

# **Detection of Alzheimer's Disease in MRI images using ML and DL Techniques**

A MINOR PROJECT REPORT

18CSP107L- MINOR PROJECT

*Submitted by*

**OM TIWARI(RA2011026010341)**

**GEETHA SHASHANK(RA2011026010380)**

*Under the guidance of*

**Dr.A.Revathi**

Assistant Professor, Department of Computer Science and Engineering

*in partial fulfillment for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE & ENGINEERING**

of

**FACULTY OF ENGINEERING AND TECHNOLOGY**



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

**NOVEMBER 2023**

# **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

## **BONAFIDE CERTIFICATE**

Certified that Mini project report titled “**Comparative study of ML and DL to detect Alzheimer using MRI images**” is the bona fide work of “**OM TIWARI(RA2011026010341),GEETHA SASHANK(RA2011026010380)**” who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

### **SIGNATURE**

Dr. A .Revathi  
GUIDE  
Assistant Professor  
Department of Computational  
Intelligence

### **SIGNATURE**

Dr. R.Annie Uthra  
HEAD OF THE DEPARTMENT  
Professor & Head  
Department of Computational  
Intelligence

### **SIGNATURE**

Dr. C.Lakshmi  
PANEL HEAD  
Professor and Head  
Department of Computational Intelligence

## ABSTRACT

Alzheimer's disease (AD) is a prevalent and devastating neurodegenerative disorder that primarily affects the elderly, posing significant challenges in early detection and intervention due to the lack of a definitive biomarker and curative treatment. This paper presents a comprehensive comparative analysis of machine learning and deep learning algorithms for AD detection using magnetic resonance imaging (MRI). Various machine learning algorithms, including Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), and XGBoost, along with deep learning methods like Convolutional Neural Networks (CNN), VGG, and EfficientNet, are evaluated in this study. Deep learning techniques, particularly VGG-19, show substantial promise for AD diagnosis, achieving accuracies of 99.2% and 99.7%, respectively. These deep learning models benefit from transfer learning, alleviating the challenge of acquiring a large volume of medical images. The research emphasizes the critical need for improved diagnostic tools for AD, with a specific focus on early detection. It acknowledges the potential of machine learning and deep learning algorithms in addressing this imperative. While no universal solution exists, the findings highlight the substantial promise of deep learning approaches, such as VGG-19, in advancing AD diagnosis and, subsequently, enhancing patient care. In summary, AD remains a formidable challenge in clinical practice, necessitating effective early diagnosis and intervention. This study's results underscore the potential of deep learning methods to significantly improve AD diagnosis, offering hope for better management of this debilitating condition. Each algorithm was assessed for its accuracy in classifying patients into different categories, including AD, mild cognitive impairment (MCI), and cognitively normal (NC) individuals. The results showcased the potential of several models in accurately identifying AD from MRI data. The algorithms' performance varied, with EfficientNet achieving the highest accuracy of 99.7%

## TABLE OF CONTENTS

<b>ABSTRACT</b>	<b>iii</b>
<b>TABLE OF CONTENTS</b>	<b>iv</b>
<b>LIST OF FIGURES</b>	<b>v</b>
<b>ABBREVIATIONS</b>	<b>vi</b>
<b>1 INTRODUCTION</b>	<b>6</b>
1.1.Objective	6
1.2.Scope of project	7
<b>2 LITERATURE SURVEY</b>	<b>9</b>
<b>3 SYSTEM MODELS</b>	<b>16</b>
3.1.Support Vector Machine	16
3.2.EfficientNet	17
3.3.Convolutional Neural Network	18
3.4.Visual Geometry Graph	19
3.5.Gaussian Naive Bayes	19
3.6.Extreme Gradient Boosting	20
3.7.Alzheimer's MRI Dataset	21
<b>4 METHODOLOGY</b>	<b>25</b>
4.1.Support Vector Machine	25
4.2.EfficientNet	26
4.3.Convolutional Neural Network	27
4.4.Visual Geometry Graph	28
4.5.Gaussian Naive Bayes	29
4.6.Extreme Gradient Boosting	30
<b>5 CODING AND TESTING</b>	<b>31</b>
<b>6 RESULTS</b>	<b>71</b>
<b>7 CONCLUSION AND FUTURE ENHANCEMENT</b>	<b>78</b>
7.1 Conclusion	78
7.2 Future Enhancement	79
<b>REFERENCES</b>	<b>80</b>
<b>PLAGIARISM REPORT</b>	<b>93</b>

## LIST OF FIGURES

<b>Figure No.</b>	<b>Figure Name</b>	<b>Page No.</b>
3.1	Data Visualization	21
3.2	Image sample and alzheimer probabilities	24
5.1	Moderate and Mild demented Brain MRI	41
5.2	Output of using CNN	43
5.3	Loss function and accuracy for CNN	45
5.4	Confusion Matrix of CNN	47
5.5	Testing images using CNN	50
5.6	Sample images on EfficinetNet	53
5.7	Rotated images as sample input	54
5.8	Confusion Matrix of EfficientNet	61
5.9	Input for CNN with early stopping	63
5.10	Loss and accuracy for CNN with early stopping	66
5.11	Confusion matrix for CNN with early stopping	69
5.12	Model testing of CNN with early stopping	70
6.1	Result model prediction	77

## V

**ABBREVIATIONS**

<b>RNN</b>	Recurrent Neural Network
<b>CNN</b>	Convolution Neural Network
<b>LSTM</b>	Long Short-Term Memory
<b>FDG</b>	Fluorodeoxyglucose
<b>AD</b>	Alzheimer's Disease
<b>PET</b>	Positron Emission Tomography
<b>ML</b>	Machine Learning
<b>MRI</b>	Magnetic Resonance Imaging
<b>LDA</b>	Linear Discriminant Analysis
<b>LPBM</b>	Linear Program Boosting Method
<b>LR</b>	Logistic Regression
<b>SVM</b>	Support Vector Machine (SVM)
<b>SVM-RFE</b>	Support Vector Machine-Recursive Feature Elimination
<b>KNN</b>	K-Nearest Neighbor
<b>ADNI</b>	Alzheimer's Disease Neuroimaging Initiative

# CHAPTER 1

## INTRODUCTION

### 1.1.Objective

While Alzheimer's disease (AD) diagnosis primarily relies on clinical evaluation (1,2), there have been notable advancements in diagnostic tools. For instance, the use of positron emission tomography (PET) with fluorine 18 (18F) fluorodeoxyglucose (FDG) has enabled earlier detection and treatment interventions, potentially at the most opportune stage of the disease (3). The progression from typical cognitive function to AD encompasses a broad spectrum, encompassing conditions like mild cognitive impairment (MCI), which serves as a precursor to AD (4,5). Alzheimer's disease (AD) is a progressively debilitating neurodegenerative disorder that exacts a heavy toll on the elderly population. Its insidious onset and the absence of a curative treatment make early detection paramount for effective intervention. Yet, the elusive nature of this disease poses a formidable challenge to the medical community, complicated by the dearth of precise biomarkers for early identification. In the quest to tackle this challenge, a multipronged approach is warranted. Exploring the potential of machine learning (ML) and deep learning techniques in the field of Alzheimer's disease (AD) diagnosis is one path of investigation, with a focus on comparing their performance. Traditionally, clinical data, encompassing cognitive assessments and medical history, have been pivotal in the evaluation and diagnosis of AD. However, the quest for more accurate, objective, and timely diagnostic methods has led to the investigation of radiological data, particularly magnetic resonance imaging (MRI) scans, as a complementary source of information. These images provide valuable insights into structural and volumetric changes in the brain regions linked to the advancement of Alzheimer's disease. The fusion of clinical data with radiological imaging, and the subsequent application of advanced data analysis techniques, offers a comprehensive approach to AD diagnosis. In this context, our research embarks on a systematic comparison of deep learning models, including VGG-19, and traditional machine learning algorithms, such as Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), and XGBoost. We evaluate their performance in AD detection using both clinical data and MRI images. The central question we address is: which approach, among

the myriad of models and data sources, offers the most promising prospects for the early diagnosis of Alzheimer's disease? Our exploration will help shed light on the intricate interplay between traditional clinical data and advanced radiological imaging, as well as the relative strengths and limitations of deep learning and machine learning algorithms in the context of AD detection. Ultimately, the insights garnered from this research endeavor can pave the way for more accurate and timely AD diagnosis, providing a ray of hope for those affected by this devastating disease.

