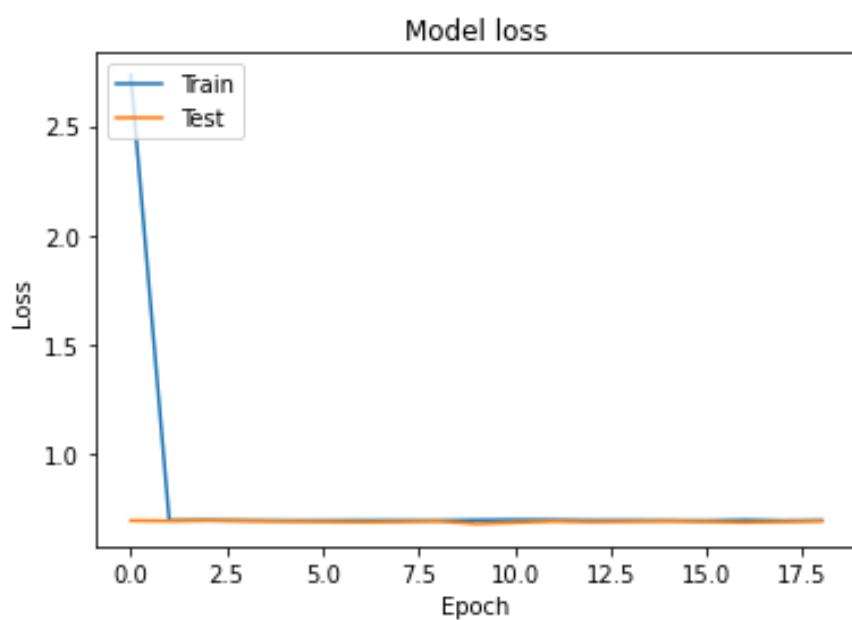
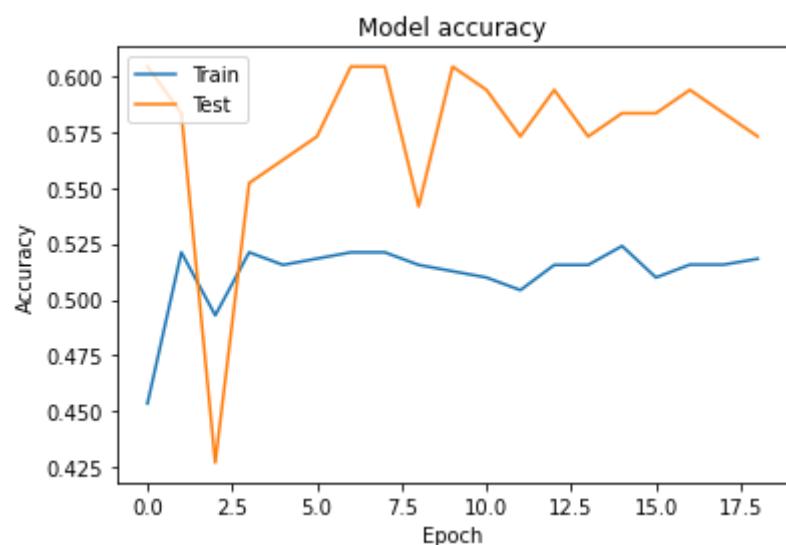
✓ 0s completed at 15:52

```
+ Code + Text
import matplotlib.pyplot as plt

[26] def graph():
    #Plot training & validation accuracy values
    plt.plot(history.history['accuracy'])
    plt.plot(history.history['val_accuracy'])
    plt.title('Model accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.show()

    # Plot training & validation loss values
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('Model loss')
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.show()
```

TESTING DATA:



MODULE - 3:

