

EARLY DIAGNOSIS OF ALZHEIMER'S DISEASE USING DEEP LEARNING

Submitted in partial fulfillment of the requirements for the award of Bachelor
of Engineering Degree in Computer Science and Engineering

By

INBASAGAR D (38110205)
HRITHIK R (38110201)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

MAY-2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **INBASAGAR D (Reg. No: 38110205), HRITHIK R (Reg. No: 38110201)** who carried out the project entitled "**EARLY DIAGNOSIS OF ALZHEIMER'S DISEASE USING DEEP LEARNING**" under my supervision from June 2021 to November 2021.

Internal Guide

Dr. M. MAHESWARI M.E., Ph.D.,

Head of the Department

Dr.L. LAKSHMANAN M.E.,Ph.D.,

Dr. S. VIGNESHWARI M.E., Ph.D.,

Submitted for Viva Voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, **INBASAGAR D** and **HRITHIK R** hereby declare that the project report entitled **EARLY DIAGNOSIS OF ALZHEIMER'S DISEASE USING DEEP LEARNING** done by me under the guidance of **Dr. M. Maheswari M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, Dean, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D.** and **Dr. L. Lakshmanan, M.E., Ph.D.**, Heads of the **Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. M. Maheswari M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

There are many different causes of dementia, but Alzheimer's Disease is the most usual form. As the condition progresses, it limits one's ability to perform any task without aid, and the diagnosis timeline and aging population are expected to cause its prevalence to increase. The conventional ways of detecting Alzheimer's is tiring for both patients, doctors where it involves retrieving the past medical records and having Magnetic Resonance Imaging scans and even neurophysical testing which can be inconvenient for patients. An early diagnosis of brain diseases makes a big difference when it comes to attempting to cure them. Our work has used deep learning (neural networks) to detect Alzheimer's disease earlier than usual by combining it with deep learning. As the obtained dataset from Kaggle is heavily imbalanced, we evenly distributed the data between the categories using SMOTE. Then the model is trained and tested with the categorized MRI data i.e. very mild, mild, moderate and severe AD and finally extract features to examine the results. The results we achieved are compared with the previous attempts on detection of Alzheimer's and came out to be significantly greater in terms of precision and accuracy.

	TABLE OF CONTENTS	
CHAPTER No.	TITLE	PAGE No
	ABSTRACT	v
	LIST OF FIGURES	ix
1	INTRODUCTION	1
	1.1 Overview	3
	1.2 Objective	3
2	LITERATURE SURVEY	4
3	AIM AND SCOPE OF PRESENT INVESTIGATION	7
	3.1 Aim	7
	3.2 SCOPE OF PRESENT INVESTIGATION	7
	3.3 Existing System	8
	3.3.1 Disadvantages	8
	3.4 Proposed System	8
	3.4.1 Advantages	9

4	EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED	10
	4.1 Deep Learning Overview	10
	4.2 Steps to Download & Install Python	10
	4.3 Python Libraries Needed	11
	4.3.1 NumPy Library	11
	4.3.2 Pandas Library	11
	4.3.3 Matplotlib Library	11
	4.3.4 Seaborn Library	12
	4.3.5 Pillow Library	12
	4.3.6 TensorFlow Library	12
	4.3.7 Keras Library	13
	4.3.8 Scikit-Learn Library	13
	4.4 Modules Implementation	13
	4.4.1 Data Pre-processing	14

	4.4.2 Data Validation/ Data Cleaning/ Preparing Process	16
	4.4.3 Exploration Data Analysis Of Visualization	16
	4.4.4 Comparing Algorithm with prediction in the form of best accuracy result	17
	4.5 Algorithms and Techniques	20
	4.5.1 Algorithm Explanation	20
	4.5.2 Used Python Packages	21
	4.5.3 Neural Networks	21
	4.5.4 Convolution Neural Network	22
	4.5.5 Optimization Algorithm	28
	4.6 Model Architecture	28
	4.7 Workflow Diagram	29
5	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	30
	5.1 Results	30
	5.2 Performance Analysis	30
6	SUMMARY AND CONCLUSION	31
	6.1 Summary	31
	6.2 Conclusion	31
	REFERENCES	32
	APPENDIX	36
	A. SCREENSHOTS	36
	B. SOURCE CODE	41
	C. PLAGIARISM REPORT	45

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.1	Pre-Processing Diagram	16
4.2	Data Visualization Diagram	17
4.3	CNN Architecture Representation	23
4.4	LeNet Architecture Representation	25
4.5	VGG-16 Architecture Representation	26
4.6	Inception Architecture Representation	26
4.7	ResNet Architecture Representation	27
4.8	Model Architecture	28
4.9	Workflow Diagram	29

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
4.1	Dataset Distribution in Obtained Dataset	15
4.2	Dataset Distribution after Applying SMOTE	16
5.1	Evaluation Metrics of Individual Class	30
5.2	Performance Analysis	30

CHAPTER-1

INTRODUCTION

A person with Alzheimer's Disease (AD) suffers from a developing neurobiological disorder that affects brain cells to die & atrophy. AD is the prevalent cause of dementia, which causes memory loss and impaired reasoning abilities. Around 6 million people in the U.S of age 65 and above are caused with Alzheimer's disease. Of those 80 percent are above 75 years old. In India it is calculated around 5.3 million people live with dementia of which Alzheimer is the common cause.

Dementia with Alzheimer's is classified into four categories:

- a. Very Mild Dementia: Individuals suffers from memory loss as they age.
- b. Mild Dementia: Symptoms which includes lack of memory, Behavioral changes, inability to perform routine tasks.
- c. Moderate Dementia: The day to day life becomes complex for the individuals with moderate dementia, where the patients require extra care and support.
- d. Severe Dementia: The patients in this stage may not able to communicate properly, and they require medical care. One may lose physical control.

Alzheimer's is not a curable disease, but an early diagnosis can help prevent the patient from suffering from the later stages. In order to diagnose AD manual detection systems for example: Positron Emission Tomography (PET) were used to track the progression of the various stages of AD, MRI scans and genotype sequencing results were taken for diagnosis.

Among the most popular fields of research in recent years has been brain-computer interface (BCI), thanks to its applications in areas such as brain fingerprinting, detecting neurological illnesses, tiredness, adaptive e-learning and more. By extracting the most significant characteristics, BCI creates an effective link to interact between the brain and the device. A complex brain structure varying with age and pathology makes it very difficult to detect neurodegenerative diseases in their early stages. Computer-assisted techniques are more successful in detecting these disorders than traditional

approaches. A timely diagnosis and identifying of Alzheimer's disease is essential to reducing medical expenses, improving treatment, and preventing brain cell degeneration.

Some of the methods included in an early examination include Positron Emission Tomography (PET), digital imaging methods and genotyping-by-sequencing. It is difficult to take decisions by analyzing different methods. Furthermore, the patients will have to undergo radioactive effects during PET medical procedure. According to our findings, MRIs can provide valuable information on the brain, because they provide flexible imaging, superior tissue contrast, and do not expose the brain to ionizing radiation.

It is crucial to develop a model that can take MR images as input and detect whether patients are normal or not. By utilizing a dataset, machine learning can extract knowledge. Computer science, artificial intelligence, and statistics combine to make up this field.. The ML is done through training a computer to produce the output based on its past experience to solve a given problem. Machine learning can be applied in a variety of fields in order to solve problems quicker than humans, and therefore be more efficient, and reduce time spent on repetitive tasks. Nowadays, because of the reduction in the cost of computing power and memory. This allows processing and analyzing huge amounts of data to generate insights.

Additionally, Deep Learning is a subset of ML and an advanced mode of analyzing and learning information from raw data that computers are able to replicate, much like how humans are able to do, with a computer. Deep learning is becoming increasingly popular for diagnosing diseases. Several Machine learning approaches have been proposed recently to aid in this diagnosis, providing doctors with more information to make informed decisions.

We use Deep Neural Networks (DNNs) for feature extraction using deep learning techniques in the proposed model. A solution to underfitting is to use sampling techniques especially oversampling to resolve class imbalance. DL performs classification on given MR images using the cortical surface of the brain as input.

Using a dementia-specific (Alzheimer's) dataset, the models are evaluated by

NonDemeneted, Moderate Dementia, Mild Dementia and Very Mild Dementia obtained from Kaggle. Utilizing the CNN technique, we extract discriminating features for AD classification by improving the accuracy. By using this model, we can accurately classify the stages of AD.

1.1 Overview:

Dementia is a term for a decline in mental ability severe enough to interfere with daily life. Alzheimer's is the one type of dementia. Alzheimer's is the most serious yet common neurodegenerative disease that initially destroys cells of the part of the brain. It's responsible for language and memory resulting in memory loss of the patient and also the ability to perform regular tasks.

As there is no cure for Alzheimer's disease, it's better to detect as early as possible to slow down the severity of the disease. Usually to diagnose the disease radiologists use manual methods such as previous medical history, continuous monitoring of the patient to detect the various stages of AD, however these manual methods may lead to errors!

1.2 Objective:

Implementing Deep Learning algorithm efficiently to identify the stage of the Alzheimer's disease patient. Analyzing the various performance metrics of the deep learning algorithm.

CHAPTER-2

LITERATURE SURVEY

Researchers have made several attempts to classify Alzheimer's patients using the brain structure information obtained from MR images. Tested the effectiveness of a method using brain shape information for classification of healthy subjects and Alzheimer's disease patients. Here, the shape information of the brain, particularly that of the lateral ventricles, was analyzed without considering the septum lucidum.[1].

The neurodegenerative disease is currently the 6th leading source of death in the US. In 2017 this disease costed the nation \$1.1 trillion. This study have shown that deep CNNs are capable of detecting Alzheimer's disease and dementia based on 3D MRI imaging and can be used to detect the disease. As of the this papers writing, detecting Alzheimer's is a difficult and time consuming task, but requires brain imaging report and human expertise. Needless to say, this conventional approach to detect Alzheimer's is costly and often error prone. In this paper an alternative approach has been discussed, that is fast, costs less and more reliable. Deep Learning represents the true bleeding edge of Machine Intelligence. Convolutional Neural Networks are biologically inspired Multilayer perceptron specially capable of image processing[2].

Neelaveni J et al. [3] used machine-learning algorithms, they used psychological factors to predict Alzheimer's, such as age, MMSE scores, and educational level. Diagnosis of the disease is done but that too at the later stage only. If the disease can be predicted earlier, symptoms or progression can be slowed down.

They demonstrate how whole-brain volume can be automatically calculated and estimated spinal fluid volume used for determining differences associated with ideal aging and Alzheimer's disease[4].

It is well-established, widely used, and non-invasive to use structural MR imaging in Alzheimer's disease studies of downstream effect of the neurodegeneration, or atrophy. In peer-reviewed journals, the imaging and cohort results have been published in detail, and the data have been made freely available[5].

Zhao Fan et al. [6], help aid auxiliary diagnosis of Alzheimer's disease, with the help of SVM model to analyze and classify structural brain MR imaging data, in order to achieve much greater classification predictions, the extracted MRI detail is combined with the SVM model. The accuracy of classification and prediction is the best. According to the predicted results, the data characteristics related to diseases can be determined, which can provide a basis for clinical and basic research, etiology and pathological changes.

Acharya et al, used several feature extraction techniques along with MR imaging for classification, the system uses several algorithms for Computer-Assisted-BrainDiagnosis (CABD) to assess if the scan image shows indications of Alzheimer's disease. The paradigm consists of a series of quantitative techniques: filtering, feature extraction, Student's t-test based feature selection, and k-Nearest Neighbor (KNN) based classification. Additionally, a comparative analysis is done by implementing other feature extraction procedures that are described in the literature. According to their results, Shearlet Transform (ST) feature extraction is superior to alternative methods for detecting Alzheimer's disease[7].

Chima Stanley Eke et al. [8] the main basis for their method is machine learning(ML) techniques (specifically support vector machines), which are capable of learning patterns from complex data in order to create multi-variable models. The objective in this study is to develop a method to identify potential blood-based non-amyloid biomarkers for early AD detection. The use of blood is attractive because it is accessible and relatively inexpensive. On the basis of the identified panels, disease detection models were determined to have a sensitivity greater than 80%, a specificity greater than 70%, and AUC greater

than 0.80 at the prodromal stage of the disease.

R. Jain et al. [9] proposed convolutional neural network model (CNNs) were used to locate gray matter in brain voxels and to segment segmented gray matter for clinical appraisal. VGG-16 trained on ImageNet dataset is used as a feature extractor for the classification task. Experimentation is performed on data collected from Alzheimer's Disease Neuroimaging Initiative (ADNI) database. The proposed methodology achieved an accuracy rate of 90.47% for segments of gray matter for clinical evaluation.

Chenjie Ge et al. [10] the paper showcases, an innovative idea to detect Alzheimer's disease (AD) using 3D multiscale neural network architecture. They proposed a multiscale deep learning architecture for learning AD features. The main contributions of the paper include, proposing a novel 3D multiscale CNN architecture for the dedicated task of AD detection, secondly proposing a feature fusion and enhancement strategy for multiscale features and at last empirical study on the impact of several settings, including two dataset partitioning approaches, and the use of multiscale and feature enhancement. It successfully achieved an accuracy of 87.24% for detection.

Danail Stoyanov et al. [11] used brain structural MRI scans, they developed a 3D CNN to identify Alzheimer's disease. They highlighted relevant areas of the network using four distinct ways to visualize gradients and occlusions. Their neural network model achieved 77% accuracy.