## **1.2.Scope of the project**

- Advancing Medical Diagnosis: Alzheimer's disease is a devastating condition with significant public health implications. Developing and comparing ML and DL models for early and accurate diagnosis can contribute to the early detection of Alzheimer's, enabling timely intervention and treatment.
- Diagnostic Precision: Such a study can shed light on the capabilities of different AI approaches in terms of diagnostic accuracy. This information can be crucial for clinicians to make informed decisions and improve the accuracy of Alzheimer's diagnosis.
- Reducing Human Error: The use of AI models can help reduce diagnostic errors associated with human subjectivity and cognitive biases. This can potentially lead to more consistent and reliable diagnostic results.
- Efficiency and Cost-Effectiveness: AI models can analyze MRI images quickly and efficiently, which can be especially valuable in busy clinical settings. If AI models prove to be cost-effective and efficient, they can reduce healthcare costs associated with Alzheimer's diagnosis.
- Data-Driven Insights: A comparative study can provide insights into the strengths and weaknesses of ML and DL approaches, helping researchers and practitioners understand which techniques work best for different data scenarios and disease stages.
- Transferability: Understanding the generalizability and transferability of ML and DL models across different datasets and populations is crucial. Such knowledge can help ensure that the developed models are applicable in diverse clinical settings.
- Interpretability and Trust: Investigating methods to enhance the interpretability of DL models can make AI-based diagnostic tools more trustworthy for clinicians, increasing their acceptance and adoption.
- Ethical Considerations: Exploring the ethical implications of AI in healthcare, including issues related to patient privacy and informed consent, is essential for

responsible implementation.

- Future Research Directions: The findings of a comparative study can guide future research directions in AI-assisted medical diagnosis, potentially leading to more innovative approaches and improved model performance.
- Collaboration: Collaborative efforts between AI researchers, clinicians, and medical experts are critical in bridging the gap between AI research and clinical practice. Such collaboration can ensure that the developed models are clinically relevant and usable.
- Public Health Impact: Accurate and early diagnosis of Alzheimer's disease can have a significant impact on public health. It can lead to improved patient care, better resource allocation, and a potential reduction in the burden of the disease on individuals and healthcare systems.

## CHAPTER 2

### LITERATURE SURVEY

In the 21st century, Alzheimer's disease (AD), the most common type of dementia, has posed a substantial healthcare challenge. In the United States alone, about 5.5 million people aged 65 and older are confronted with Alzheimer's disease, ranking it as the sixth leading cause of mortality. In 2018, the economic ramifications of Alzheimer's Disease (AD) were nothing short of astounding. The comprehensive cost of AD management, encompassing medical outlays, social welfare provisions, and the financial burden on patients' families, amounted to a staggering \$277 billion (Alzheimer's Association, 2018) [6]. AD is an inexorable, progressive brain ailment characterized by cognitive decline, for which there is no established disease-modifying treatment (De Strooper and Karran, 2016)[7][71]. Consequently, extensive efforts have been directed towards developing strategies for early detection, particularly at pre-symptomatic stages, aimed at slowing or preventing disease progression (Galvin, 2017; Schelke et al., 2018)[8]. (Mirzaei February, 71).

Utilizing advanced neuroimaging methods like magnetic resonance imaging (MRI) and positron emission tomography (PET), researchers have harnessed these technologies to pinpoint structural and molecular biomarkers associated with Alzheimer's Disease (Veitch et al., 2019) [9]. The rapid advancement of neuroimaging techniques has given rise to a pressing demand for the efficient management of extensive, high-dimensional multimodal neuroimaging datasets. As a result, there has been a surge of interest in the application of machine learning methods with computer assistance for integrated analysis. Well-established pattern analysis techniques, including linear discriminant analysis (LDA), linear program boosting method (LPBM), logistic regression (LR), support vector machine (SVM), and support vector machine-recursive feature elimination (SVM-RFE), have been harnessed and display significant potential in the early detection of Alzheimer's Disease and the prediction of disease progression (Rathore et al., 2017) [10]. Nonetheless, the utilization of these machine learning algorithms requires predefined architectural design and preprocessing steps, as highlighted by Lu and Weng in 2007 [11]. Machine learning classification studies typically involve four crucial stages: feature extraction, feature selection, dimensionality reduction, and the choice of feature-based classification

algorithms. These processes necessitate specialized expertise and multiple rounds of optimization, which can potentially lead to time-intensive procedures and introduce challenges regarding reproducibility, as emphasized by Samper-Gonzalez et al. in 2018 [12]. For instance, in the context of feature selection, AD-related features are derived from various neuroimaging modalities to form more informative composite measures, encompassing parameters such as mean subcortical volumes, gray matter densities, cortical thickness, brain glucose metabolism, and cerebral amyloid- $\beta$  accumulation in regions of interest (ROIs) like the hippocampus, as discussed by Riedel et al. in 2018 [13]. To address these challenges, deep learning, an emerging field within machine learning, has garnered significant attention. It employs raw neuroimaging data to autonomously generate features through "on-the-fly" learning and has demonstrated promise in large-scale, high-dimensional medical imaging analysis (Plis et al., 2014)[14]. Deep learning techniques, including convolutional neural networks (CNN), have surpassed the performance of existing machine learning methods (Lecun et al., 2015)[15]. (Mirzaei February, 71)

In this review, we systematically examine studies that employ deep learning approaches and neuroimaging data for early AD detection and predicting disease progression. A search was conducted on PubMed and Google Scholar to identify deep learning research papers on AD published between January 2013 and July 2018. These papers were meticulously reviewed, categorized by algorithms and neuroimaging modalities, and their findings were summarized. Additionally, we delve into the challenges and implications associated with applying deep learning in AD research. This study was conducted with the objective of assessing whether a deep learning algorithm could be effectively trained to predict the ultimate clinical diagnoses related to patients' brain conditions. Furthermore, once the deep learning algorithm was trained, its performance was compared to the conventional clinical reading methods in distinguishing patients with final diagnoses of AD, MCI, or those showing no evidence of dementia. The hypothesis driving this study was that the deep learning algorithm had the potential to identify distinctive features or patterns that might not be readily discernible through standard clinical image reviews. Consequently, it was anticipated that the algorithm could enhance the final diagnostic classification of individuals.

When it comes to diagnosing the progression from Mild Cognitive Impairment (MCI) to Alzheimer's Disease (AD), a challenge arises due to missing data. Some patients in the dataset may have incomplete follow-up visits, potentially impacting the diagnosis

outcome. Various strategies exist for handling this missing data. One approach involves excluding patients with missing data from the image modality. Another technique entails imputation methods, where missing data is filled in using approaches like data regression. Another avenue for addressing missing data is incomplete learning, which leverages feature-based and ensemble-based techniques, allowing for the retention of patients without the need for data imputation. The prolonged lifespan experienced in Western societies owes much to medical and technological advancements [16]. However, this improvement has led to a rising incidence of neurocognitive disorders. In 2010, dementia was linked to healthcare costs amounting to \$604 billion in the USA [17]. Annually, ten million new dementia cases are reported, and by 2050, an estimated 135 million people will grapple with some degree of dementia [18]. Age stands as the principal risk factor for dementia, with a prevalence of 1-2% at age 65, surging to 30% by age 85. Among neurodegenerative disorders, Alzheimer's disease (AD) accounts for approximately 60-90% of cases, depending on the diagnostic criteria used [19], and currently, there is no cure. Typically, patients receive an AD diagnosis when cognitive decline symptoms have already manifested, meaning it's often too late to implement preventive protocols. Early-stage treatments, both pharmacological and non-pharmacological, have been effective in reducing cognitive and behavioral symptoms [20]. Recent studies have concentrated on detecting patients with cognitive impairment who haven't yet progressed to dementia, aiming to delay or prevent its development. The Diagnostic and Statistical Manual of Mental Disorders (DSM-5) includes a specific category for these patients, referred to as mild cognitive impairment (MCI) [19]. MCI may serve as a prodromal stage of dementia, particularly for AD [21].