In this module, our CNN model classifies the MRI - image into two categories whether it is demented or non-demented. Here the machine predicts the result by using the trained data from the dataset. If it is shown as demanted it means that person has Alzheimer's disease. If it is shown as non-demanted it means that person is not having Alzheimer's disease.

```
[1] from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, EarlyStopping
[ ] from google.colab import drive
drive.mount('/content/drive')
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[2] from tensorflow.keras.models import Sequential
[3] from tensorflow.keras.layers import Convolution2D
[4] from tensorflow.keras.layers import MaxPooling2D
[5] from tensorflow.keras.layers import Flatten
[6] from tensorflow.keras.layers import Dense
[7] import warnings
warnings.filterwarnings('ignore')

[8] Classifier=Sequential()
```

```
+ Code + Text
[9] Classifier.add(Convolution2D(32,3,3,input_shape=(208,208,3),activation='relu'))
Classifier.add(MaxPooling2D(pool_size=(2,2)))
Classifier.add(Convolution2D(128,3,3,activation='relu'))
Classifier.add(MaxPooling2D(pool_size=(2,2)))
Classifier.add(Flatten())
Classifier.add(Dense(256, activation='relu'))
Classifier.add(Dense(2, activation='softmax'))

Classifier.compile(optimizer='rmsprop',loss='categorical_crossentropy',metrics=['accuracy'])
Classifier.summary()

Model: "sequential"
-----  
Layer (type)          Output Shape         Param #
-----  
conv2d (Conv2D)        (None, 69, 69, 32)    896  
max_pooling2d (MaxPooling2D) (None, 34, 34, 32)    0  
conv2d_1 (Conv2D)       (None, 11, 11, 128)   36992  
max_pooling2d_1 (MaxPooling 2D) (None, 5, 5, 128)    0  
flatten (Flatten)      (None, 3200)        0  
dense (Dense)          (None, 256)        819456  
dense_1 (Dense)        (None, 2)         514  
-----  
Total params: 857,858  
Trainable params: 857,858  
Non-trainable params: 0
```

```

[10] from tensorflow.keras.preprocessing.image import ImageDataGenerator
[11] train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
[12] test_datagen=ImageDataGenerator(rescale=1./255)

[ ] training_set=train_datagen.flow_from_directory('/content/drive/MyDrive/Dataset-20221013T025357Z-001/Dataset/train',target_size=(208,208),batch_size=32,class_mode='categorical')
    Found 387 images belonging to 2 classes.

[ ] test_set=test_datagen.flow_from_directory('/content/drive/MyDrive/Dataset-20221013T025357Z-001/Dataset/test',target_size=(208,208),batch_size=32,class_mode='categorical')
    Found 104 images belonging to 2 classes.

[ ] from IPython.display import display

[ ] img_dims = 150
[ ] epochs = 2
[ ] batch_size = 32

[ ] #pip install Pillow

[ ] from PIL import Image

```

```

+ Code + Text
[ ] Classifier.fit_generator( training_set, steps_per_epoch=training_set.samples // batch_size,
    epochs=epochs,
    validation_data=test_set,validation_steps=test_set.samples // batch_size)

Epoch 1/2
12/12 [=====] - 68s 6s/step - loss: 0.7197 - accuracy: 0.5625 - val_loss: 0.5286 - val_accuracy: 1.0000
Epoch 2/2
12/12 [=====] - 4s 321ms/step - loss: 0.3795 - accuracy: 0.8704 - val_loss: 0.1872 - val_accuracy: 0.9896
<keras.callbacks.History at 0x7fcf80124bd0>

[ ] import h5py

[ ] Classifier.save('alzhimer5.h5')

[ ] from keras.models import load_model

[ ] model=load_model('alzhimer5.h5')

[ ] import numpy as np

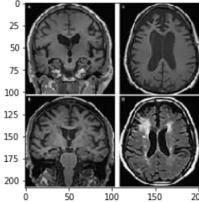
[ ] from tensorflow.keras.preprocessing import image
test_image=image.load_img('/content/drive/MyDrive/Dataset-20221013T025357Z-001/Dataset/test/Demented/mri1.jpg',target_size=(208,208))