According to a paper by K.S. Biju et al. [12], the authors presented a mathematical model that uses MRI slices to create a 3 dimension representation of the brain that can indicate whether Alzheimer's disease is present. A number of processing steps are carried out on MRI slices, including up sampling, 3D construction. This 3D image is inspected by a medical professional.

CHAPTER-3

AIM AND SCOPE OF PRESENT INVESTIGATION

3.1 AIM:

This project aims to detect the stages of the Alzheimer's disease by using deep learning techniques.

3.2 SCOPE OF THE PRESENT INVESTIGATION:

Alzheimer's disease is a progressive neurological disorder that causes the brain to shrink (atrophy) and brain cells to die. Alzheimer's disease is the most common cause of dementia. As the disease progresses, a person with Alzheimer's disease will develop severe memory impairment and lose the ability to carry out everyday tasks. A number of medicines may be prescribed for Alzheimer's disease to help temporarily improve some symptoms but there is no permanent cure. An accurate, timely diagnosis gives you the best chance to adjust, prepare and plan for the future, as well as access to treatments and support that may help. An early detection can give a patient a better chance to cure and recover. It is crucial to develop a model that can take MR images as input and detect whether patients are normal or not. Computer science, artificial intelligence, and statistics combine to make up this field. been applied in the healthcare sector. Deep Learning is a subset of ML and an advanced mode of analyzing and learning information from raw data that computers are able to replicate, much like how humans are able to do, with a computer. Deep learning is becoming increasingly popular for diagnosing diseases. We use Deep Neural Networks (DNNs) for feature extraction using deep learning techniques in the proposed model. A solution to underfitting is to use sampling techniques especially oversampling to resolve class imbalance. DL performs classification on given MR images using the cortical surface of the brain as input.

3.3 EXISTING SYSTEM:

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. In Existing system machine learning algorithms are used to predict the Alzheimer disease using psychological parameters like age, number of visit, MMSE and education. Different modalities are used for AD study include MRI, Positron Emission Tomography (PET), and genotype sequencing results. It is time-consuming to analyze different modalities to take a decision. Furthermore, the patients can encounter radioactive effects in the modalities like PET. Previously researchers performed 3D tissue segmentation of white matter, gray matter, and cerebrospinal fluid from MR images after skull stripping using FSL tool, calculate the surface fractal dimension from segmented brain tissue. From the survey, Numerous techniques exist for AD classification using machine and deep learning. However, the high model parameter and class imbalance in the multiclass AD classification is still an issue.

3.3.1 Disadvantages:

- The existing model shows significant accuracy only when MMSE score, education, etc., is given.
- The patients can encounter radioactive effects in the modalities like PET.
- 3D MRI scan is hard to train and time consuming process.

3.4 PROPOSED SYSTEM:

It is considered important to develop a better computer-aided diagnostic system that can interpret MRI imaging and determine whether patients are healthy or have Alzheimer's disease. Conventional deep learning systems use the cortical surface to input the CNN to perform AD classification on raw MRI images. In this proposed work, We believe that the MRI modality benefits from its greater imaging flexibility, excellent tissue contrast, lack of ionizing radiation, and ability to provide useful information on human brain anatomy. This paper proposes a model that uses the convolutional neural network to extract the discriminative features. Class imbalance is addressed using the Synthetic Minority Oversampling Technique (SMOTE) technique. The model is developed from scratch to classify the stages of AD more accurately by reducing its parameters and computation cost. The models are evaluated by training them over the MRI dataset from the Kaggle. The dataset comprises four types of dementia such as Mild Dementia (MID), Moderate Dementia (MOD), Non-Demented (ND) and

Very Mild Dementia (VMD).

A new convolutional neural network architecture is proposed with relatively small parameters to detect the types of dementia which is suitable for training a smaller dataset

SMOTE technique is used to address the class imbalance problem in the dataset is by randomly duplicating the minority class of images in the dataset to minimize the overfitting problem.

We created the generalized model that learns from the smaller dataset with reduced parameters and computation cost, which still performs better for AD diagnosis.

We also compared the proposed model with deep features and hand-crafted features to detect AD stages in terms of Accuracy, AUC and Cohen's kappa score.

3.4.1 Advantages:

- A neural network architecture is proposed with relatively small parameters to detect the types of dementia which is suitable for training a smaller dataset
- SMOTE technique is used to address the class imbalance problem.
- We created the generalized model that learns from the smaller dataset with reduced parameters and computation cost.

CHAPTER 4

EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED

4.1 Deep Learning Overview:

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. It is essentially a neural network with three or more layers, these neural networks attempts to mimic the human brain through a combination of data inputs, weights, and bias, allowing it to learn from large amounts of data. These elements work together to accurately recognize, classify, and describe objects within the data.

There are different types of neural networks to address specific problems or datasets. For example, Convolutional neural networks (CNNs), used primarily in computer vision and image classification applications, can detect features and patterns within an image, enabling tasks, like object detection or recognition. Recurrent neural network (RNNs) are typically used in natural language and speech recognition applications as it leverages sequential or times series data.

4.2 Steps to Download & Install Python:

Download the Latest version of the **Python** executable installer

(<https://www.python.org/downloads/>). Watch the PIP list where pip is the package installer for python. Now upgrade the pip and setup tools using the command

```
pip install --upgrade pip and pip install --upgrade setuptools
```

ENVIRONMENT INSTALLATION FOR PYTHON

Jupyter Notebook is an open-sourced web-based application which allows you to create and share documents containing live code, equations, visualisations, and narrative text. It is a server-client application that allows editing and running notebook documents via a web browser. The Jupyter Notebook App can be executed on a local desktop requiring no internet access.

Installation- (<https://jupyter.org/install/>)

```
pip install notebook
```

4.3 PYTHON LIBRARIES NEEDED

There are many libraries in python. In those, we only use a few main libraries needed

4.3.1 NUMPY LIBRARY

NumPy is an open-source numerical Python library. NumPy contains a multi-dimensional array and matrix data structures. It can be utilized to perform a number of mathematical operations on arrays such as trigonometric, statistical, and algebraic routines like mean, mode, standard deviation, etc.,

Installation- (<https://numpy.org/install/>)

Here we mainly use an array, to find the mean and standard deviation.

4.3.2 PANDAS LIBRARY

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the Data Frame. Data Frames allow you to store and manipulate tabular data in rows of observations and columns of variables. There are several ways to create a DataFrame.

Installation- (https://pandas.pydata.org/getting_started.html/)

```
pip install pandas
```

4.3.3 MATPLOTLIB LIBRARY

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Use interactive figures that can zoom, pan, update, visualize etc.,

Installation- (<https://matplotlib.org/users/installing.html/>)

```
pip install matplotlib
```

Here we use pyplot mainly for plotting graphs.

matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

4.3.4 SEABORN LIBRARY

Seaborn package was developed based on the Matplotlib library. It is used to create more attractive and informative statistical graphics. While seaborn is a different package, it can also be used to develop the attractiveness of matplotlib graphics.

Installation-(<https://seaborn.pydata.org/installing.html/>)

```
pip install seaborn
```

4.3.5 PILLOW LIBRARY

The Python Imaging Library adds image processing capabilities to your Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities. The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

Installation-(<https://pillow.readthedocs.io/en/stable/installation.html/>)

```
pip install pillow
```

4.3.6 TENSORFLOW LIBRARY

TensorFlow is a framework created by Google for creating Deep Learning models. Deep Learning is a category of machine learning models (=algorithms) that use multi-layer neural networks. TensorFlow is a great tool which, if used properly has innumerable benefits. The major uses of the library include classification, perception, understanding, discovering, prediction and creation.

Installation-(<https://www.tensorflow.org/install/>)

```
pip install tensorflow
```

4.3.7 KERAS LIBRARY

Keras is a powerful and easy-to-use free open source Python library for developing and evaluating deep learning models. It is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error.

Installation-(https://keras.io/getting_started/)

```
pip install keras
```

4.3.8 SCIKIT-LEARN LIBRARY

Scikit-learn is a free machine learning library for the Python. It features various algorithms like support vector machine, random forests, regression and k-neighbours, and it also supports Py(thon numerical and scientific libraries like NumPy and SciPy. Installation-(<https://scikit-learn.org/stable/install.html/>)

```
pip install scikit-learn
```

Here use scikit-learn's regression methods for prediction purpose.

4.4 Modules Implementation:

A modular design reduces complexity, facilitates change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different part of system. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Software architecture embodies modularity that is software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements.

Modularity is the single attribute of software that allows a program to be intellectually manageable. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular

decomposability, Modular Comps ability, Modular Understand ability, Modular continuity, Modular Protection.

The following are the modules of the project, which is planned in aid to complete the project with respect to the proposed system, while overcoming existing system and also providing the support for the future enhancement.

4.4.1 Data Pre-processing

The dataset is derived from the Kaggle an open source platform, which contains around 6500 images comprising of four classes namely moderately demented, very mildly demented and non-demented, mildly demented. The dimension of images used in the dataset is of form 176x208. The images are reshaped into 180x180. The SMOTE technique is applied to the dataset to solve the class imbalance problem in the dataset by randomly duplicating minority classes in the dataset to match the majority classes [33]. With the random seed of 42, the minority classes oversampled using SMOTE technique. The benefits of using SMOTE include the ability to reduce knowledge loss and minimize over-fitting. Table 2 shows the dataset distribution after SMOTE technique increased to 12800 images, with each class contains 3200 images. It as machine learning engineers use this data to fine- tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model.

A number of different data cleaning tasks using Python Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend lesstime cleaning data, and more time exploring and modeling.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it's important to understand the sources of missing data. Here are some

typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- Due to Non-availability of data.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

- import libraries for access and functional purpose and read the given dataset
- General Properties of Analyzing the given dataset
- Display the given dataset in the form of data frame
- show columns
- shape of the data frame
- To describe the data frame
- Checking data type and information about dataset
- Checking for duplicate data
- Checking Missing values of data frame
- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To create extra columns

Table 4.1 Dataset Distribution in Obtained Dataset

Class	Number of Images
Non-Demented	2240
Moderate Demented	3200
Mild Demented	896
Very Mild Demented	64

Table 4.2 Dataset Distribution after applying SMOTE

Class	Number of Images
Non-Demented	3200
Moderate Demented	3200
Mild Demented	3200
Very Mild Demented	3200

4.4.2 Data Validation/ Cleaning/ Preparing Process

Importing the library packages with loading the given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process.

The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.



Fig. 4.1 : Pre-Processing Diagram

GIVEN INPUT EXPECTED OUTPUT

Input: data.

Output: removing noisy data, Solve class imbalance problem.

4.4.3 Exploration data analysis of visualization

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with

identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

- 4 How to chart time series data with line plots and categorical quantities with bar charts.
- 5 How to summarize data distributions with histograms and box plots.

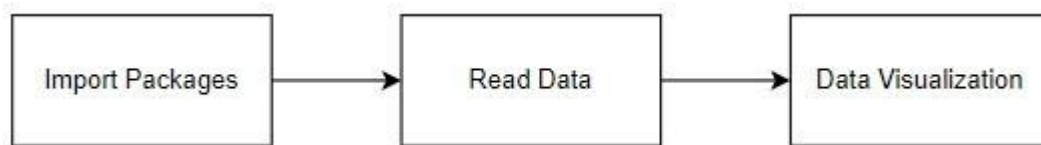


Fig. 4.2 : Data Visualization Diagram

GIVEN INPUT EXPECT OUTPUT

Input: data

Output: visualized data

4.4.4 Comparing Algorithm with prediction in the form of best accuracy result

It is important to compare the performance of multiple different deep learning algorithms consistently and it will discover to create a test harness to compare multiple different deep learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own problems and add more and different algorithms to compare. Each models will have different performance characteristics.

Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different

techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance, and other properties of the distribution of model accuracies.

In the next section, you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieve better results from the applied model the data has to be in a proper manner. Different deep learning algorithms are executed in a given dataset. In the example below these 2 different algorithms are compared with our algorithm:

- ResNet50
- Inception V3

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly

predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

True Positive Rate(TPR) = $TP / (TP + FN)$

False Positive Rate(FPR) = $FP / (FP + TN)$

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:

Accuracy = $(TP + TN) / (TP + TN + FP + FN)$

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct.

Precision = $TP / (TP + FP)$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

Recall = $TP / (TP + FN)$

Recall(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to

understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

General Formula:

$$F\text{- Measure} = 2TP / (2TP + FP + FN)$$

F1-Score Formula:

$$F1 \text{ Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

4.5 ALGORITHM AND TECHNIQUES

4.5.1 Algorithm Explanation

Deep learning algorithms play a crucial role in determining the features and can handle the large number of processes for the data that might be structured or unstructured. Although, deep learning algorithms can overkill some tasks that might involve complex problems because they need access to huge amounts of data so that they can function effectively. Deep learning algorithms are highly progressive algorithms that learn about the image that we discussed previously by passing it through each neural network layer. The layers are highly sensitive to detect low-level features of the image like edges and pixels and henceforth the combined layers take this information and form holistic representations by comparing it with previous data. Convolutional neural network (CNN) popularly known as ConvNets majorly consists of several layers and are specifically used for image processing and detection of objects.

CNNs process the data by passing it through multiple layers and extracting features to exhibit convolutional operations. The Convolutional Layer consists of Rectified Linear Unit (ReLU) that outlasts to rectify the feature map. The Pooling layer is used to rectify these feature maps into the next feed. Pooling is generally a sampling algorithm that is down-sampled and it reduces the dimensions of the feature map. Later, the result generated consists of 2-D arrays consisting of single, long, continuous, and linear vector flattened in the map. The next layer i.e., called Fully Connected Layer which forms the flattened matrix or 2-D array fetched from the Pooling Layer as input and identifies the image by classifying it.