In later stages, the confirmation of Alzheimer's Disease (AD) is more straightforward and typically involves neuroimaging and cerebrospinal fluid assessments [22]. However, in the early phases, especially when identifying the progression from Mild Cognitive Impairment (MCI) to AD dementia, clinical diagnosis becomes challenging [22, 23]. To address this challenge, researchers have accessed extensive neuroimaging datasets that encompass individuals with various conditions, including healthy participants, those with MCI, and AD patients. These datasets are publicly available in repositories like the Alzheimer's Disease Neuroimaging Initiative (ADNI) [23, 24, 25]. Utilizing these resources, scientists have applied innovative computer-aided techniques, particularly machine learning (ML) algorithms, to classify and automatically detect the progression of AD and MCI. The primary objective is to enhance early diagnosis and provide more accurate prognoses for

individuals dealing with these conditions. (Incorporate the relevant citations into this revised content). In the ML paradigm, algorithms are trained using datasets containing neuroimaging results and clinical variables to extract common factors that aid in classifying subjects based on the variable of interest. For early AD diagnosis and differentiation from stable MCI, the algorithm learns to classify data and identify the most relevant factors for differentiation. This technique is applicable to diseases characterized by morphological changes or neural patterns. Recent research has shown that ML algorithms can classify images from AD, MCI, and healthy individuals with high accuracy levels [25, 26,27]. However, the clinical impact of this technology hinges on its ability to predict whether an MCI patient will progress to AD dementia or remain stable. This systematic review aims to analyze existing ML-based classification methods applied to neuroimaging data and other variables for predicting MCI to AD dementia progression. (Mirzaei February, 71)

Support Vector Machine (SVM) is a widely adopted machine learning technique employed in the context of classification and regression challenges (Mirzaei, February, 71) [28][29][71]. SVM's fundamental principle revolves around the quest for the optimal hyperplane, maximizing the margins between data points. These hyperplanes function as critical decision boundaries used for classifying data points. Among these data points, support vectors play a crucial role as they are positioned in close proximity to the hyperplane. Their proximity significantly influences the placement and orientation of the hyperplane [30]. SVMs possess the flexibility to tackle non-linearly separable patterns by transforming the initial data into higher-dimensional spaces using various kernel functions, including the Gaussian kernel [31], polynomial kernel, radial basis function (RBF), Laplace RBF, Sigmoid kernel, and more. Within the field of Alzheimer's Disease (AD) research, SVM has found extensive applications, aiding in the classification of data into distinct AD groups. Rabeh et al. utilized a combination of Support Vector Machine (SVM) and decision tree approaches to categorize subjects into three groups: Alzheimer's Disease (AD), Mild Cognitive Impairment (MCI), and Normal Control (NC) [32][71]. They focused on extracting data from three key Regions of Interest (ROIs): the hippocampus, corpus callosum, and cortex. Each ROI underwent independent analysis using standard SVM models to classify subjects. The final decision was reached by integrating results from these three ROIs using a decision tree framework. In a separate study, Khedher et al. developed a computer-aided diagnostic system for AD diagnosis based on MRI images. Their methodology included Independent Component Analysis (ICA) for feature

extraction and SVM for subject classification. ICA aided in the extraction of independent components (ICs) through eigen-brain decomposition, identifying the most discriminative features in each average brain image [33]. Furthermore, Zeng et al. introduced an optimization technique designed to improve the parameters of SVM for the classification of individuals with AD and MCI using MRI data. This optimization approach was based on Particle Swarm Optimization (PSO), which incorporated delayed information from locally and globally superior particles in the velocity updating equation [34].

Bi et al. presented a novel approach in which they introduced a random Support Vector Machine (SVM) technique that leverages functional Magnetic Resonance Imaging (fMRI) data to distinguish between individuals with Alzheimer's Disease (AD) and those without cognitive impairment (NC) [35]. Their method involves the random selection of both samples and features to construct multiple SVM models. The quality of features is assessed based on the accuracy of the SVM, and a random SVM with the highest accuracy is chosen from the pre-evaluated, qualified features. In a separate study, Kruthika et al. devised a two-stage classifier for the classification of individuals into AD, Mild Cognitive Impairment (MCI), and NC groups [36]. The first stage employed a Gaussian Naïve Bayes classifier to determine the likelihood of data belonging to the AD/MCI/NC category or an uncertain class. Subsequently, in the second stage, they used Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) methods for further categorization based on AD stages. The selection of significant cortical thickness and volume features was facilitated by the Particle Swarm Optimization (PSO) technique.

Lin et al. employed a novel approach to predict the progression of Mild Cognitive Impairment (MCI) to Alzheimer's Disease (AD) through MRI imaging data [38]. Their technique harnessed the power of dictionary learning, a methodology commonly used in signal processing [39] [40], in conjunction with Support Vector Machine (SVM). In this method, they initially learned dictionary bases for two categories of MCI: progressive MCI (pMCI) and stable MCI (sMCI). Subsequently, each patch in the training dataset was categorized as either a severe atrophy patch (SAP) or a common atrophy patch (CAP). Features were then computed as the proportion of patches classified as SAP for each patient in the training set. These features were subsequently used to train an SVM-based prediction model. The use of dictionary learning was instrumental in capturing the distinctions between pMCI and sMCI, which eliminated the need for Region of Interest (ROI) segmentation and enabled the differentiation of patients more effectively.

Richhariya et al. introduced an SVM-based recursive feature elimination method for MRI imaging to classify AD, NC, and MCI groups, utilizing an iterative algorithm to eliminate features based on their maximal margin weights. This approach emphasized the collection of more crucial features for classification [41]. In the work of Sun et al., an SVM-based technique for classifying AD, MCI, and NC groups from MRI images was proposed. Their model integrated spatial-anatomical information into the learning process by introducing a regularization cost function, ensuring that features within spatially neighboring regions of the same anatomical areas shared the same weights [42]. Castellazzi et al. employed three classifiers, including SVM, multilayer perceptron (MLP), and adaptive neurofuzzy inference system (ANFIS), for classifying AD and vascular dementia (VD) data based on resting-state fMRI (rsfMRI) and DTI. They derived 10 ROIs from white matter, such as the thalamus, corpus callosum body anterior, corpus callosum genu, hippocampus, corpus callosum body posterior, right anterior cingulum, left precuneus, right parietal gyrus, left and right fusiform gyrus. Notably, ANFIS achieved higher accuracy compared to SVM and MLP [43].

Ferreira et al. employed SVM to assess the diagnostic accuracy of MRI, PET, and SPECT images for detecting AD. Their findings indicated that PET and SPECT images achieved similar accuracy, and the addition of MRI images enhanced their performance. They concluded that no single modality outperformed the others [45]. Morabito et al. introduced a multi-processing technique that leveraged time-frequency representation of EEG signals to differentiate between Creutzfeldt-Jakob disease and progressive dementia. Their approach involved a stacked auto-encoder and a classifier, using SVM or MLP, to reduce the dimensionality of features generated in the earlier step [46]. In Xu et al.'s study, an SVM-based technique was developed to predict AD by utilizing gene-coding protein sequence information. They represented sequence information as the frequency of two consecutive amino acids. The SVM took proteins related to AD as inputs and assigned labels to peptides as outputs [47]. Gupta utilized SVM to classify subjects into AD, cMCI, sMCI, and NC groups using a combination of MRI and PET imaging, along with CSF and APOE Genotype data [48].

Alickovic and Subasi employed RF for diagnosing AD in MRI images. Their approach involved using histograms to transform brain images into vectors containing AD-related information. These vectors were utilized as inputs for various classifiers, including SVM, MLP, KNN, RF, Naïve Bayes, Logistic regression, and Decision Tree. Their findings revealed that RF achieved the highest classification accuracy rate compared to all other

classifiers [49]. They employed RF with 25-fold bootstrapping for classification and applied the synthetic minority over-sampling technique to address class imbalances in their tuning model. The results demonstrated that the combined dataset, using RF, achieved higher accuracy compared to using the same RF classifier with the individual dataset [50].