```

```

+ Code + Text
[ ] from tensorflow.keras.preprocessing import image
test_image=image.load_img('/content/drive/MyDrive/Dataset-20221013T025357Z-001/Dataset/test/Demented/mri1.jpg',target_size=(208,208))

[ ] import matplotlib.pyplot as plt
img = plt.imshow(test_image)

[ ] test_image=image.img_to_array(test_image)

[ ] test_image=np.expand_dims(test_image,axis=0)

[ ] result=model.predict(test_image)

1/1 [=====] - 0s 16ms/step

[ ] prediction = result[0]

```

```
+ Code + Text
[ ] classes=training_set.class_indices
[ ] classes
{'Demented': 0, 'NonDemented': 1}
[ ] prediction=list(prediction)
[ ] prediction
[1.0, 5.6821294e-18]
[ ] classes=['Demented', 'NonDemented']
[ ] output=zip(classes,prediction)
[ ] output=dict(output)
[ ] output
{'Demented': 1.0, 'NonDemented': 5.6821294e-18}
[ ] if output['Demented']==1.0 :
    print("Demented")
elif output['NonDemented']==1.0:
    print("NonDemented")
Demented
```

CHAPTER 6

RESULTS AND DISCUSSION:

We evaluate various performance metrics like accuracy, precision, and loss. To determine the best we performed no of epochs so that for each epoch the accuracy got increased. We created a novel Machine Learning classifier for predicting and separating actual Alzheimer's disease-affected people from a given population, and we verified the model's ability to identify these individuals. For this analysis, we used these parameters to compute precision, accuracy, and loss metrics. The accuracy of an Alzheimer's diagnosis is measured by how often healthy individuals are accurately excluded from the diagnosis. On the other hand, accuracy measures how many people were correctly identified, whereas recall is the weighted average of these two metrics. Patients are given a report detailing their Alzheimer's disease stage based on the test results. Identifying the stages is crucial because the stages are determined by the patients' reactions.

The disease's progression can be better understood by doctors if they are aware of the patient's stage. The experiments and analyses in this study made use of the following environments, tools, and libraries:

- men are more prone than women to develop dementia or Alzheimer's disease.
- Dementia sufferers had less years of schooling than the general population.
- The volume of the brain is larger in people who do not have dementia compared to people Who do
- The percentage of patients aged 70-80 in the dementia group is much higher than that of the non-dementia group.

CHAPTER 7

CONCLUSION & FUTURE ENHANCEMENT :

We conducted a systematic assessment of the relevant literature and discovered that the majority of the articles in this field concentrate on biomarkers and neuroimaging, with an increasing emphasis on image processing. The majority of the selected patients already had a diagnosis of AD, hence the work didn't contribute much to the initial discovery of the disease. Some of the most relevant AD datasets, as well as diagnostic methods and detection strategies, were discussed in this article. For preliminary neuroimaging studies, this method is practical.

Predictions of the start of Alzheimer's disease can be made by ML systems using a feature selection and extraction technique using the oasis longitudinal dataset. This project will provide a high-level overview of the field by quickly discussing the several approaches utilized to analyze brain images to identify brain diseases. This research utilizes data from the reviewed literature to address some serious issues with applying machine learning and deep learning to diagnose brain illnesses. Findings from this study about the most effective method of detecting brain disorders can be utilized to create more precise diagnostic tools in the future.