4.5.2 Used Python Packages:

Sklearn:

- In python, sklearn is a machine learning package which include a lot of ML algorithms.
- Here, we are using some of its modules like train_test_split, DecisionTreeClassifier or Logistic Regression and accuracy_score.

NumPy:

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

Pandas:

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

Matplotlib:

- Data visualization is a useful way to help with identify the patterns from given dataset.
- Data manipulation can be done easily with data frames.

4.5.3 Neural Networks

Neural networks are artificial systems that were inspired by biological neural networks. These systems learn to perform tasks by being exposed to various datasets and examples without any task-specific rules. The idea is that the system generates identifying characteristics from the data they have been passed without being programmed with a pre-programmed understanding of these datasets.

Components of a typical neural network involve neurons, connections, weights, biases, propagation function, and a learning rule. Neurons will receive an input $p_{j(t)}$ from predecessor neurons that have an activation $\alpha_j(t)$, threshold j , an activation function f , and an output function f_{out} . Connections consist of connections, weights and biases which rules how neuron transfers output to neuron . Propagation computes the input and outputs the output and sums the predecessor neurons function with the weight. The learning rule modifies the weights and thresholds of the variables in the network.

The most well-known and simplest-to-understand neural network is the feedforward

multilayer neural network. It has an input layer, one or many hidden layers, and a single output layer. Each layer can have a different number of neurons and each layer is fully connected to the adjacent layer.

Types of Neural Networks

There are seven types of neural networks that can be used.

- The first is a multilayer perceptron which has three or more layers and uses a nonlinear activation function.
- The second is the convolutional neural network that uses a variation of the multilayer perceptrons.
- The third is the recursive neural network that uses weights to make structured predictions.
- The fourth is a recurrent neural network that makes connections between the neurons in a directed cycle. The long short-term memory neural network uses the recurrent neural network architecture and does not use activation function.

4.5.4 Convolutional Neural Network

A convolutional neural network, or CNN, is a deep learning neural network designed for processing structured arrays of data such as images. The CNN architectures are the most popular deep learning framework. CNNs are used for a variety of applications, ranging from computer vision to natural language processing. Convolutional neural networks can operate directly on a raw image and do not need any preprocessing.

A convolutional neural network is a feed-forward neural network, often with up to 20 or 30 layers. The power of a convolutional neural network comes from a special kind of layer called the convolutional layer.

The architecture of a convolutional neural network is a multi-layered feed-forward neural network, made by stacking many hidden layers on top of each other in sequence. It is this sequential design that allows convolutional neural networks to learn hierarchical features.

The hidden layers are typically convolutional layers followed by activation layers, some of them followed by pooling layers.

The key building block in a convolutional neural network is the convolutional layer. We can

visualize a convolutional layer as many small square templates, called convolutional kernels, which slide over the image and look for patterns. Where that part of the image matches the kernel's pattern, the kernel returns a large positive value, and when there is no match, the kernel returns zero or a smaller value.

Convolutional networks are used for alternating between convolutional layers and max-pooling layers with connected layers (fully or sparsely connected) with a final classification layer. The learning is done without unsupervised pre-training. Each filter is equivalent to a weights vector that has to be trained.

The diagram below represents CNN architecture.

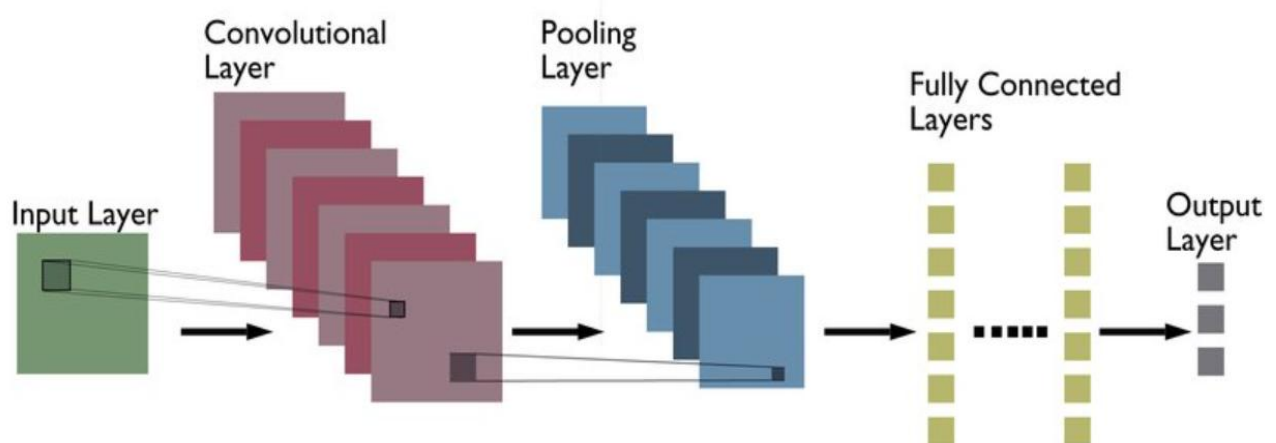


Fig. 4.3 : CNN Architecture Representation

The following are definitions of different layers shown in the above architecture:

- **Input Layer:** As input, normalized MRI image datasets are given as the first layer of the presented neural network model
- **Convolutional layer:** Convolutional layers are made up of a set of filters (also called kernels) that are applied to an input image. The output of the convolutional layer is a feature map, which is a representation of the input image with the filters applied. Convolutional layers can be stacked to create more complex models, which can learn more intricate features from images.
- **Pooling layer:** Pooling layers are a type of convolutional layer used in deep learning. Pooling layers reduce the spatial size of the input, making it easier to process and requiring less memory. Pooling also helps to reduce the number of parameters and makes training faster. There are two main types of pooling: max pooling and average pooling. Max pooling takes the maximum value from each feature map, while average

pooling takes the average value. Pooling layers are typically used after convolutional layers in order to reduce the size of the input before it is fed into a fully connected layer.

- Fully connected layer: Fully-connected layers are one of the most basic types of layers in a convolutional neural network (CNN). As the name suggests, each neuron in a fully-connected layer is Fully connected- to every other neuron in the previous layer. Fully connected layers are typically used towards the end of a CNN- when the goal is to take the features learned by the previous layers and use them to make predictions. For example, if we were using a CNN to classify images of animals, the final Fully connected layer might take the features learned by the previous layers and use them to classify an image as containing a dog, cat, bird, etc.
- Dense Layer: By using flatten layer, high-dimensional data becomes column vectors, which follows the convolutional layer, and then there is a dense layer, which does the same functions as a neural network.
- Dropout Layer: The dropout is regularization technique for neural networks. During training, neurons in the hidden layers are randomly dropped out of the dropout layer. By eliminating some of the neurons, this layer minimizes overfitting in the model.

The following is a list of different types of CNN architectures:

LeNet:

The LeNet architecture is an excellent “first architecture” for Convolutional Neural Networks (especially when trained on the MNIST dataset, an image dataset for handwritten digit recognition). LeNet is small and easy to understand — yet large enough to provide interesting results. Furthermore, the combination of LeNet + MNIST is able to run on the CPU, making it easy for beginners to take their first step in Deep Learning and Convolutional Neural Networks.

The main reason behind the popularity of this model was its simple and straightforward architecture. It is a multi-layer convolution neural network for image classification.

The LeNet5 network has 5 layers with learnable parameters and hence named Lenet-5. It has three sets of convolution layers with a combination of average pooling. After the convolution and average pooling layers, we have two fully connected layers. At last, a Softmax classifier which classifies the images into respective class.

The input to this model is a 32 X 32 grayscale image hence the number of channels is one. We then apply the first convolution operation with the filter size 5X5 and we have 6

such filters. As a result, we get a feature map of size $28 \times 28 \times 6$. Here the number of channels is equal to the number of filters applied.

After the first pooling operation, we apply the average pooling and the size of the feature map is reduced by half.

Then we have a final convolution layer of size 5×5 with 120 filters. As shown in the above image. Leaving the feature map size $1 \times 1 \times 120$. After which flatten result is 120 values. After these convolution layers, we have a fully connected layer with eighty-four neurons. At last, we have an output layer with ten neurons since the data have ten classes.

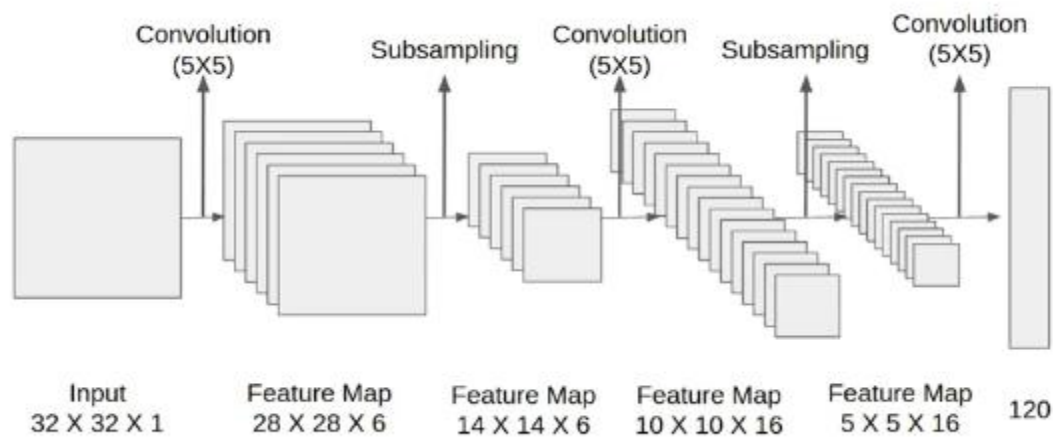


Fig. 4.4 : LeNet Architecture Representation

VGGNet:

VGGNet is the CNN architecture that was developed by Karen Simonyan, Andrew Zisserman et al. at Oxford University. VGGNet is a 16-layer CNN with up to 95 million parameters and trained on over one billion images (1000 classes). The VGG-16 is one of the most popular pre-trained models for image classification. It can take large input images of 224×224 -pixel size for which it has 4096 convolutional features. CNNs with such large filters are expensive to train and require a lot of data, which is the main reason why CNN architectures like GoogLeNet (AlexNet architecture) work better than VGGNet for most image classification tasks where input images have a size between 100×100 -pixel and 350×350 pixels. The VGG CNN model is computationally efficient and serves as a strong baseline for many applications in computer vision due to its applicability for numerous tasks including object detection.

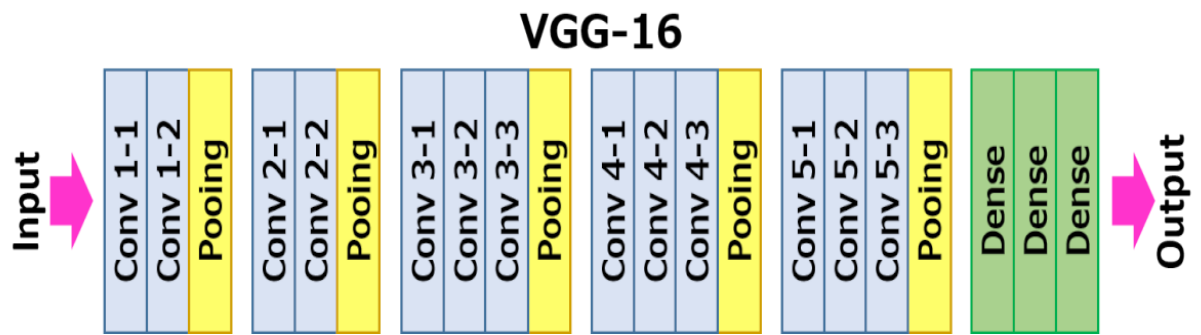


Fig. 4.5 : VGG-16 Architecture Representation

Inception:

Inception network was once considered a state-of-the-art deep learning architecture (or model) for solving image recognition and detection problems.

Inception Layer is a combination of all those layers (namely, 1×1 Convolutional layer, 3×3 Convolutional layer, 5×5 Convolutional layer) with their output filter banks concatenated into a single output vector forming the input of the next stage.

Along with the above-mentioned layers, there are two major add-ons in the original inception layer 1×1 Convolutional layer before applying another layer, which is mainly used for dimensionality reduction. A parallel Max Pooling layer, which provides another option to the inception layer, the Inception Module just performs convolutions with different filter sizes on the input, performs Max Pooling, and concatenates the result for the next Inception module. The introduction of the $1 * 1$ convolution operation reduces the parameters drastically. The Inceptionv2 model was a major improvement on the Inceptionv1 model which increased the accuracy and further made the model less complex. The researchers has introduced the Inceptionv3 model with a few more improvements on v2.