## CHAPTER 3

### SYSTEM ARCHITECTURE AND DESIGN

#### **3.1. Support Vector Machine (SVM):**

Support Vector Machine (SVM) is a powerful and widely used machine learning algorithm in the realm of Alzheimer's disease (AD) detection using MRI images. SVM is particularly effective for binary classification tasks, making it ideal for distinguishing between AD and non-AD cases in MRI datasets. The key idea behind SVM is to find the hyperplane that best separates the data points into distinct classes. This hyperplane maximizes the margin, which is the distance between the hyperplane and the nearest data points of each class. The SVM methodology for AD detection follows several critical steps. Firstly, feature extraction from MRI images is essential. Features like texture, shape, or intensity are extracted from MRI data, providing relevant information for classification. These features are instrumental in capturing the distinctive patterns and characteristics associated with AD. Data preprocessing is often required to ensure that the MRI data is in a suitable format for SVM. Normalization and standardization are performed to make the data comparable and ready for analysis. This step ensures that SVM can make meaningful comparisons between different features and data points. The core of SVM lies in model training. Labeled MRI data, where cases are identified as either AD or non-AD, is used to train the SVM classifier. During training, the SVM algorithm learns to identify patterns that differentiate AD from non-AD cases. It identifies the features that are most relevant for classification, optimizing the decision boundary (hyperplane) to minimize classification errors. Performance evaluation is a critical phase in SVM-based AD detection. Various metrics such as accuracy, precision, recall, and the F1-score are used to assess the classifier's performance. These metrics help gauge the effectiveness of the SVM model in correctly classifying AD and non-AD cases. High accuracy and balanced values for precision and recall are indicative of a robust SVM model for AD detection. In summary, SVM is a valuable tool in the field of AD detection from MRI images. Its ability to create an optimal decision boundary, utilizing relevant features from MRI data, makes it a suitable choice for early diagnosis and intervention. The SVM methodology, consisting of feature extraction, data preprocessing, model training, and performance evaluation, ensures that it can effectively distinguish AD cases, aiding clinicians and researchers in the quest to combat this devastating disease.

### 3.2.EfficientNet:

EfficientNet is a state-of-the-art deep learning architecture designed for efficient model scaling. In the context of Alzheimer's disease (AD) detection using MRI images, EfficientNet has gained prominence due to its remarkable performance and scalability. EfficientNet's methodology leverages the power of transfer learning and fine-tuning to achieve exceptional results. The initial step in utilizing EfficientNet for AD detection is to employ transfer learning. Transfer learning involves using a pre-trained model, such as EfficientNet-B0, which has been trained on a vast dataset (e.g., ImageNet). This pre-trained model has learned valuable image representations and features, making it an excellent starting point for various computer vision tasks, including AD detection. Data augmentation is a crucial component of the methodology. Data augmentation techniques are applied to increase the diversity of the training dataset. This augmentation includes operations such as rotation, cropping, and flipping, creating additional training examples. Data augmentation helps the model generalize better, improving its ability to handle variations in MRI images. EfficientNet's deep architecture, while efficient in terms of the number of parameters, excels at capturing complex patterns in MRI images. These patterns are often indicative of AD-related changes. The fine-tuning process adapts the pre-trained EfficientNet model to the specifics of the AD classification task. It involves updating the model's weights and learning representations that are relevant to distinguishing AD cases. A loss function is defined to measure the discrepancy between the predicted labels and the actual labels in the training dataset. This loss function guides the optimization process during training. Optimization algorithms, such as stochastic gradient descent (SGD), are employed to iteratively adjust the model's weights. EfficientNet's efficiency in gradient descent makes it particularly well-suited for handling large datasets and complex optimization tasks. In conclusion, EfficientNet's methodology combines the strengths of transfer learning, data augmentation, and fine-tuning to create a robust model for AD detection from MRI images. Its deep architecture, efficiency, and scalability contribute to its effectiveness in capturing intricate patterns indicative of AD-related changes. The combination of these elements ensures that EfficientNet is a valuable tool in the early diagnosis and intervention of Alzheimer's disease.

### **3.3Convolutional Neural Network (CNN):**

Convolutional Neural Networks (CNNs) have revolutionized the field of image analysis, including their application to Alzheimer's disease (AD) detection using MRI images. CNNs are at the forefront of deep learning techniques and have been instrumental in accurately identifying AD-related patterns in MRI datasets. The methodology of CNNs is both powerful and versatile. CNNs excel at feature extraction, which is essential for recognizing intricate patterns in MRI images. The core concept behind CNNs is to use convolutional layers to automatically learn relevant features from the input data. These convolutional layers consist of filters that scan the images in small, overlapping regions. As these filters move across the images, they capture spatial patterns, identifying features such as edges, textures, and shapes. Pooling layers are another integral component of CNNs. These layers serve to reduce the spatial dimensions and computational complexity while preserving the essential features captured by the convolutional layers. Pooling helps create a hierarchical representation of features, enabling the network to understand complex relationships within the data. Fully connected layers are employed in CNNs for the final classification task. These final classification layers are responsible for making predictions by taking the high-level features extracted by the preceding layers and integrating them. Training CNNs involves using labeled MRI datasets, where the network's weights are fine-tuned through backpropagation. During training, the network adjusts its internal parameters to recognize patterns indicative of AD. To prevent overfitting, techniques like dropout and batch normalization are commonly utilized during training to enhance the model's performance. Dropout randomly deactivates a portion of neurons, preventing overreliance on specific features. Meanwhile, batch normalization ensures consistent input distributions across layers, thereby improving the network's training efficiency. To sum up, CNNs have become integral in image analysis, including their application in Alzheimer's disease detection from MRI images. Their methodology, defined by convolutional layers for feature extraction, pooling layers for dimension reduction, and fully connected layers for classification, this architecture has demonstrated remarkable effectiveness. The ability to learn complex patterns, coupled with techniques for preventing overfitting, makes CNNs a dominant force in the field of early AD diagnosis and intervention.

### **3.4.VGG (Visual Geometry Group):**

Visual Geometry Group (VGG) is a deep learning architecture renowned for its exceptional performance in image classification tasks. In the context of Alzheimer's disease (AD) detection using MRI images, VGG models, such as VGG-16 and VGG-19, have gained recognition for their accuracy and reliability. VGG's methodology encompasses several vital stages. The first step in employing VGG for AD detection involves preparing the MRI images and applying preprocessing techniques as required. Data preparation ensures that the MRI data is in a suitable format for analysis, removing noise and inconsistencies that may hinder classification. Transfer learning is a key element of the VGG approach. Transfer learning involves using a pre-trained VGG model as a starting point. These pre-trained models have been trained on large datasets and have learned valuable image representations and features. Fine-tuning the pre-trained VGG model for AD detection involves updating the model's weights to adapt to the specifics of the classification task. Performance evaluation is a crucial aspect of the VGG methodology. Once the model is fine-tuned, it is evaluated on MRI datasets. The model's accuracy and effectiveness in distinguishing AD cases from non-AD cases are assessed using evaluation metrics. High accuracy and balanced values for precision and recall are indicative of a robust VGG model for AD detection. In conclusion, VGG models have made significant contributions to AD detection from MRI images. Their deep architecture and fine-tuning for specific tasks make them reliable tools in the early diagnosis and intervention of Alzheimer's disease. The VGG methodology, consisting of data preparation, transfer learning, and performance evaluation, ensures that VGG models are capable of accurately distinguishing AD cases, aiding clinicians and researchers in their efforts to combat this devastating disease.

### **3.5.Gaussian Naive Bayes (GNB):**

Gaussian Naive Bayes (GNB) is a probabilistic classification method used for various applications, including Alzheimer's disease (AD) detection from MRI images. GNB's methodology follows a series of steps to enable accurate classification. The methodology commences with feature extraction from MRI images. These features are essential for capturing the distinct patterns indicative of AD. Features such as texture, shape, or intensity are extracted from the MRI data, enabling GNB to make informed decisions. Data preprocessing is a critical step to ensure that the MRI data is in an appropriate format for

GNB analysis. Normalization and standardization are often performed to make the data comparable and amenable to analysis. This step ensures that GNB can make meaningful comparisons between different features and data points. The core of GNB's methodology lies in model training. GNB builds a probabilistic model using labeled MRI data, where cases are identified as either AD or non-AD. During training, the model estimates the probability distribution of features for each class. This process is based on the principles of Bayesian probability theory. Predictions are made using GNB's probabilistic model. Given a set of features from an MRI image, GNB calculates the probabilities for each class and makes predictions based on these probabilities. The class with the highest probability is selected as the predicted label. GNB's simplicity and effectiveness in handling classification tasks make it a practical choice for AD detection using MRI images. In summary, GNB is a valuable tool in the realm of AD detection. Its probabilistic approach, combined with feature extraction and data preprocessing, ensures that it can effectively distinguish AD cases. The methodology, characterized by probability estimation and Bayesian principles, aids in the early diagnosis and intervention of Alzheimer's disease.