REFERENCES

- [1] Early Alzheimer's Disease Detection Using Machine Learning, Year:2022, Author:-Carlos Andrés Tavera Romero, Osamah Ibrahim Khalaf, S. R. Srividhya, and C. Kavitha Vinodhini Mani.
- [2]Title: A Deep Learning Model to Predict a Diagnosis of Alzheimer's Disease by Using 18F-FDG PET of the Brain, Author: Yiming Ding; Jae Ho Sohn, MD, MS; Michael G. Kawczynski, MS; Hari Trivedi, MD, Year: 2013.
- [3]Title: DeepAD: Alzheimer's Classification with Deep Convolutional Neural Networks using MRI and fMRI, Author: Saman Sarraf; Danielle D.DeSouza, Year: 2016
- [4]Title: Classification of Alzheimer's disease in MobileNet, Author: Xiaoling Lu, Haifeng Wu * and Yu Zeng, Year: 2018
- [5]Title: Sorting out Alzheimer's Disease With Functional Magnetic Resonance Imaging Data and Deep Convolutional Neural Networks, Author: Saman Sarraf, Ghassem Tofighi, Year: 2016
- [6]Title:-New Multi-class Classification Approach for Alzheimer's Disease Diagnosis Using Brain MRI Data Based on Deep Learning, Author: Jyothi Islam, Yanqing ZHnag, Year:- 2017
- [7]Sivakasi GA, Ansari R. A framework for integrating machine learning for Alzheimer's disease. Int Conference Commun Signal Process.
- [8] Prajapati R, Khatri U, Kwon GR. "An effective deep neural network binary classifier for the classification of Alzheimer's disease" In Proceedings of the International Conference on Artificial Intelligence in Information and Communication (ICAC), pages 231–234, 2021.
- [9]Helaly HA, Badawy M, Haikal AY. An approach based on deep learning for early Alzheimer's disease identification. Cogn Computing.
- [10]Alzheimer's Disease and Related Disorders, Volume 33, Issue 2, Pages 87–94, 2019. All of us at J.-Y. Han, L. M. Besser, C. Xiong, W. A. Kukull, and J. C. Morris. It's possible that drugs that block cholinesterase won't help those with mild cognitive impairment or early Alzheimer's disease.

[11]Concepts of dementia and Alzheimer's disease throughout history, Article by Vatanabe, Manzini, and Cominetti, cited in Revue Neurologique, Volume 176, Issue 3, Pages 140-147, March 2020.

[12]Applications of magnetic resonance imaging and machine learning for the prognosis of Alzheimer's disease, N. 2018. Ph.D. dissertation by Bhagwat, Graduate Department of Biomedical Engineering, University of Toronto, Toronto, Ontario, Canada.

[13]Title: K. F. Hunter, M. Northwood, V. Haggar, and F. Bates's Management of Fecal Incontinence for the Advanced Practice Nurse. 127-148 in Springer (2018, cited).

[14]J. established a multimodal framework for cardiac computed tomography gating. Yao. Georgia Institute of Technology, School of Electrical and Computer Engineering, Atlanta, Georgia, USA, 2018.

[15]"Assemblies of patch-based classifiers for Alzheimer disease diagnosis," S. Choi, J. J. Lee, B. C. Kim, G.-R. Kwon, K. H. Lee, and H. Ahmed Jung, Y., IEEE Access, volume 7, pp. 73373-73383, 2019.

APPENDIX A

CONFERENCE PRESENTATION

Paper is prepared and submitted to IEEE but Paper acceptance is not yet received.

APPENDIX B

PUBLICATION DETAILS

Paper is submitted to a conference, acceptance is pending.

Not secure | cmswebonline.com/iitcee/index.php/iitcee/author

The screenshot shows the CMS Web Online platform interface. At the top, there's a banner for the 'International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics' held on 27-28 January 2023 at BNM Institute of Technology, Bengaluru, India. The banner also features logos for IEEE Bangalore Section and Vidyaamruthamashnute. Below the banner, there's a navigation bar with 'USER HOME' and 'LOG OUT' links. The main content area is titled 'Active Submissions' and has tabs for 'ACTIVE' and 'ARCHIVE'. A table lists one submission: ID 306, DD-MM SUBMIT 04-11, TRACK SS20, AUTHORS ALLA, CHEDURUPALLI, K.R., TITLE 'APPLYING DEEP LEARNING TO THE PROBLEM OF CATEGORIZING...', and STATUS 'Awaiting assignment'. There are buttons for 'UPLOAD REVISED PAPER' and 'DELETE'.

ID	DD-MM SUBMIT	TRACK	AUTHORS	TITLE	STATUS
306	04-11	SS20	ALLA, CHEDURUPALLI, K.R.	APPLYING DEEP LEARNING TO THE PROBLEM OF CATEGORIZING...	Awaiting assignment UPLOAD REVISED PAPER DELETE

1 - 1 of 1 Items

APPENDIX C

PLAGIARISM REPORT