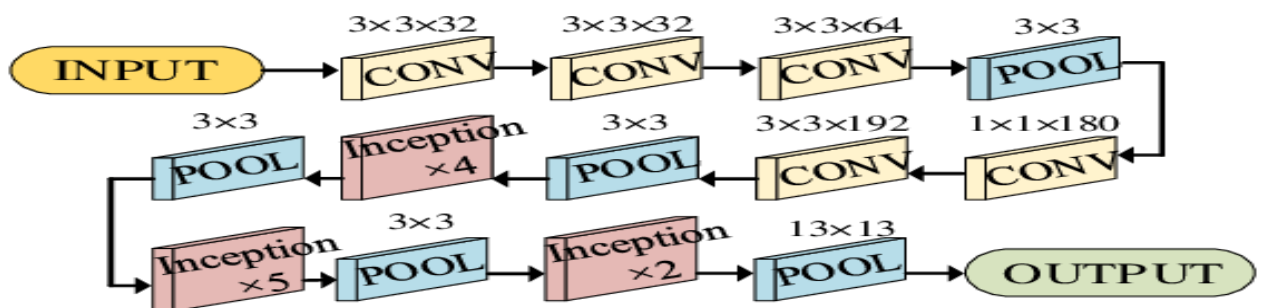


Fig. 4.6 : Inception Architecture Representation

ResNet:

ResNet is the CNN architecture that was developed by Kaiming He et al. The network has 152 layers and over one million parameters, which is considered deep even for CNNs because it would have taken more than 40 days on 32 GPUs to train the network on the ILSVRC 2015 dataset. CNNs are mostly used for image classification tasks with 1000 classes, but ResNet proves that CNNs can also be used successfully to solve natural language processing problems like sentence completion or machine comprehension, where it was used by the Microsoft Research Asia team in 2016 and 2017 respectively. ResNet50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. Real-life applications/examples of ResNet CNN architecture include Microsoft's machine comprehension system, which has used CNNs to generate the answers for more than 100k questions in over 20 categories. The CNN architecture ResNet is computationally efficient and can be scaled up or down to match the computational power of GPUs.

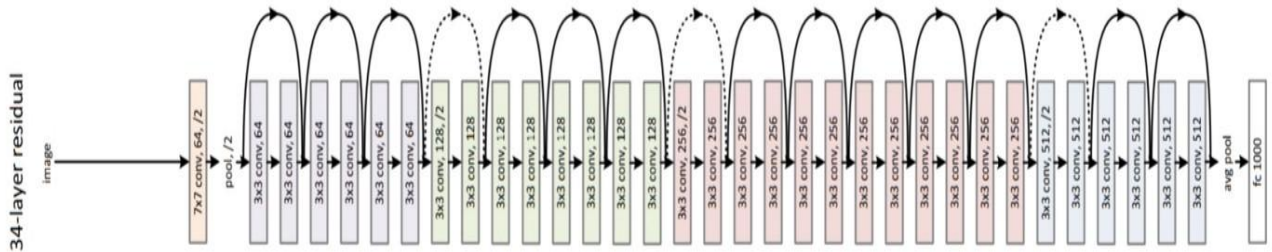


Fig. 4.7 : ResNet Architecture Representation

4.5.5 Optimization Algorithm

As the optimal algorithm, the proposed model is trained with the Adaptive Moment Estimation (Adam) algorithm. This method is very efficient when working big data set or parameters. This algorithm is a combination of the root mean square propagation algorithm and gradient descent algorithm. This algorithm takes the best characteristics of the two methods and builds on them to produce a more optimized gradient descent algorithm. The rule is given by equation 1, 2, and 3.

$$w_{t+1} = w_t - m_t \left(\frac{\alpha}{\sqrt{v_t^2 + \epsilon}} \right) \quad (1)$$

$$m_t = \frac{\beta_1^t m_{t-1} + (1 - \beta_1^t) g_t}{1 - \beta_1^t} \quad (2)$$

$$v_t = \frac{\beta_2^t v_{t-1} + (1 - \beta_2^t) g_t^2}{1 - \beta_2^t} \quad (3)$$

Where, w_t =weights at time, w_{t+1} =weights at time t+1, α =learning rate, ϵ =small positive constant, m_t =bias corrected weight parameters, m_t =bias corrected, v_t =sum of square of past gradients, β_1^t & β_2^t =decay rates of average of gradients. Adam optimizer gives high performance and outperforms other optimization algorithms.

Deep Learning Pipeline and Implementation:

The two main stages of implementing an algorithm are the training and testing phases. First, the training phase creates the CNN model using a training dataset based on the chosen models. The three most commonly known CNN models, namely, LeNet, Residual Network (ResNet), and Inception, are employed. The performance validation of the training stage guarantees the general performance of the classifier model and is used to avoid the overfitting issue. Then, verification is performed on the trained model in the testing stage using the testing dataset as input to the trained classifier. This testing dataset is the other partitioned data from the original dataset; therefore, it has identical characteristics to the training dataset. The original dataset is partitioned into 70% and 30% for training and testing, respectively.

Performance metrics are used to evaluate both stages. By comparing the performance of both training and testing, any overfitting issue can be determined. It occurs when the training performance is relatively higher than the testing performance.

4.6 MODEL ARCHITECTURE

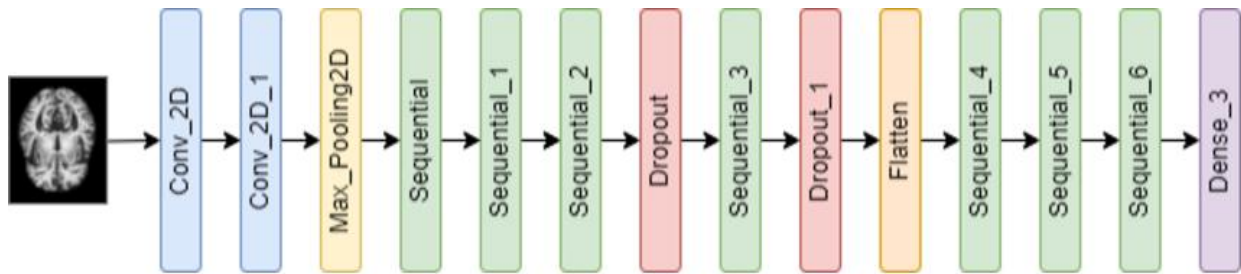


Fig. 4.8 : Model Architecture

4.7 WORKFLOW DIAGRAM

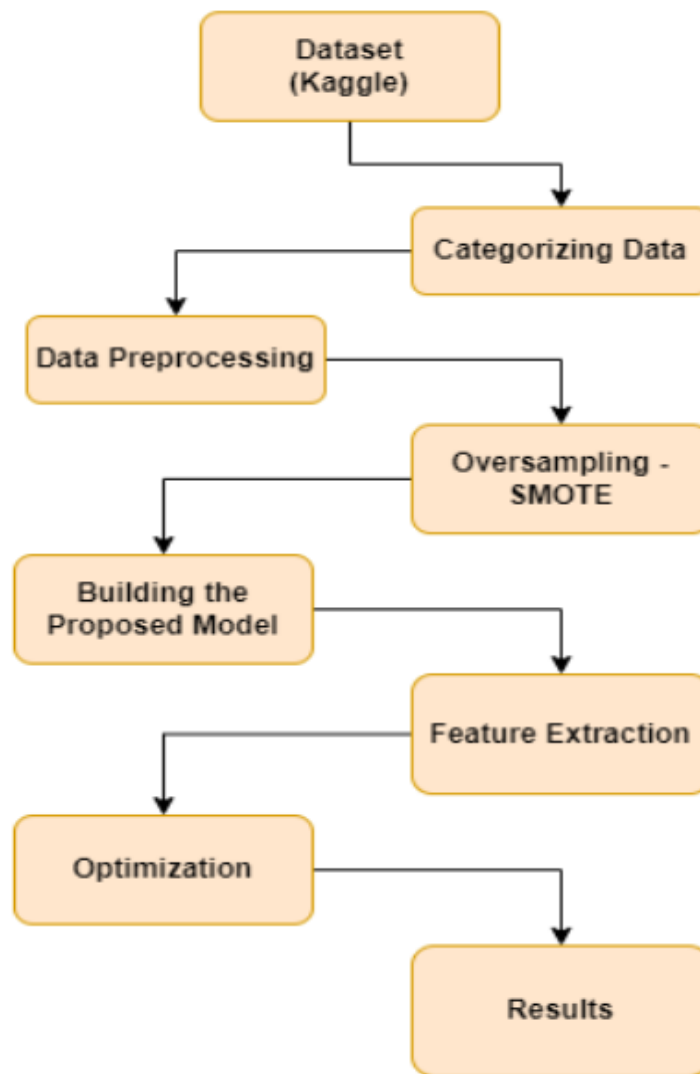


Fig. 4.9 : Workflow Diagram

CHAPTER-5

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

5.1 RESULTS:

The model training is performed using parameter 100 epoch, batch size of 6500. The model attains a validation accuracy of 94% and training accuracy of 98%. In order to determine whether the model is correctly distinguishing positive and negative categories, at every epoch, AUC (Area Under Curve) is determined. The confusion matrix is used to calculate the each and every class metrics and shown in table.

Table 5.1 : Evaluation Metrics of Individual Class

Disease	PR	REC	F1-Score	Support
ND	1	0.92	0.96	639
VMD	1	1	1	635
MD	0.9	0.94	0.92	662
MOD	0.9	0.92	0.91	624

5.2 PERFORMANCE ANALYSIS:

Comparing with previous works, they got an accuracy of 91%, but in our proposed model we have achieved an overall accuracy of 94%.

Table 5.2 : Performance Analysis

Metric	Our Model	ResNet50	InceptionV3
Test Accuracy	94.59%	75.70%	88.83%
Training Accuracy	0.98	0.77	0.94
Precision	0.94	0.5833333134651184	0.89
Recall	0.94	0.09851446747779846	0.89
F1-Score	0.94	0.16603372991085052	0.89

CHAPTER-6

SUMMARY AND CONCLUSIONS

6.1 Summary:

This project objective is to detect the stage of Alzheimer's Disease using MR images. So this model will greatly reduce the time taken to diagnose the disease, which is the important factor.

6.2 Conclusion:

In this paper, a custom CNN model is proposed to predict the class of Alzheimer disease among the given classified images. The proposed model has been tested with testing data consisting of four classes and accomplish a 93.59% accuracy. The dataset has a major disadvantage - a class disparity, the SMOTE method is employed to resolve this issue. In this way, it is well suited to identify brain areas often associated with AD and can facilitate physicians' decision making by helping them determine each patient's AD severity level according to the level of dementia. Here we observe and compare the accuracy of three models namely VGG16, ResNet50, Inception_V3 among these, The ResNet 50 model has the best overall accuracy and F1 score.

REFERENCES:

1. Fuse Hiroki; Oishi Kota; Maikusa Norihide; Fukami Tadanori; (2018). Detection of Alzheimer's Disease with Shape Analysis of MRI Images. Japanese Alzheimer's Disease Neuroimaging Initiative.
2. H. M. Tarek Ullah; Zishan Ahmed Onik; Riashat Islam; Dr. Dip Nandi; (2018). Alzheimer's Disease And Dementia Detection From 3D Brain MRI Data Using Deep Convolutional Neural Networks. 2018 3rd International Conference for Convergence in Technology.
3. Neelaveni, J.; Devasana, M.S.Geetha (2020). 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS) - Alzheimer Disease Prediction using Machine Learning Algorithms.
4. Marcus, Daniel S.; Wang, Tracy H.; Parker, Jamie; Csernansky, John G.; Morris, John C.; Buckner, Randy L. (2007). Open Access Series of Imaging Studies (OASIS): Crosssectional MRI Data in Young, Middle Aged, Nondemented, and Demented Older Adults. Journal of Cognitive Neuroscience.
5. Malone, Ian B.; Cash, David; Ridgway, Gerard R.; MacManus, David G.; Ourselin, Sebastien; Fox, Nick C.; Schott, Jonathan M. (2013). MIRIAD—Public release of a multiple time point Alzheimer's MR imaging dataset.
6. Fan, Zhao; Xu, Fanyu; Qi, Xuedan; Li, Cai; Yao, Lili (2019). Classification of Alzheimer's disease based on brain MRI and machine learning. Neural Computing and Applications.
7. Acharya, U. Rajendra; Fernandes, Steven Lawrence; WeiKoh, Joel En; Ciaccio, Edward J.; Fabell, Mohd Kamil Mohd; Tanik, U. John; Rajinikanth, V.; Yeong, Chai Hong (2019). Automated Detection of Alzheimer's Disease Using Brain MRI Images—A Study with Various Feature Extraction Techniques. Journal of Medical Systems.
8. Eke, Chima Stanley; Jammeh, Emmanuel; Li, Xinzhong; Carroll, Camille; Pearson, Stephen; Ifeakor, Emmanuel (2020). Early Detection of Alzheimer's Disease with Blood Plasma Proteins using Support Vector Machines. IEEE Journal of Biomedical and Health Informatics.
9. R. Jain, N. Jain, A. Aggarwal and D. J. Hemanth (2019), Convolutional neural network based Alzheimer's disease classification from magnetic resonance brain images.