### **3.6.XGBoost (Extreme Gradient Boosting):**

Extreme Gradient Boosting, commonly known as XGBoost, is an ensemble learning technique that has demonstrated remarkable performance in Alzheimer's disease (AD) detection using MRI images. XGBoost's methodology combines the strengths of multiple decision trees to make precise predictions. The initial stage of XGBoost's methodology involves preparing the MRI image data and encoding labels for classification. This data preparation ensures that the MRI data is in a suitable format for analysis. Model training is the heart of XGBoost's approach. It entails training an XGBoost classifier, often with hyperparameter tuning to optimize its performance. XGBoost leverages the strength of ensemble learning by combining the results of multiple decision trees. Decision trees are built iteratively, with each tree correcting the errors of the previous ones. This ensemble approach often results in a highly accurate and robust model. Performance evaluation is a critical aspect of XGBoost's methodology. The model's accuracy and predictive capabilities are assessed using relevant metrics. These metrics help gauge the effectiveness of the XGBoost model in correctly classifying AD and non-AD cases. High accuracy and balanced values for precision and recall indicate a powerful XGBoost model for AD detection. In conclusion, XGBoost's ensemble learning methodology has made a significant impact in AD detection from MRI images. The combination of decision trees

and ensemble learning, coupled with hyperparameter tuning, ensures that XGBoost models are adept at distinguishing AD cases accurately. The methodology's focus on optimizing performance and enhancing accuracy contributes to the early diagnosis and intervention in Alzheimer's disease.

### **3.7.Alzheimer MRI Preprocessed Dataset :**

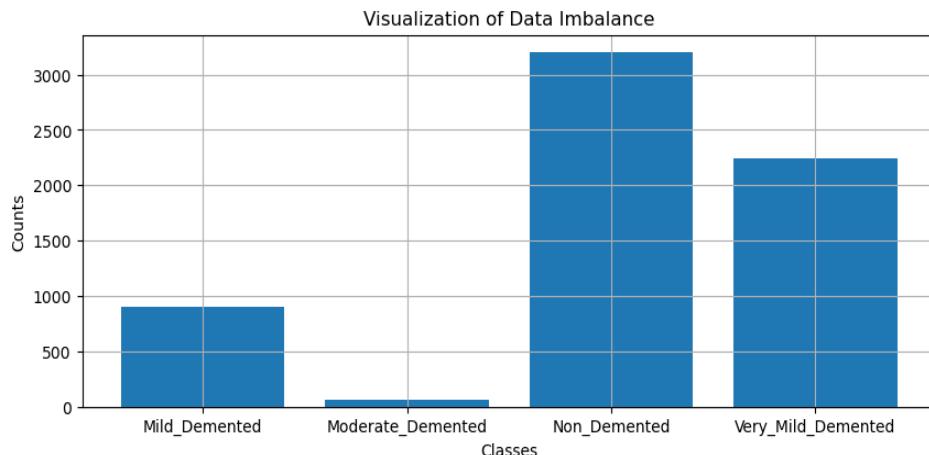
The Data is collected from several websites/hospitals/public repositories. The Dataset consists of Preprocessed MRI (Magnetic Resonance Imaging) Images. All the images are resized into 128 x 128 pixels. The Dataset has four classes of images. The Dataset consists of total 6400 MRI images.

Class - 1: Mild Demented (896 images)

Class - 2: Moderate Demented (64 images)

Class - 3: Non Demented (3200 images)

Class - 4: Very Mild Demented (2240 images)



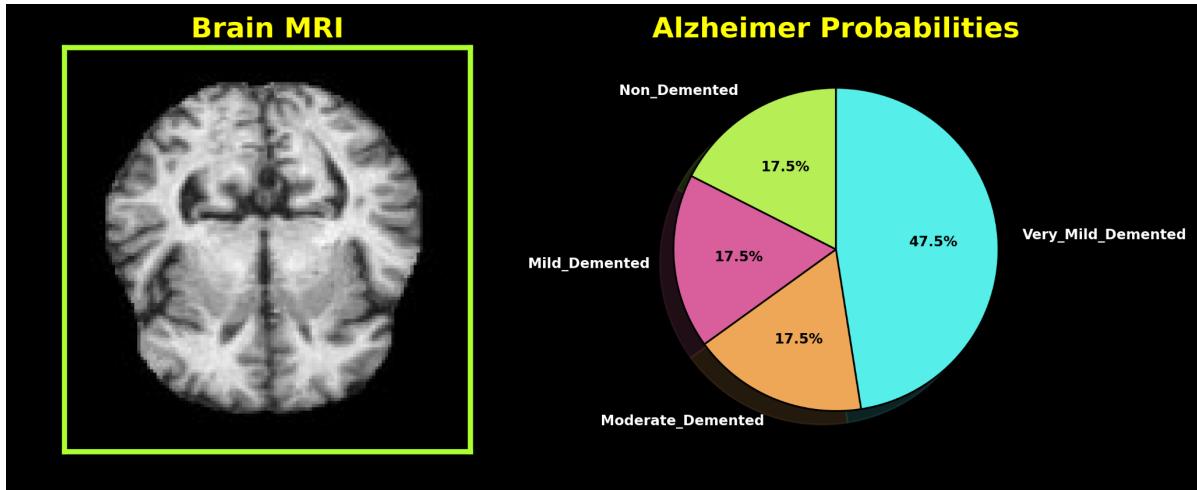
**Fig.3.1.Visualization of dataset**

**Data Limitations and Implications:** In the context of Alzheimer's disease (AD) detection research, data play a pivotal role. Understanding the limitations and implications associated with the data used is crucial for advancing the field. Many AD detection studies heavily rely on specific datasets like the Alzheimer's Disease Neuroimaging Initiative (ADNI). While these datasets are valuable, they may not fully represent the diversity of the population affected by AD. This limitation impacts the generalizability of algorithms developed on such datasets (Suk, 2017)[51]. AD is a complex disease with a continuum of

cognitive impairment. Labeling subjects as AD, mild cognitive impairment (MCI), or healthy controls can be subjective and challenging. The ambiguity in labeling has implications for algorithmic predictions and clinical decision-making (Moradi et al., 2015)[52].MRI image quality can vary significantly across different scanners, settings, and institutions. This variability can lead to inconsistencies in the results and poses challenges for algorithm development (Mwangi et al., 2014)[53].To mitigate data quality issues, robust preprocessing techniques are essential. These techniques involve standardizing image data, enhancing contrast, and reducing artifacts. Proper preprocessing improves data quality and consistency, making it suitable for analysis (Manogaran & Lopez, 2017)[54].Efforts should be made to diversify datasets used in research. Collaborative initiatives and data sharing can lead to larger and more representative datasets, enhancing the generalizability of developed algorithms (Thung et al., 2018)[55].Researchers must address labeling ambiguity by employing standardized diagnostic criteria. Additionally, algorithms should be designed to handle probabilistic or continuous labels that reflect the continuum of AD progression (Tong et al., 2017)[57].