10. Ge, Chenjie; Qu, Qixun; Gu, Irene Yu-Hua; Store Jakola, Asgeir (2019). 2019 IEEE International Conference on Image Processing (ICIP) - Multiscale Deep Convolutional Networks for Characterization and Detection of Alzheimer's Disease Using MR images.
11. Stoyanov, Danail; Taylor, Zeike; Kia, Seyed Mostafa; Oguz, Ipek; Reyes, Mauricio; Martel, Anne; Maier-Hein, Lena; Marquand, Andre F.; Duchesnay, Edouard; Löfstedt, Tommy; Landman, Bennett; Cardoso, M. Jorge; Silva, Carlos A.; Pereira, Sergio; Meier, Raphael (2018). Understanding and Interpreting Machine Learning in Visualizing Convolutional Networks for MRI-Based Diagnosis of Alzheimer's Disease. First International Workshops.
12. Biju, K.S.; Alfa, S.S.; Lal, Kavya; Antony, Alvia; Akhil, M Kurup (2017). Alzheimer's Detection Based on Segmentation of MRI Image. Procedia Computer Science.
13. Khan, A., & Zubair, S. (2020). A Machine Learning-based robust approach to identify Dementia progression employing Dimensionality Reduction in Cross-Sectional MRI data. 2020 First International Conference of Smart Systems and Emerging Technologies.
14. Tanveer, M., Richhariya, B., Khan, R. U., Rashid, A. H., Khanna, P., Prasad, M., & Lin, C. T. (2020). Machine Learning Techniques for the Diagnosis of Alzheimer's Disease. ACM Transactions on Multimedia Computing, Communications, and Applications.
15. Subramanyam Rallabandi, V. P., & Seetharaman, K. (2021). Machine Learning-Based Classification of Dementia Types: MRI Study. 2021 International Conference on Artificial Intelligence and Smart Systems.
16. Mehmood, A., Yang, S., Feng, Z., Wang, M., Ahmad, A. S., Khan, Rizwan Khan, Muazzam Maqsood, Yaqub, M. (2021). A Transfer Learning Approach for Early Diagnosis of Alzheimer's Disease on MRI Images.
17. A. De and A. S. Chowdhury (2021), DTI based Alzheimer's disease classification with rank modulated fusion of CNNs and random forest, Expert Syst. Appl.
18. J. Shi, X. Zheng, Y. Li, Q. Zhang and S. Ying (2018), Multimodal neuroimaging feature learning with multimodal stacked deep polynomial networks for diagnosis of Alzheimer's disease.
19. Islam, Jyoti; Zhang, Yanqing (2018). Brain MRI analysis for Alzheimer's disease

diagnosis using an ensemble system of deep convolutional neural networks. *Brain Informatics*.

20. Mandal, Pravat K.; Mahajan, Rashima; Dinov, Ivo D.; Mandal, Pravat K. (2012). Structural Brain Atlases: Design, Rationale, and Applications in Normal and Pathological Cohorts. *Journal of Alzheimer's Disease*.
21. Jing Zhang; Chunshui Yu; Guilian Jiang; Weifang Liu; Longzheng Tong (2012). 3D texture analysis on MRI images of Alzheimer's disease.
22. Krishnakumar, V.; Parthiban, Latha (2019). A Novel Texture Extraction Technique with T1 Weighted MRI for the Classification of Alzheimer's Disease. *Journal of Neuroscience Methods*.
23. Xiao, Zhe; Ding, Yi; Lan, Tian; Zhang, Cong; Luo, Chuanji; Qin, Zhiguang (2017). Brain MR Image Classification for Alzheimer's Disease Diagnosis Based on Multifeature Fusion. *Computational and Mathematical Methods in Medicine*.
24. Ben Ahmed, Olfa; Benois-Pineau, Jenny; Allard, Michèle; Ben Amar, Chokri; Catheline, Gwénaëlle (2015). Classification of Alzheimer's disease subjects from MRI using hippocampal visual features. *Multimedia Tools and Applications*.
25. Beheshti, I.; Demirel, H. (2015). Feature-ranking-based Alzheimer's disease classification from structural MRI. *Magnetic Resonance Imaging*.
26. Zhou, Qi; Goryawala, Mohammed; Cabrerizo, Mercedes; Wang, Jin; Barker, Warren; Loewenstein, David A.; Duara, Ranjan; Adjouadi, Malek (2014). An Optimal Decisional Space for the Classification of Alzheimer's Disease and Mild Cognitive Impairment. *IEEE Transactions on Biomedical Engineering*.
27. Prashanthi Vemuri; Jeffrey L. Gunter; Matthew L. Senjem; Jennifer L. Whitwell; Kejal Kantarci; David S. Knopman; Bradley F. Boeve; Ronald C. Petersen; Clifford R. Jack Jr. (2008). Alzheimer's disease diagnosis in individual subjects using structural MR images: Validation studies. *Neuro Image*.
28. Liu, Manhua; Zhang, Daoqiang; Shen, Dinggang (2014). Hierarchical fusion of features and classifier decisions for Alzheimer's disease diagnosis. *Human Brain Mapping*.
29. Khedher, L.; Ramírez, J.; Górriz, J.M.; Brahim, A.; Segovia, F. (2015). Early diagnosis of Alzheimer's disease based on partial least squares, principal component analysis and support vector machine using segmented MRI images. *Neurocomputing*.
30. Ben Ahmed, Olfa; Mizotin, Maxim; Benois-Pineau, Jenny; Allard, Michèle; Catheline,

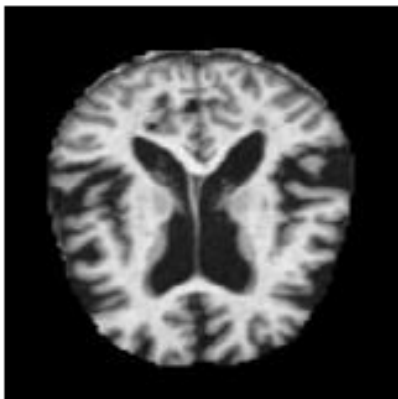
Gwénaëlle; Ben Amar, Chokri (2015). Alzheimer's disease diagnosis on structural MR images using circular harmonic functions descriptors on hippocampus and posterior cingulate cortex. Computerized Medical Imaging and Graphics.

31. S. Dubey, Alzheimer's Dataset, (2019), Available at:
<https://www.kaggle.com/tourist55/alzheimers-dataset-4-class-of-images>

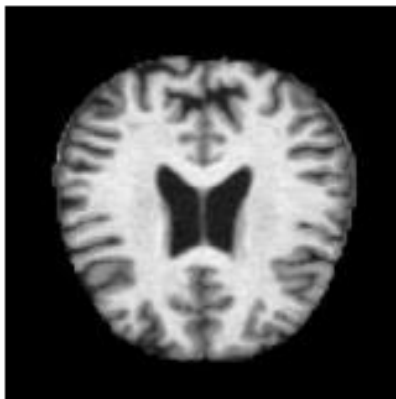
APPENDIX:

Screen Shots:

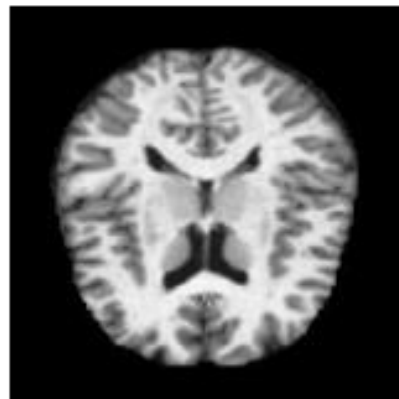
Class:MildDemented



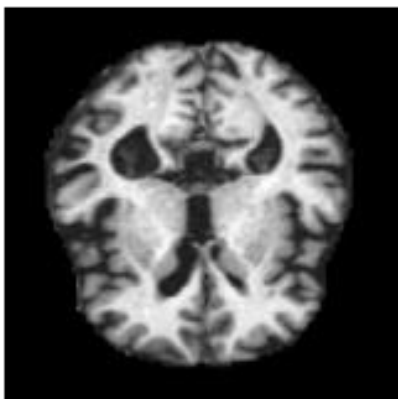
Class:ModerateDemented



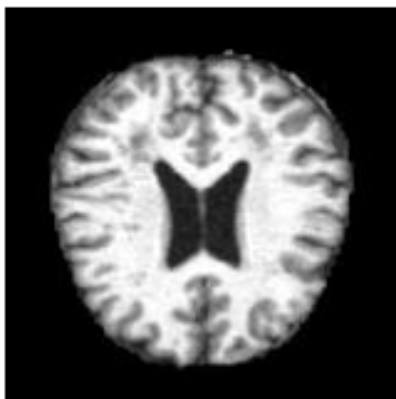
Class:MildDemented



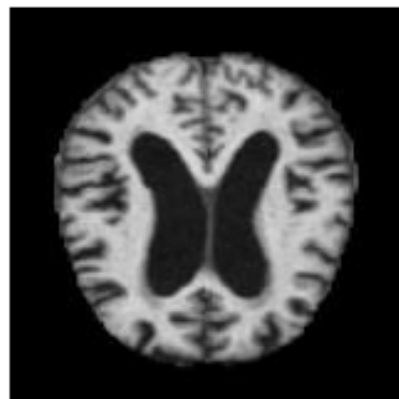
Class:NonDemented



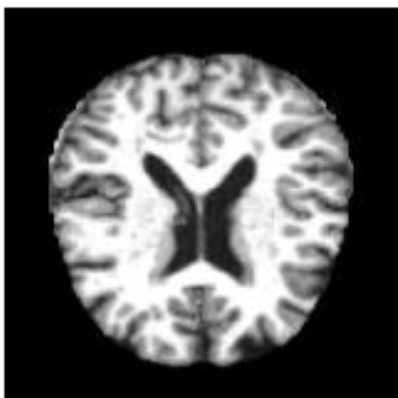
Class:MildDemented



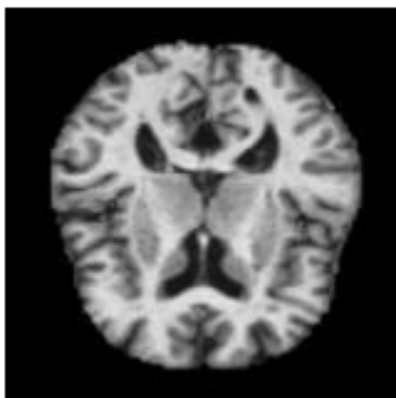
Class:ModerateDemented



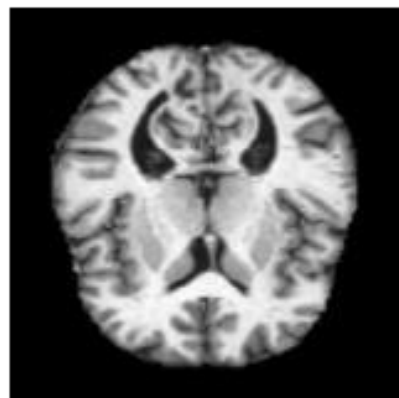
Class:MildDemented



Class:MildDemented



Class:MildDemented

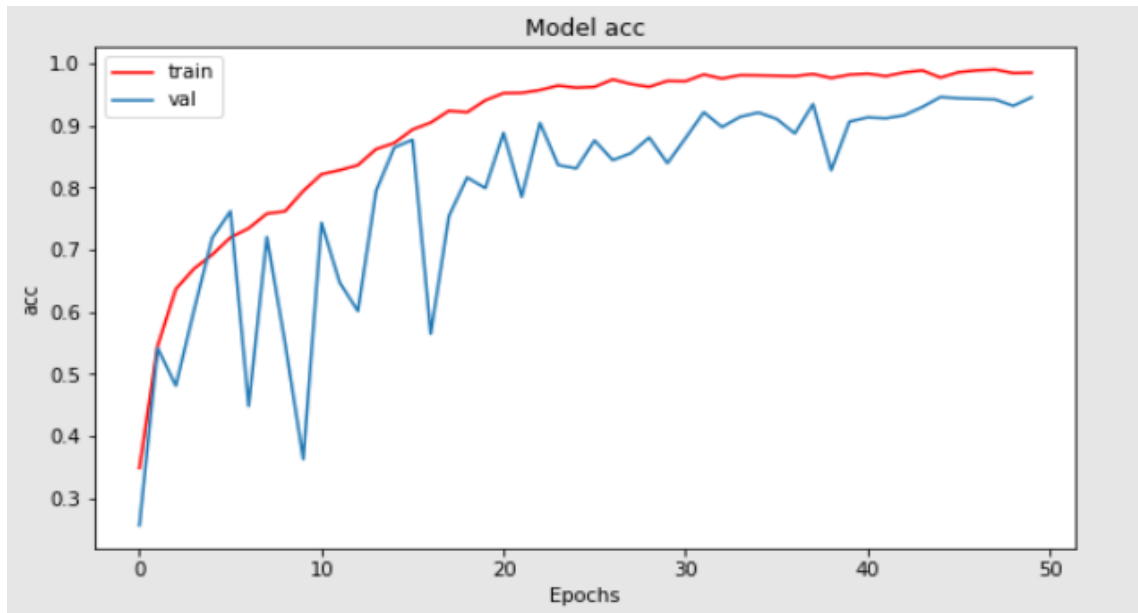


a) Dataset Overview

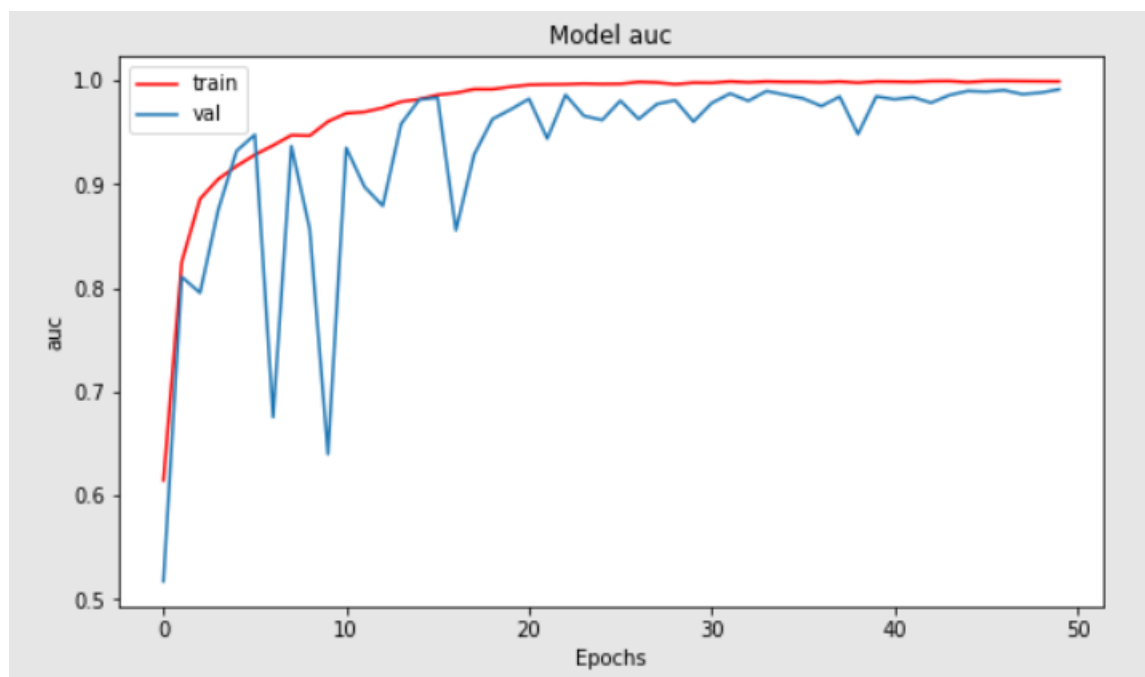
Model: "cnn_model"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 176, 176, 16)	448
conv2d_1 (Conv2D)	(None, 176, 176, 16)	2320
max_pooling2d (MaxPooling2D)	(None, 88, 88, 16)	0
sequential (Sequential)	(None, 44, 44, 32)	14016
sequential_1 (Sequential)	(None, 22, 22, 64)	55680
sequential_2 (Sequential)	(None, 11, 11, 128)	221952
dropout (Dropout)	(None, 11, 11, 128)	0
sequential_3 (Sequential)	(None, 5, 5, 256)	886272
dropout_1 (Dropout)	(None, 5, 5, 256)	0
flatten (Flatten)	(None, 6400)	0
sequential_4 (Sequential)	(None, 512)	3279360
sequential_5 (Sequential)	(None, 128)	66176
sequential_6 (Sequential)	(None, 64)	8512
dense_3 (Dense)	(None, 4)	260
=====		
Total params: 4,534,996		
Trainable params: 4,532,628		
Non-trainable params: 2,368		

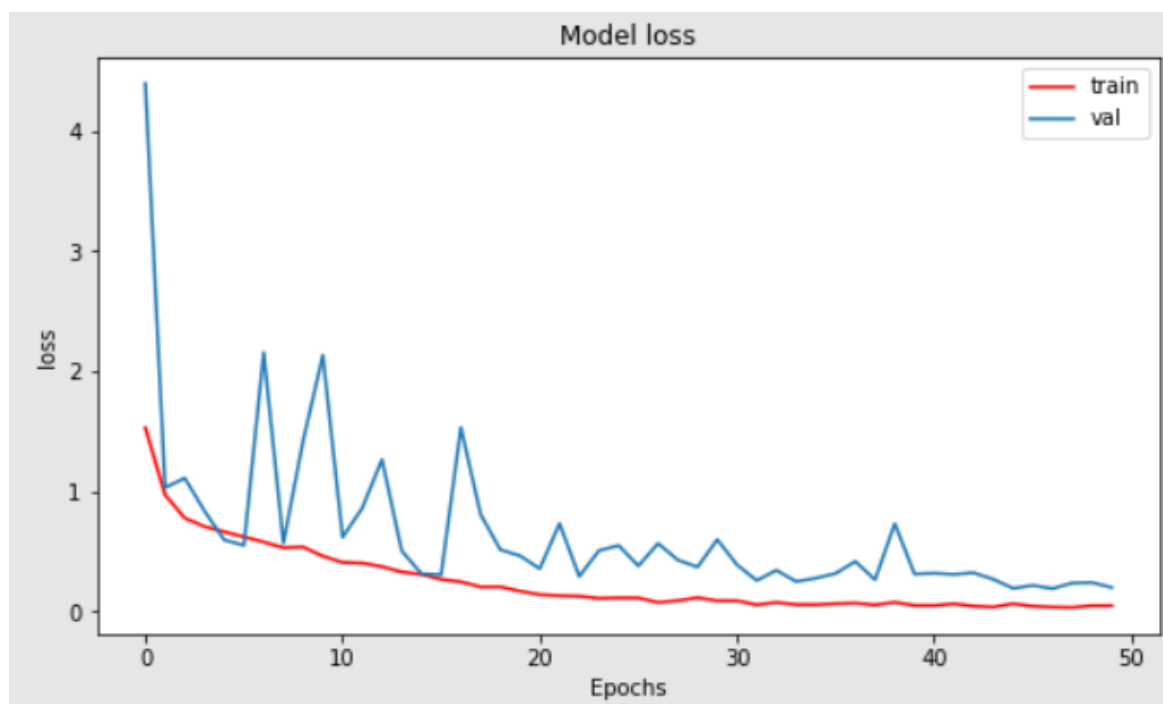
b) CNN Model Layers



c) Training Model Accuracy



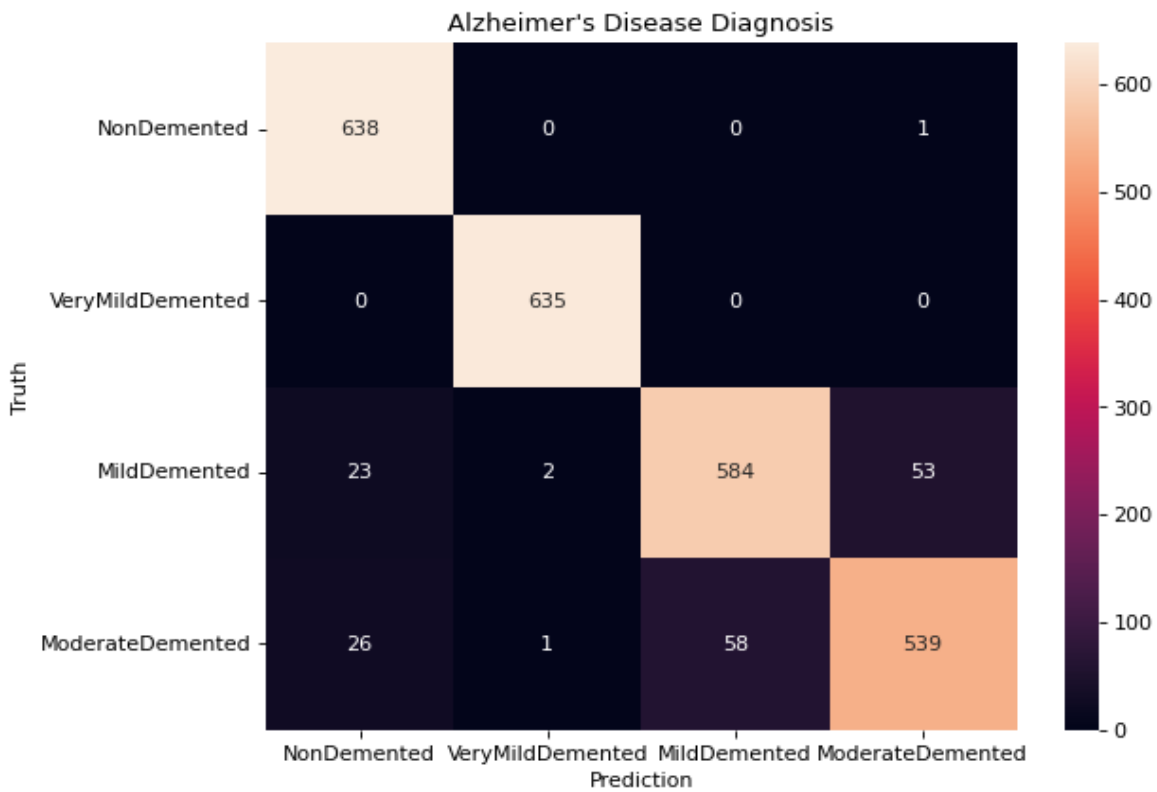
d) Training Model AUC



e) Training Model Loss

	precision	recall	f1-score	support
NonDemented	0.93	1.00	0.96	639
VeryMildDemented	1.00	1.00	1.00	635
MildDemented	0.91	0.88	0.90	662
ModerateDemented	0.91	0.86	0.89	624
micro avg	0.94	0.94	0.94	2560
macro avg	0.94	0.94	0.94	2560
weighted avg	0.94	0.94	0.94	2560
samples avg	0.94	0.94	0.94	2560

f) Detailed Final Model Analysis of Each Category



g) Visualization of the Prediction

```
test_scores = model.evaluate(test_data, test_labels)

print("Test Accuracy: %.2f%%"%(test_scores[1] * 100))
```

80/80 [=====] - 35s 438ms/step - loss: 0.1987 - acc: 0.9359 - auc: 0.9926 - f1_score: 0.9354
 Test Accuracy: 93.59%

h) Final Model Test Accuracy

B. SOURCE CODE:

```
import os
import numpy as np
import pandas as pd
import seaborn as sns
import tensorflow as tf
import matplotlib.pyplot as plt
from distutils.dir_util import copy_tree, remove_tree
from PIL import Image
from random import randint
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.metrics import matthews_corrcoef as MCC
from sklearn.metrics import balanced_accuracy_score as BAS
from sklearn.metrics import classification_report, confusion_matrix
import tensorflow_addons as tfa
from tensorflow.keras import Sequential, Input
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.layers import Conv2D, Flatten
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.preprocessing.image import ImageDataGenerator as IDG
from tensorflow.keras.layers import SeparableConv2D, BatchNormalization,
MaxPool2D

default_dir = "C:/Users/hrith/Downloads/project/AlzheimerDataset/"
root_dir = "C:/Users/hrith/Downloads/project/"
test_dir = default_dir + "test/"
train_dir = default_dir + "train/"
work_dir = root_dir + "dataset/"
if os.path.exists(work_dir):
    remove_tree(work_dir)
os.mkdir(work_dir)
copy_tree(train_dir, work_dir)
copy_tree(test_dir, work_dir)
print("Working Directory Contents:", os.listdir(work_dir))

WORK_DIR = 'C:/Users/hrith/Downloads/project/dataset/'
CLASSES = [ 'NonDemented',
            'VeryMildDemented',
            'MildDemented',
            'ModerateDemented']

IMG_SIZE = 176
IMAGE_SIZE = [176, 176]
DIM = (IMG_SIZE, IMG_SIZE)
```

```

ZOOM = [.99, 1.01]
BRIGHT_RANGE = [0.8, 1.2]
HORZ_FLIP = True
FILL_MODE = "constant"
DATA_FORMAT = "channels_last"
work_dr = IDG(rescale = 1./255, brightness_range=BRIGHT_RANGE, zoom_range=ZOOM,
data_format=DATA_FORMAT, fill_mode=FILL_MODE, horizontal_flip=HORZ_FLIP)
train_data_gen = work_dr.flow_from_directory(directory=WORK_DIR,
target_size=DIM, batch_size=6500, shuffle=False)

def show_images(generator,y_pred=None):
    labels =dict(zip([0,1,2,3], CLASSES))
    x,y = generator.next()
    plt.figure(figsize=(10, 10))
    if y_pred is None:
        for i in range(9):
            ax = plt.subplot(3, 3, i + 1)
            idx = randint(0, 6400)
            plt.imshow(x[idx])
            plt.axis("off")
            plt.title("Class:{}".format(labels[np.argmax(y[idx])]))
    else:
        for i in range(9):
            ax = plt.subplot(3, 3, i + 1)
            plt.imshow(x[i])
            plt.axis("off")
            plt.title("Actual:{}
\nPredicted:{}".format(labels[np.argmax(y[i])],labels[y_pred[i])])
show_images(train_data_gen)

train_data, train_labels = train_data_gen.next()
print(train_data.shape, train_labels.shape)

sm = SMOTE(random_state=42)
train_data, train_labels = sm.fit_resample(train_data.reshape(-1, IMG_SIZE *
IMG_SIZE * 3), train_labels)
train_data = train_data.reshape(-1, IMG_SIZE, IMG_SIZE, 3)
print(train_data.shape, train_labels.shape)

train_data, test_data, train_labels, test_labels = train_test_split(train_data,
train_labels, test_size = 0.2, random_state=42)
train_data, val_data, train_labels, val_labels = train_test_split(train_data,
train_labels, test_size = 0.2, random_state=42)

def conv_block(filters, act='relu'):
    block = Sequential()

```

```

        block.add(Conv2D(filters, 3, activation=act, padding='same'))
        block.add(Conv2D(filters, 3, activation=act, padding='same'))
        block.add(BatchNormalization())
        block.add(MaxPool2D())
        return block

def dense_block(units, dropout_rate, act='relu'):
    block = Sequential()
    block.add(Dense(units, activation=act))
    block.add(BatchNormalization())
    block.add(Dropout(dropout_rate))
    return block

def construct_model(act='relu'):
    model = Sequential([
        Input(shape=(*IMAGE_SIZE, 3)),
        Conv2D(16, 3, activation=act, padding='same'),
        Conv2D(16, 3, activation=act, padding='same'),
        MaxPool2D(),
        conv_block(32),
        conv_block(64),
        conv_block(128),
        Dropout(0.2),
        conv_block(256),
        Dropout(0.2),
        Flatten(),
        dense_block(512, 0.7),
        dense_block(128, 0.5),
        dense_block(64, 0.3),
        Dense(4, activation='softmax')
    ], name = "cnn_model")
    return model

model = construct_model()
METRICS = [tf.keras.metrics.CategoricalAccuracy(name='acc'),
            tf.keras.metrics.AUC(name='auc'),
            tfa.metrics.F1Score(num_classes=4)]
CALLBACKS = [my_callback]
model.compile(optimizer='adam',
              loss=tf.losses.CategoricalCrossentropy(),
              metrics=METRICS)
model.summary()

EPOCHS = 50