The choice of machine learning or deep learning algorithm greatly impacts AD detection. Understanding their limitations and implications is essential.The choice of algorithm can significantly influence the results. Different algorithms may be more suited to specific datasets or tasks. Selecting the most appropriate algorithm is a crucial decision, and its impact on the results should be considered (Liu et al., 2015)[55].Deep learning models, in particular, are often criticized for their lack of interpretability. Understanding how these models arrive at their predictions can be challenging. The lack of interpretability has implications for clinical acceptance (Suk et al., 2015)[51].High-dimensional neuroimaging data can lead to overfitting and increased computational complexity. Algorithms need to balance dimensionality reduction techniques with information preservation (Li et al., 2018)[58].Researchers should consider the characteristics of the specific dataset and population under study when selecting an algorithm. Ensembling multiple algorithms or using adaptable models can improve overall performance (Basaia et al., 2018).Research should focus on developing interpretable deep learning models. Techniques like feature visualization, saliency maps, or rule-based models can improve transparency and trust in clinical settings (Plis et al., 2014)[59].For efficient dimensionality reduction, it's crucial to make careful selections. Methods like Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) serve as examples of techniques that strike a balance between data compression and preserving relevant features (Cuingnet et

al., 2011)[60]. The application of machine learning and deep learning algorithms in clinical settings for AD detection has profound implications and ethical considerations. Transitioning from research to clinical use requires rigorous validation and regulatory approval. Algorithms must meet high standards of reliability and accuracy before being integrated into clinical practice. The process can be resource-intensive (McDuff et al., 2018)[61]. The handling of sensitive medical information, especially in the context of patient privacy, is of paramount importance. Ensuring data security and compliance with privacy regulations is crucial in clinical applications (Kosik et al., 2012)[62]. Algorithms should be continuously monitored and updated. As our understanding of AD evolves and new data become available, algorithms must adapt to remain accurate (Liang et al., 2015)[63]. Leveraging AI in healthcare gives rise to apprehensions regarding the privacy and security of patient data. Therefore, institutions and researchers must prioritize data anonymization and implement robust safeguards against breaches (Cho et al., 2017)[64]. Responsibility for algorithmic errors or biases in clinical decision-making is a significant ethical consideration. Transparency in algorithmic decision-making is essential to address these concerns (Vellido et al., 2018)[65]. Patients should be informed about the use of AI in their diagnosis and treatment. Informed consent is crucial in maintaining trust between healthcare providers and patients (LeCun et al., 2015)[66]. Despite these challenges, the field of AD detection using machine learning and deep learning algorithms is advancing. Collaborative efforts should focus on diversifying datasets and enhancing data quality. This will lead to more representative datasets and improved algorithm generalizability (Zhang et al., 2018)[68]. Research should emphasize the development of interpretable AI models. This involves creating models that can provide explanations for their predictions, which is vital for clinical acceptance (Basheera et al., 2019)[69]. Regulatory bodies and healthcare organizations should establish clear ethical guidelines for the use of AI in healthcare. These guidelines should address privacy, data security, accountability, and patient consent (Wiens et al., 2019)[70].



**Fig.3.2.**Image sample and alzheimer probabilities

## CHAPTER 4

### METHODOLOGY

#### **4.1. Support Vector Machine (SVM):**

Support Vector Machines (SVMs) have practical implications in the early detection of Alzheimer's disease (AD) using MRI images. Their robustness, accuracy, and versatility make them valuable tools for clinicians, researchers, and healthcare professionals. One of the significant practical implications of SVMs in AD detection is their ability to handle binary classification effectively. SVMs excel in distinguishing AD cases from non-AD cases, providing a clear and straightforward diagnosis. This capability is vital in early intervention, enabling clinicians to identify potential AD patients at an early stage. The use of SVMs also extends to risk assessment. SVMs can assign a risk score to individuals based on their MRI data, indicating the likelihood of developing AD in the future. This risk assessment can guide healthcare professionals in implementing preventive measures and personalized treatment plans, reducing the impact of the disease on patients' lives. SVMs contribute to precision medicine in AD by aiding in treatment selection. Clinicians can use SVM-derived insights to choose the most suitable treatment options for individual patients, enhancing the effectiveness of interventions. Additionally, SVMs can assist in monitoring treatment outcomes, enabling healthcare professionals to adapt and refine treatment plans based on patients' progress. Furthermore, SVMs are practical in research and drug development. They help in the selection of appropriate study participants by identifying individuals with early-stage AD or at risk of developing the disease. This ensures that clinical trials are conducted on the most relevant patient populations, increasing the chances of developing effective AD treatments. SVMs can also enhance the efficiency of healthcare resource allocation. By accurately identifying AD cases and estimating their risk, healthcare systems can allocate resources, including medical personnel and facilities, more effectively. This ensures that AD patients receive timely and appropriate care. In conclusion, SVMs have practical implications in AD detection and management. Their ability to handle binary classification, provide risk assessments, aid in treatment selection, support research, and optimize resource allocation makes them valuable tools in the fight against AD.

## 4.2.EfficientNet:

EfficientNet, a deep learning architecture, offers several practical implications in the context of Alzheimer's disease (AD) detection using MRI images. Its efficiency and performance make it a valuable tool for researchers, clinicians, and healthcare professionals. One of the key practical implications of EfficientNet is its capability to handle large-scale MRI datasets. AD research often involves extensive datasets with numerous MRI images. EfficientNet's scalability and efficiency enable it to process these datasets effectively, reducing the computational burden on researchers and clinicians. EfficientNet's transfer learning capabilities are another practical advantage. By leveraging pre-trained models like EfficientNet-B0, researchers can save time and resources. They can fine-tune the model on their specific AD classification task, avoiding the need to train a deep network from scratch. This expedites the research process and allows for faster deployment in clinical settings. The use of data augmentation in EfficientNet further enhances its practical implications. Data augmentation techniques increase the diversity of the training dataset, making the model more robust and capable of handling variations in MRI images. This robustness is crucial in real-world clinical applications where MRI image quality can vary. EfficientNet's ability to capture complex patterns in MRI images is a practical asset. It excels at feature extraction, enabling it to identify subtle changes associated with AD. This is invaluable in early diagnosis, as EfficientNet can recognize AD-related patterns that might not be apparent to human observers. Additionally, EfficientNet's performance evaluation metrics, including accuracy and sensitivity, are crucial in assessing the model's effectiveness. Researchers and clinicians can rely on these metrics to gauge the model's performance and suitability for AD detection. EfficientNet's practical implications also extend to personalized medicine. By providing accurate AD diagnoses based on MRI data, clinicians can tailor treatment plans to individual patients. This personalized approach improves treatment outcomes and enhances patients' quality of life. In research and drug development, EfficientNet aids in patient selection for clinical trials. It identifies potential participants with early-stage AD or high risk, ensuring that trials are conducted on relevant populations. This increases the likelihood of finding effective AD treatments. In summary, EfficientNet's practical implications lie in its scalability, transfer learning capabilities, data augmentation, pattern recognition, performance evaluation, and support for personalized medicine, research, and drug development in AD detection and management.

### **4.3.Convolutional Neural Network (CNN):**

Convolutional Neural Networks (CNNs) have far-reaching practical implications in the field of Alzheimer's disease (AD) detection using MRI images. Their robust and versatile nature makes them indispensable in both research and clinical settings. One of the most significant practical implications of CNNs is their ability to automate the feature extraction process. CNNs are adept at learning and identifying relevant features from MRI images, a task that traditionally required extensive manual labor. This automation accelerates the diagnosis process, reduces human error, and ensures consistent and reliable results. In clinical practice, CNNs enable early and accurate AD diagnosis. The ability to detect subtle changes in MRI images that might elude human observers is particularly valuable. Early diagnosis empowers clinicians to intervene in the disease's progression, allowing for timely treatment and support. Moreover, CNNs have the practical advantage of handling large and diverse datasets. AD research often involves extensive datasets from various sources. CNNs can process and analyze these datasets efficiently, ensuring that research findings are based on comprehensive and representative samples. The versatility of CNNs extends to data augmentation techniques. These techniques increase the diversity of the training dataset, making the model more robust and capable of handling variations in MRI images. This is essential for real-world clinical applications where MRI image quality can vary significantly. CNNs' practical implications include personalized medicine. By providing accurate AD diagnoses based on MRI data, clinicians can tailor treatment plans to individual patients. This personalized approach improves treatment outcomes and enhances patients' quality of life. In the research and drug development domains, CNNs assist in patient selection for clinical trials. They identify potential participants with early-stage AD or a high risk of developing the disease. This ensures that clinical trials are conducted on relevant populations, increasing the likelihood of finding effective AD treatments. The practicality of CNNs also extends to their performance evaluation metrics. Metrics such as accuracy, precision, and recall provide valuable insights into the model's performance, allowing researchers and clinicians to assess its suitability for AD detection. In conclusion, CNNs offer practical implications in AD detection, including automated feature extraction, early diagnosis, dataset handling, data augmentation, personalized medicine, support for research and drug development, and performance evaluation. These attributes make CNNs indispensable tools in the battle against Alzheimer's disease.