```

```

history = model.fit(train_data, train_labels, validation_data=(val_data,
val_labels), callbacks=CALLBACKS, epochs=EPOCHS)

fig, ax = plt.subplots(1, 3, figsize = (30, 5))
ax = ax.ravel()
for i, metric in enumerate(["acc", "auc", "loss"]):
    ax[i].plot(history.history[metric],color="r")
    ax[i].plot(history.history["val_" + metric])
    ax[i].set_title("Model {}".format(metric))
    ax[i].set_xlabel("Epochs")
    ax[i].set_ylabel(metric)
    ax[i].legend(["train", "val"])

test_scores = model.evaluate(test_data, test_labels)
print("Test Accuracy: %.2f%%"%(test_scores[1] * 100))

pred_labels = model.predict(test_data)

def roundoff(arr):
    arr[np.argwhere(arr != arr.max())] = 0
    arr[np.argwhere(arr == arr.max())] = 1
    return arr

for labels in pred_labels:
    labels = roundoff(labels)
print(classification_report(test_labels, pred_labels, target_names=CLASSES))

pred_ls = np.argmax(pred_labels, axis=1)
test_ls = np.argmax(test_labels, axis=1)
conf_arr = confusion_matrix(test_ls, pred_ls)
plt.figure(figsize=(8, 6), dpi=80, facecolor='w', edgecolor='k')
ax = sns.heatmap(conf_arr, annot=True, fmt='d', xticklabels=CLASSES,
yticklabels=CLASSES)
plt.title('Alzheimer\'s Disease Diagnosis')
plt.xlabel('Prediction')
plt.ylabel('Truth')
plt.show(ax)

print("Balanced Accuracy Score: {} %".format(round(BAS(test_ls, pred_ls) * 100,
2)))
print("MCC: {} %".format(round(MCC(test_ls, pred_ls) * 100, 2)))

model_dir = work_dir + "alzheimer_cnn_model"
model.save(model_dir, save_format='h5')
os.listdir(work_dir)

```

C. Plagiarism Report:

Application of Deep Learning to detect different stages of Alzheimer's Disease

Inbasagar D^{1*}, Hrithik R², Maheswari M³, and Selvi M⁴

¹Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.

²Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.

³Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.

⁴Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.

¹inbal72001@gmail.com

²hrithikmikel0@gmail.com

³mahisara10@gmail.com

⁴selvi.cse@sathyabama.com

Abstract. There are many different causes of dementia, but Alzheimer's Disease is the most usual form. As the condition progresses, it limits one's ability to perform any task without aid, and the diagnosis timeline and aging population are expected to cause its prevalence to increase. The conventional ways of detecting Alzheimer's is tiring for both patients, doctors where it involves retrieving the past medical records and having Magnetic Resonance Imaging scans and even neurophysical testing which can be inconvenient for patients. An early diagnosis of brain diseases makes a big difference when it comes to attempting to cure them. Our work has used deep learning (neural networks) to detect Alzheimer's disease earlier than usual by combining it with deep learning. As the obtained dataset from Kaggle is heavily imbalanced, we evenly distributed the data between the categories using SMOTE. Then the model is trained and tested with the categorized MRI data i.e. very mild, mild, moderate and severe AD and finally extract features to examine the results. The results we achieved are compared with the previous attempts on detection of Alzheimer's and came out to be significantly greater in terms of precision and accuracy.

Keywords: MRI, Alzheimer's, Deep Learning, Dementia, Neural Networks, Detection.

1 Introduction

A person with Alzheimer's Disease (AD) suffers from a developing neurobiological disorder that affects brain cells to die & atrophy. AD is the prevalent cause of dementia, which causes memory loss and decreased reasoning abilities. Around 6 million people in the U.S of age 65 and above are caused with Alzheimer's disease. Of those 80 percent are above 75 years old. In India it is calculated around 5.3 million people live with dementia of which Alzheimer is the common cause.

Dementia with Alzheimer's is classified into four categories

- a. Very Mild Dementia: Individuals suffers from memory loss as they age.
- b. Mild Dementia: Symptoms which includes lack of memory, Behavioral changes, inability to perform routine tasks
- c. Moderate Dementia: The day to day life becomes complex for the individuals with moderate dementia, where the patients require extra care and support.
- d. Severe Dementia: The patients in this stage may not able to communicate properly, and they require medical care. One may lose physical control.

Alzheimer's is not a curable disease, but an early diagnosis can help prevent the patient from sufferin¹⁵ on the later stages. In order to diagnose AD manual detection systems for example: Positron Emission Tomography (PET) were used to track the progression of the various stages of AD. MRI scans and genotype sequencing results were taken for diagnosis.

Among the most popular fields of research in recent years has been brain-computer interface (BCI), thanks to its applications in areas such as brain fingerprinting, detecting neurological illnesses, tiredness, adaptive e-learning and more. By extracting the most significant characteristics, BCI creates an effective link to interact between the brain and the device. A complex brain structure varying with age and pathology makes it very difficult to detect neurodegenerative diseases in their early stages. Computer-assisted techniques are more successful in detecting these disorders than traditional approaches.

¹⁶ A timely diagnosis and identifying of Alzheimer's disease is essential to reducing medical expenses, improving treatment, and preventing brain cell degeneration. Some of the methods included in an early examination include Positron Emission Tomography (PET), digital imaging methods and genotyping-by-sequencing. It is difficult to take decisions by analyzing different methods. Furthermore, the patients will have to undergo radioactive effects during PET medical procedure. According to our findings, MRIs can provide valuable information on the brain, because they provide flexible imaging, superior tissue contrast, and do not expose the brain to ionizing radiation.

It is crucial to develop a model that can take MR images as input and detect whether patients are normal or not. By utilizing a dataset, machine learning can extract knowledge. Computer science, artificial intelligence, and statistics combine to make up this field.. The ML is done through training a computer to produce the output based on its past experience to solve a given problem. Machine learning can be applied in a variety of fields in order to solve problems quicker than humans, and therefore be more efficient, and reduce time spent on repetitive tasks. Nowadays, because of the reduction in the cost of computing power and memory. This allows processing and analyzing huge amounts of data to generate insights.

Additionally, Deep Learning is a subset of ML and a advanced mode of analyzing and learning information from raw data that computers are able to replicate, much like how humans are able to do, with a computer. Deep learning is becoming increasingly popular for diagnosing diseases. Several Machine learning approaches have been proposed recently to aid in this diagnosis, providing doctors with more information to make informed decisions.

We use Deep Neural Networks (DNNs) for feature extraction using deep learning techniques in the proposed model. A solution to underfitting is to use sampling techniques especially oversampling to resolve class imbalance. DL performs classification on given MR images using the cortical surface of the brain as input.

Using a dementia-specific (Alzheimer's) dataset, the models are evaluated by Non-Demented, Moderate Dementia, Mild Dementia and Very Mild Dementia obtained from Kaggle. Utilizing the CNN technique, we extract discriminating features for AD classification by improving the accuracy. By using this model, we can accurately classify the stages of AD.

2 Related Work

Researchers have made several attempts to classify Alzheimer's patients using the brain structure information obtained from MR images. Here, the shape information of the brain, particularly that of the lateral ventricles, was analyzed without considering the septum lucidum.[1].

This study have shown that deep CNNs are capable of detecting Alzheimer's disease and dementia based on 3D MRI imaging and can be used to detect the disease[2].

Neelaveni J et al. [5] used machine-learning algorithms, they used psychological factors to predict Alzheimer's, such as age, MMSE scores, and educational level. If the disease can be predicted earlier, symptoms or progression can be slowed down.

They demonstrate how whole-brain volume can automatically calculated and estimated spinal fluid volume used for determining differences associated with ideal aging and Alzheimer's disease[6].

It is well-established, widely used, and non-invasive to use structural MR imaging in Alzheimer's disease studies of downstream effect of the neurodegeneration, or atrophy. In peer-reviewed journals, the imaging and cohort results have been published in detail, and the data have been made freely available[7].

Zhao Fan et al. [9], help aid auxiliary diagnosis of Alzheimer's disease, with the help of SVM model to analyze and classify structural brain MR imaging data, in order to achieve much greater classification predictions, the extracted MRI detail is combined with the SVM model.

Acharya et al, used several feature extraction techniques along with MR imaging for classification, the system uses several algorithms for Computer-Assisted-Brain-Diagnosis (CABD) to assess if the scan image shows indications of Alzheimer's disease. According to their results, Shearlet Transform (ST) feature extraction is superior to alternative methods for detecting Alzheimer's disease[10].

Chima Stanley Eke et al. [11] the main basis for their method is machine learning (ML) techniques (specifically support vector machines), which are capable of learning patterns from complex data in order to create multi-variable models. On the basis of the identified panels, disease detection models were determined to have a sensitivity greater than 80%, a specificity greater than 70%, and AUC greater than 0.80 at the prodromal stage of the disease.

R. Jain et al. [12] proposed convolutional neural network model (CNNs) were used to locate gray matter in brain voxels and to segment segmented gray matter for clinical appraisal. The proposed methodology achieved an accuracy rate of 90.47% for segments of gray matter for clinical evaluation.

Chenjie Ge et al. [13] the paper showcases, an innovative idea to detect Alzheimer's disease (AD) using 3D multiscale neural network architecture. It successfully achieved an accuracy of 87.24% for detection.

Danail Stoyanov et al. [14] used brain structural MRI scans, they developed a 3D CNN to identify Alzheimer's disease. They highlighted relevant areas of the network using four distinct ways to visualize gradients and occlusions. Their neural network model achieved 77% accuracy.

According to a paper by K.S. Biju et al. [15], the authors presented a mathematical model that uses MRI slices to create a 3 dimension representation of the brain that can indicate whether Alzheimer's disease is present. A number of processing steps are carried out on MRI slices, including up sampling, segmentation, 3D construction, and remnant volume calculations. Then this 3D image is inspected by a medical professional.

3 Proposed System

3.1 Dataset Description

The dataset is derived from the Kaggle an open source platform [34] which contains around 6500 images comprising of four classes namely moderately demented, very mildly demented and non-demented, mildly demented. The dimension of images used in the dataset is of form 176x208. The images are rescaled into 180x180. The Images for the four mentioned classes are shown below in the Table 1.

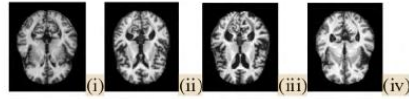


Figure 1.1: Samples of brain MR images different AD stage. (i) Nondemented; (ii) moderate dementia; (iii) mild dementia; (iv) very mild dementia.

An analysis of the dataset shows that the dataset is clearly class imbalanced. To solve the problem of class imbalance, a Synthetic Minority Oversampling method (SMOTE) is applied. By this method minority classes are duplicated in the dataset by random chance.

Table 1. Dataset Distribution.

Class	Number of Images
Non-Demented	2240
Moderate Demented	3200
Mild Demented	896
Very Mild Demented	64

3.2 Detection Using Convolutional Neural Network

The datasets are pre-processed and normalized before they are applied to the network model to extract the features that indicate Alzheimer's affected regions. The proposed architecture includes convolutional layers with a linear activation function (ReLU), a dense layer, a flatten layer, and a max-pooling layer. The hidden layer enables the network to learn and perform faster than any other function by using this activation function.

1. Input Layer: As input, normalized MRI image datasets are given as the first layer of the presented neural network model.

2. Convolutional Layer: These layers are the building blocks of CNN structure. These layers transform the input data which is images in our case, a dot product between the input and output layers will be computed based on their weights.
3. Pooling Layers: The purpose of using these layers is to control overfitting and reduce the data representation. On each input, Max pooling layers are applied to reduce the computational cost of subsequent layers
4. Dropout Layer: The dropout is regularization technique for neural networks. During training, neurons in the hidden layers are randomly dropped out of the dropout layer. By eliminating some of the neurons, this layer minimizes overfitting in the model.
5. Dense Layer: By using flatten layer, high-dimensional data becomes column vectors, which follows the convolutional layer, and then there is a dense layer, which does the same functions as a neural network.

3.3 Optimization Algorithms

As the optimal algorithm, the proposed model is trained with the Adaptive Moment Estimation (Adam) algorithm. This method is very efficient when working big data set or parameters. This algorithm is a combination of the root mean square propagation algorithm and gradient descent algorithm. This algorithm takes the best characteristics of the two methods and builds on them to produce a more optimized gradient descent algorithm. The rule is given by equation 1, 2, and 3.

$$w_{t+1} = w_t - m_t \wedge \left(\frac{\alpha}{\sqrt{v_t} + \epsilon} \right) \quad (1)$$

$$m_t = \frac{m_t}{1 - \beta_1^t} \quad (2)$$

$$v_t = \frac{v_t}{1 - \beta_2^t} \quad (3)$$

Where, w_t =weights at time, w_{t+1} =weights at time t+1, α =learning rate, ϵ =small positive constant, m_t =bias corrected weight parameters, m_t =bias corrected, v_t =sum of square of past gradients, β_1^t & β_2^t =decay rates of average of gradients. Adam optimizer gives high performance and outperforms other optimization algorithms.

3.4 Evaluation Metrics

The proposed model is evaluated by calculating recall, AUC, F1-score, precision, accuracy and Cohen's kappa. To check the efficiency metrics, the confusion matrix have been used. In order to find the model's accuracy, we need to predict the true positives and negatives with a high degree of accuracy. Using the equation 4 the accuracy of the model is calculated.

$$Accuracy = \left(\frac{TN+TP}{TP+FP+TN+FN} \right) \quad (4)$$

The TP value is returned if both the image and label are normal. The value of FP indicates that the image was predicted as normal but the actual image was abnormal. The value FN corresponds to an abnormal image predicted by the model but the real label is normal.

A model's precision (PR) is estimated by dividing the true positive observations by the total positive prediction. The precision is calculated using the equation 5.

$$PR = \left(\frac{TP}{TP+FP} \right) \quad (5)$$

To achieve the highest level of precision, the recall must be considered, also known as the sensitivity or true positive rate (TPR). In order to calculate recall, we use equation 6.

$$recall = \left(\frac{TP}{TP+FN} \right) \quad (6)$$

The F1-score value ranges from 0 to 1, elucidated by the weighted average of the recall and precision. Using equation (7) F1-score is found.

$$F1 - Score = \left(\frac{2 \cdot PR \cdot REC}{PR + REC} \right) \quad (7)$$

Equation (8) is used to calculate Cohens kappa (CK) score. The Cohens kappa is defined as:

$$CK = \left(\frac{po - pe}{1 - pe} \right) \quad (8)$$

Where po is observed accuracy and pe is expected accuracy.

4 Experimental Results

The model training is performed using parameter 100 epoch, batch size of 6500. The model attains a validation accuracy of 94% and training accuracy of 98%. In order to determine whether the model is correctly distinguishing positive and negative categories, at every epoch, AUC (Area Under Curve) is determined. The confusion matrix is used to calculate the each and every class metrics and shown in table 2.

Table 2. Evaluation metrics of Individual class

Disease	PR	REC	F1-Score
ND	1	0.92	0.96
VMD	1	1	1
MD	0.9	0.94	0.92
MOD	0.9	0.92	0.91

On the testing dataset, ND, VMD, MD, and MOD all performed well in terms of F1-score, recall and precision.

5 Conclusion

In this paper, CNN model is proposed to predict the class of Alzheimer disease among the given 4-classified images. The proposed model is trained and tested using Kaggle dataset. The major disadvantage of the dataset is the class disparity. The SMOTE method is employed to resolve this issue. The proposed model has been tested with testing data consisting of four classes and accomplish a 94.68% accuracy with AUC of 98%. In this way, it is well suited to identify brain areas often associated with AD and can facilitate physicians' decision making by helping them determine each patient's AD severity level according to the level of dementia.

References

1. Fuse Hiroki; Oishi Kota; Maikusa Norihide; Fukami Tadanori; (2018). Detection of Alzheimer's Disease with Shape Analysis of MRI Images. Japanese Alzheimer's Disease Neuroimaging Initiative.
2. H. M. Tarek Ullah; Zishan Ahmed Onik; Riashat Islam; Dr. Dip Nandi; (2018). Alzheimer's Disease And Dementia Detection From 3D Brain MRI Data Using Deep Convolutional Neural Networks. 2018 3rd International Conference for Convergence in Technology.
3. Khan, A., & Zubair, S. (2020). A Machine Learning-based robust approach to identify Dementia progression employing Dimensionality Reduction in Cross-Sectional MRI data. 2020 First International Conference of Smart Systems and Emerging Technologies.
4. Subramanyam Rallabandi, V. P., & Seetharaman, K. (2021). Machine Learning-Based Classification of Dementia Types: MRI Study. 2021 International Conference on Artificial Intelligence and Smart Systems.
5. Neelaveni, J.; Devasana, M.S.Geetha (2020). 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS) - Alzheimer Disease Prediction using Machine Learning Algorithms.
6. Marcus, Daniel S.; Wang, Tracy H.; Parker, Jamie; Csernansky, John G.; Morris, John C.; Buckner, Randy L. (2007). Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI Data in Young, Middle Aged, Nondemented, and Demented Older Adults. Journal of Cognitive Neuroscience.
7. Malone, Ian B.; Cash, David; Ridgway, Gerard R.; MacManus, David G.; Ourselin, Sebastien; Fox, Nick C.; Schott, Jonathan M. (2013). MIRIAD—Public release of a multiple time point Alzheimer's MR imaging dataset.
8. Tanveer, M., Richhariya, B., Khan, R. U., Rashid, A. H., Khanna, P., Prasad, M., & Lin, C. T. (2020). Machine Learning Techniques for the Diagnosis of Alzheimer's Disease. ACM Transactions on Multimedia Computing, Communications, and Applications.
9. Fan, Zhao; Xu, Fanyu; Qi, Xuedan; Li, Cai; Yao, Lili (2019). Classification of Alzheimer's disease based on brain MRI and machine learning. Neural Computing and Applications.
10. Acharya, U. Rajendra; Fernandes, Steven Lawrence; WeiKoh, Joel En; Ciacio, Edward J.; Fabell, Mohd Kamil Mohd; Tanik, U. John; Rajinikanth, V.; Yeong, Chai Hong (2019). Automated Detection of Alzheimer's Disease Using Brain MRI Images— A Study with Various Feature Extraction Techniques. Journal of Medical Systems.
11. Eke, Chima Stanley; Jammeh, Emmanuel; Li, Xinzhong; Carroll, Camille; Pearson, Stephen; Ifeakor, Emmanuel (2020). Early Detection of Alzheimer's Disease with Blood Plasma Proteins using Support Vector Machines. IEEE Journal of Biomedical and Health Informatics.
12. R. Jain, N. Jain, A. Aggarwal and D. J. Hemanth (2019), Convolutional neural network based Alzheimer's disease classification from magnetic resonance brain images.
13. Ge, Chenjie; Qu, Qixun; Gu, Irene Yu-Hua; Store Jakola, Asgeir (2019). 2019 IEEE International Conference on Image Processing (ICIP) - Multiscale Deep Convolutional Networks for Characterization and Detection of Alzheimer's Disease Using MR images.
14. Stoyanov, Danail; Taylor, Zeike; Kia, Seyed Mostafa; Oguz, Ipek; Reyes, Mauricio; Martel, Anne; Maier-Hein, Lena; Manquand, Andre F.; Duchesnay, Edouard; Löfstedt, Tommy; Landman, Bennett; Cardoso, M. Jorge; Silva, Carlos A.; Pereira, Sergio; Meier, Raphael (2018). Understanding and Interpreting Machine Learning in Visualizing Convolutional Networks for MRI-Based Diagnosis of Alzheimer's Disease. First International Workshops.
15. Biju, K.S.; Alfa, S.S.; Lal, Kavya; Antony, Alvia; Akhil, M Kurup (2017). Alzheimer's Detection Based on Segmentation of MRI Image. Procedia Computer Science.
16. Mehmood, A., Yang, S., Feng, Z., Wang, M., Ahmad, A. S., Khan, Rizwan Khan, Muazzam Maqsood, Yaqub, M. (2021). A Transfer Learning Approach for Early Diagnosis of Alzheimer's Disease on MRI Images.
17. A. De and A. S. Chowdhury (2021), DTI based Alzheimer's disease classification with rank modulated fusion of CNNs and random forest, Expert Syst. Appl.
18. J. Shi, X. Zheng, Y. Li, Q. Zhang and S. Ying (2018), Multimodal neuroimaging feature learning with multimodal stacked deep polynomial networks for diagnosis of Alzheimer's disease.
19. Islam, Jyoti; Zhang, Yancing (2018). Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks. Brain Informatics.
20. Mandal, Pravat K.; Mahajan, Rashima; Dinov, Ivo D.; Mandal, Pravat K. (2012). Structural Brain Atlases: Design, Rationale, and Applications in Normal and Pathological Cohorts. Journal of Alzheimer's Disease.
21. Jing Zhang; Chunshui Yu; Guilian Jiang; Weifang Liu; Longzheng Tong (2012). 3D texture analysis on MRI images of Alzheimer's disease.

22. Krishnakumar, V.; Parthiban, Latha (2019). A Novel Texture Extraction Technique with T1 Weighted MRI for the Classification of Alzheimer's Disease. *Journal of Neuroscience*.
23. Xiao, Zhe; Ding, Yi; Lan, Tian; Zhang, Cong; Luo, Chuanji; Qin, Zhiguang (2017). Brain MR Image Classification for Alzheimer's Disease Diagnosis Based on Multifeature Fusion. *Computational and Mathematical Methods in Medicine*.
24. Ben Ahmed, Olfa; Benois-Pineau, Jenny; Allard, Michèle; Ben Amar, Chokri; Catheline, Gwénaëlle (2015). Classification of Alzheimer's disease subjects from MRI using hippocampal visual features. *Multimedia Tools and Applications*.
25. Beheshti, I.; Demirel, H. (2015). Feature-ranking-based Alzheimer's disease classification from structural MRI. *Magnetic Resonance Imaging*.
26. Zhou, Qi; Goryawala, Mohammed; Cabrerizo, Mercedes; Wang, Jin; Barker, Warren; Loewenstein, David A.; Duara, Ranjan; Adjouadi, Malek (2014). An Optimal Decisional Space for the Classification of Alzheimer's Disease and Mild Cognitive Impairment. *IEEE Transactions on Biomedical Engineering*.
27. Prashanthi Vemuri; Jeffrey L. Gunter; Matthew L. Senjem; Jennifer L. Whitwell; Kejal Kantarci; David S. Knopman; Bradley F. Boeve; Ronald C. Petersen; Clifford R. Jack Jr. (2008). Alzheimer's disease diagnosis in individual subjects using structural MR images: Validation studies. *Neuro Image*.
28. Liu, Manhua; Zhang, Daoqiang; Shen, Dinggang (2014). Hierarchical fusion of features and classifier decisions for Alzheimer's disease diagnosis. *Human Brain Mapping*.
29. Khedher, L.; Ramirez, J.; Górriz, J.M.; Brahim, A.; Segovia, F. (2015). Early diagnosis of Alzheimer's disease based on partial least squares, principal component analysis and support vector machine using segmented MRI images. *Neurocomputing*.
30. Ben Ahmed, Olfa; Mizotin, Maxim; Benois-Pineau, Jenny; Allard, Michèle; Catheline, Gwénaëlle; Ben Amar, Chokri (2015). Alzheimer's disease diagnosis on structural MR images using circular harmonic functions descriptors on hippocampus and posterior cingulate cortex. *Computerized Medical Imaging and Graphics*.
31. S. Dubey, Alzheimer's Dataset, (2019), Available at: <https://www.kaggle.com/tourist55/alzheimers-dataset-4-class-of-images>.

Application of Deep Learning to detect different stages of Alzheimer's Disease (1).docx

ORIGINALITY REPORT

7 %	5 %	6 %	2 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dokumen.pub Internet Source	2 %
2	Suriya Murugan, Chandran Venkatesan, M. G. Sumithra, Xiao-Zhi Gao, B. Elakkiya, M. Akila, S. Manoharan. "DEMNET: A Deep Learning Model for Early Diagnosis of Alzheimer Diseases and Dementia From MR Images", IEEE Access, 2021 Publication	1 %
3	ouci.dntb.gov.ua Internet Source	1 %
4	"Data Engineering for Smart Systems", Springer Science and Business Media LLC, 2022 Publication	<1 %
5	Shuang Liang, Yu Gu. "Computer-Aided Diagnosis of Alzheimer's Disease through Weak Supervision Deep Learning Framework with Attention Mechanism", Sensors, 2020 Publication	<1 %

6	www.ecf.utoronto.ca Internet Source	<1 %
7	www.jgenng.com Internet Source	<1 %
8	S. Soundarya, M.S. Sruthi, S. Sathya Bama, S. Kiruthika, J. Dhiyaneswaran. "Early detection of Alzheimer disease using Gadolinium material", Materials Today: Proceedings, 2021 Publication	<1 %
9	arxiv-export-lb.library.cornell.edu Internet Source	<1 %
10	www.frontiersin.org Internet Source	<1 %
11	www.hindawi.com Internet Source	<1 %
12	www.scholars.northwestern.edu Internet Source	<1 %
13	Md. Sarwar Kamal, Aden Northcote, Linkon Chowdhury, Nilanjan Dey, Ruben Gonzalez Crespo, Enrique Herrera-Viedma. "Alzheimer's Patient Analysis Using Image and Gene Expression Data and Explainable-AI to Presents Associated Genes", IEEE Transactions on Instrumentation and Measurement, 2021 Publication	<1 %

14 Prajakta Tambe, Rutuja Saigaonkar, Nidhi Devadiga, Pallavi H. Chitte. "Deep Learning techniques for effective diagnosis of Alzheimer's disease using MRI images", ITM Web of Conferences, 2021 $<1\%$

Publication

15 J. Faritha Banu, K. G. Shanthi, P. Lakshmi Priya, M. Faritha Begum. "Performance Enhancement Architecture of VoIP Applications in Multiprotocol Label Switching Networks", Circuits and Systems, 2016 $<1\%$

Publication

16 Savita Dahiya, S. Vijayalakshmi, Munish Sabharwal. "Alzheimer's Disease Detection using Machine Learning: A Review", 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2021 $<1\%$

Publication

Exclude quotes On

Exclude matches Off

Exclude bibliography On