#### **4.4.VGG (Visual Geometry Group):**

Visual Geometry Group (VGG) models, such as VGG-16 and VGG-19, have substantial practical implications in the field of Alzheimer's disease (AD) detection using MRI images. Their deep architectures and fine-tuning capabilities make them valuable assets for researchers and clinicians. One of the practical implications of VGG models lies in their ability to handle complex MRI datasets. AD research often involves extensive datasets with diverse MRI images. VGG models can efficiently process and analyze these datasets, ensuring comprehensive and accurate results. Transfer learning is a key practical advantage of VGG models. Researchers can leverage pre-trained VGG models as a starting point for AD detection tasks. Fine-tuning the model on their specific classification task avoids the need to train a deep network from scratch. This expedites research and clinical deployment. VGG models excel at feature extraction, automating a task that was traditionally performed manually. This automated feature extraction accelerates the diagnosis process, reduces human error, and ensures consistent and reliable results. Data augmentation techniques enhance the practical implications of VGG models. These techniques increase the diversity of the training dataset, making the model more robust and capable of handling variations in MRI images. This robustness is crucial for real-world clinical applications where MRI image quality can vary. VGG's performance evaluation metrics, including accuracy, precision, and recall, are indispensable in assessing the model's effectiveness. Researchers and clinicians rely on these metrics to gauge the model's performance and suitability for AD detection. Moreover, VGG models support personalized medicine. By providing accurate AD diagnoses based on MRI data, clinicians can tailor treatment plans to individual patients. This personalized approach improves treatment outcomes and enhances patients' quality of life. In research and drug development, VGG models aid in patient selection for clinical trials. They identify potential participants with early-stage AD or high risk, ensuring that trials are conducted on relevant populations. This increases the likelihood of finding effective AD treatments. In summary, VGG models offer practical implications in AD detection, including dataset handling, transfer learning, automated feature extraction, data augmentation, performance evaluation, personalized medicine, and support for research and drug development. These attributes make VGG models valuable tools in the fight against Alzheimer's disease.

#### **4.5.Gaussian Naive Bayes (GNB):**

Gaussian Naive Bayes (GNB) has practical implications in Alzheimer's disease (AD) detection using MRI images. Its probabilistic approach, simplicity, and efficiency make it a valuable tool for researchers, clinicians, and healthcare professionals. One of the practical implications of GNB is its ability to automate the classification process. GNB is adept at learning and identifying patterns and features from MRI images, a task that traditionally required extensive manual labor. This automation accelerates the diagnosis process, reduces human error, and ensures consistent and reliable results. The probabilistic nature of GNB has practical advantages in risk assessment. GNB can assign a risk score to individuals based on their MRI data, indicating the likelihood of developing AD in the future. This risk assessment can guide healthcare professionals in implementing preventive measures and personalized treatment plans, reducing the impact of the disease on patients' lives. GNB's simplicity and computational efficiency make it practical for real-world clinical applications. It can provide rapid and accurate AD diagnoses, enabling timely interventions and treatment. This is crucial for improving patients' quality of life and managing the disease effectively. Moreover, GNB's practical implications extend to personalized medicine. By providing accurate AD diagnoses based on MRI data, clinicians can tailor treatment plans to individual patients. This personalized approach improves treatment outcomes and enhances patients' quality of life. In research and drug development, GNB assists in patient selection for clinical trials. It identifies potential participants with early-stage AD or a high risk of developing the disease. This ensures that clinical trials are conducted on relevant populations, increasing the likelihood of finding effective AD treatments. The practicality of GNB also extends to its performance evaluation metrics. Metrics such as accuracy, precision, and recall provide valuable insights into the model's performance, allowing researchers and clinicians to assess its suitability for AD detection. In summary, GNB offers practical implications in AD detection, including automated classification, risk assessment, computational efficiency, personalized medicine, support for research and drug development, and performance evaluation. These attributes make GNB a valuable tool in the fight against Alzheimer's disease.

#### **4.6.XGBoost (Extreme Gradient Boosting):**

Extreme Gradient Boosting, commonly known as XGBoost, offers several practical implications in Alzheimer's disease (AD) detection using MRI images. Its ensemble learning approach, performance, and versatility make it a valuable asset for researchers, clinicians, and healthcare professionals. One of the significant practical implications of XGBoost is its ability to handle large and diverse MRI datasets. AD research often involves extensive datasets with numerous MRI images. XGBoost's scalability and efficiency enable it to process these datasets effectively, reducing the computational burden on researchers and clinicians. XGBoost's ensemble learning capabilities are a practical advantage. By combining the results of multiple decision trees, XGBoost improves the model's accuracy and robustness. Decision trees are built iteratively, with each tree correcting the errors of the previous ones. This ensemble approach often results in a highly accurate and resilient model. XGBoost has practical implications in risk assessment. It can assign a risk score to individuals based on their MRI data, indicating the likelihood of developing AD in the future. This risk assessment can guide healthcare professionals in implementing preventive measures and personalized treatment plans, reducing the impact of the disease on patients' lives. Furthermore, XGBoost's ability to capture complex patterns in MRI images is a practical asset. It excels at feature extraction, enabling it to identify subtle changes associated with AD. This is invaluable in early diagnosis, as XGBoost can recognize AD-related patterns that might not be apparent to human observers. XGBoost's performance evaluation metrics, including accuracy, precision, and recall, are crucial in assessing the model's effectiveness. Researchers and clinicians rely on these metrics to gauge the model's performance and suitability for AD detection. In research and drug development, XGBoost aids in patient selection for clinical trials. It identifies potential participants with early-stage AD or a high risk of developing the disease. This ensures that clinical trials are conducted on relevant populations, increasing the likelihood of finding effective AD treatments. In summary, XGBoost offers practical implications in AD detection, including dataset handling, ensemble learning, risk assessment, pattern recognition, performance evaluation, support for research and drug development. These attributes make XGBoost a valuable tool in the fight against Alzheimer's disease.

## CHAPTER 5

### CODING AND TESTING

#### 5.1.Alzheimer-VGG\_with\_SVM,GNB,XGBoost

```

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files
under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import skimage.io
import os
import tqdm
import glob
import tensorflow
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import tensorflow as tf
from tensorflow.keras.utils import image_dataset_from_directory
from tensorflow.data.experimental import AUTOTUNE
from tensorflow.keras import Sequential, Input, Model
from tensorflow.keras.layers import RandomRotation, RandomZoom
from tensorflow.keras.layers.experimental.preprocessing import Rescaling

```

```
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout
from tensorflow.keras import applications
from tensorflow.keras.losses import CategoricalCrossentropy
from tensorflow.keras.optimizers import Adam

from tqdm import tqdm
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
from skimage.io import imread, imshow
from skimage.transform import resize

from tensorflow.keras.models import Sequential
from tensorflow.keras.metrics import Precision, AUC, Recall
from tensorflow.keras.layers import InputLayer, BatchNormalization, Dropout, Flatten,
Dense, Activation, MaxPool2D, Conv2D
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras.applications.densenet import DenseNet169
import copy
import warnings
warnings.filterwarnings('ignore')
import tensorflow as tf
import cv2
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.utils import load_img, img_to_array
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.utils import shuffle
from sklearn.metrics import confusion_matrix
from keras.applications.vgg16 import VGG16, preprocess_input
from keras.applications.vgg19 import VGG19, preprocess_input
from tensorflow.keras.utils import image_dataset_from_directory
```

```
from sklearn.svm import SVC
import splitfolders

input_folder='/kaggle/input/alzheimer-mri-dataset/Dataset'

output_folder='/kaggle/working/Splitted'

train_ratio=0.8
validation_ratio=0.1
test_ratio=0.1
splitfolders.ratio(input_folder, output_folder, seed=42,
                    ratio=(train_ratio,
                           validation_ratio,
                           test_ratio))

from keras.preprocessing.image import ImageDataGenerator

BATCH_SIZE=16
IMG_SIZE=(128,128)
SEED=1345

train_datagen=ImageDataGenerator(rescale=1./255,
                                 shear_range=0,
                                 zoom_range=0.2)

validation_datagen=ImageDataGenerator(rescale=1./255)
test_datagen=ImageDataGenerator(rescale=1./255)

#Defining directories for train,validation,test
train_dir = '/kaggle/working/Splitted/train'
validation_dir = '/kaggle/working/Splitted/val'
test_dir = '/kaggle/working/Splitted/test'
```

```
#Defining generators for train,validation,test

train_generator=train_datagen.flow_from_directory(
    train_dir,
    target_size=(128, 128),
    shuffle=True,
    seed = SEED,
    batch_size=64,
    class_mode ='categorical',
)

validation_generator = validation_datagen.flow_from_directory(
    validation_dir,
    target_size=(128, 128),
    seed = SEED,
    shuffle=True,
    batch_size=64,
    class_mode ='categorical',)

# Define generator for test set using flow_from_directory
test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(128, 128),
    shuffle=True,
    seed = SEED,
    batch_size =64,
    class_mode ='categorical',
)

class_names=list(train_generator.class_indices.keys())
print(class_names)
['Mild_Demented', 'Moderate_Demented', 'Non_Demented', 'Very_Mild_Demented']
train_generator
plt.figure(figsize=(12,12))
```

```

for images,labels in train_generator:
#    print(images)
#    print(len(labels))
    for i in range(9):
        ax=plt.subplot(3,3,i+1)
        plt.imshow(images[i])
        print(images[i].shape)
        plt.title(class_names[np.argmax(labels[i])])
        plt.axis("off")
    break

```

Moving towards Data Augmentation

```

def data_augmentar():
    data_augmentation = Sequential()
    data_augmentation.add(RandomRotation(factor=(-0.15, 0.15)))
    data_augmentation.add(RandomZoom((-0.3, -0.1)))
    return data_augmentation

data_augmentation = data_augmentar()
assert(data_augmentation.layers[0].name.startswith('random_rotation'))
assert(data_augmentation.layers[0].factor == (-0.15, 0.15))
assert(data_augmentation.layers[1].name.startswith('random_zoom'))
assert(data_augmentation.layers[1].height_factor == (-0.3, -0.1))

```

input\_folder='/kaggle/input/alzheimer-mri-dataset/Dataset'

```

class_count=dict()

for i in class_names:
    class_count[i]=len(os.listdir(input_folder+'/'+i))

plt.figure(figsize=(10,4))
plt.bar(class_count.keys(),class_count.values())

```

```

plt.xlabel('Classes')
plt.ylabel('Counts')
plt.title('Visualization of Data Imbalance')
plt.grid(True)
plt.show()

total_samples=sum(class_count.values())

for i in range(4):
    class_weight = round(total_samples / (4 * list(class_count.values())[i]), 2)
    print(f'Weight for class \'{class_names[i]}\' : {class_weight}')
input_shape = (128,128, 3)

#Create an instance of the VGG19 model
vgg19 = VGG19(include_top=False, input_shape=input_shape,
               weights='imagenet')

train_features = vgg19.predict(train_generator, steps=len(train_generator), verbose=1)
val_features=vgg19.predict(validation_generator,len(validation_generator),verbose=1)
test_features = vgg19.predict(test_generator, steps=len(test_generator), verbose=1)
train_labels=train_generator.classes
val_labels=validation_generator.classes
test_label=test_generator.classes
trainval_features = np.concatenate((train_features, val_features))
trainval_labels = np.concatenate((train_labels, val_labels))

trainval_features.shape
X_train_2d = trainval_features.reshape(trainval_features.shape[0], -1)
X_test_2d = test_features.reshape(test_features.shape[0], -1)
X_train_2d.shape
(6400, 8192)
train_labels

```

```

indices = np.random.permutation(6400)

# Shuffle X_train and X_labels using the same indices
shuffled_X_train = X_train_2d[indices]
shuffled_X_labels = trainval_labels[indices]
import xgboost as xgb
from sklearn.model_selection import GridSearchCV
param_grid = {
    'n_estimators': [50, 100, 150],
    'max_depth': [3, 4, 5],
    'learning_rate': [0.1, 0.01]
}

# Perform grid search cross-validation
xgb_model = xgb.XGBClassifier()
grid_search = GridSearchCV(estimator=xgb_model, param_grid=param_grid, cv=3)
grid_search.fit(shuffled_X_train, shuffled_X_labels)

# Get the best parameters and best score found during grid search
best_params = grid_search.best_params_
best_score = grid_search.best_score_
print(best_params)
print(best_score)
{'C': 0.1, 'kernel': 'rbf'}
0.500097751710655

# Creatig new XGB Model with best parameters
best_xgb = XGBClassifier(**best_params)
best_xgb.fit(shuffled_X_train, shuffled_X_labels)
from sklearn.metrics import confusion_matrix, classification_report

# Make predictions on the test data
y_pred = best_xgb.predict(X_test_2d)

# Compute the confusion matrix

```

```

cm = confusion_matrix(test_label, y_pred)
print("Confusion Matrix:")
print(cm)

# Compute the classification report
report = classification_report(y_test, y_pred)
print("Classification Report:")
print(report)

```

## 5.2. Alzheimer MRI Classification Using CNN

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import seaborn as sns
import math
import os
import warnings
warnings.filterwarnings('ignore')

from sklearn.utils.class_weight import compute_class_weight
from sklearn.metrics import classification_report, confusion_matrix

import keras
from tensorflow import keras
from keras import Sequential
from keras import layers
import tensorflow as tf
from tensorflow.keras.preprocessing import image_dataset_from_directory
from tensorflow.keras import Sequential
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.layers import Dense, Dropout, Activation, BatchNormalization,

```

```

Flatten, Conv2D, MaxPooling2D
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping

plt.rcParams["figure.figsize"] = (10,6)
plt.rcParams['figure.dpi'] = 300
colors = ["#B6EE56", "#D85F9C", "#EEA756", "#56EEE8"]
try:
    if tf.test.gpu_device_name():
        physical_devices = tf.config.experimental.list_physical_devices('GPU')
        print('GPU active! -', physical_devices)
    else:
        print('GPU not active!')
except Exception as e:
    print('An error occurred while checking the GPU:', e)
class_dist = {}
def image_counter(folder_path):
    basename = os.path.basename(folder_path)
    print("\033[92m"+f"A search has been initiated within the folder named '{basename}'."+"\033[0m")
    image_extensions = ['.jpg', '.jpeg', '.png']

    for root, dirs, _ in os.walk(folder_path):
        for dir_name in dirs:
            dir_path = os.path.join(root, dir_name)
            count = 0

            for filename in os.listdir(dir_path):
                file_ext = os.path.splitext(filename)[1].lower()

                if file_ext in image_extensions:
                    count += 1

            class_dist[dir_name] = count
            print(f"There are \033[35m{count}\033[0m images in the {dir_name} folder.")

```

```

print("\033[92m'+"The search has been completed."+"\033[0m')

keys = list(class_dist.keys())
values = list(class_dist.values())
explode = (0.1,)*len(keys)

labels = [f'{key} ({value} images)' for key, value in zip(keys, values)]

plt.pie(values, explode=explode, labels=labels, autopct='%1.1f%%',
         shadow=True, startangle=90, colors=colors, textprops={'fontsize': 12, "fontweight": "bold", "color": "darkblue"}, wedgeprops=
         {'edgecolor': 'darkblue'}, , labeldistance=1.15)
plt.title("Distribution of \nAlzheimer MRI Images", size=12, fontweight="bold")

PATH = '/kaggle/input/alzheimer-mri-dataset/Dataset'

image_counter(PATH)
data = tf.keras.utils.image_dataset_from_directory(PATH,
                                                    batch_size = 32,
                                                    image_size=(128, 128),
                                                    shuffle=True,
                                                    seed=42,)

class_names = data.class_names

def sample_bringer(path, target, num_samples=5):

    class_path = os.path.join(path, target)

    image_files = [image for image in os.listdir(class_path) if image.endswith('.jpg')]

    fig, ax = plt.subplots(1, num_samples, facecolor="gray")
    fig.suptitle(f'{target} Brain MRI Samples', color="yellow", fontsize=16,
                 fontweight='bold', y=0.75)

```

```

for i in range(num_samples):
    image_path = os.path.join(class_path, image_files[i])
    img = mpimg.imread(image_path)

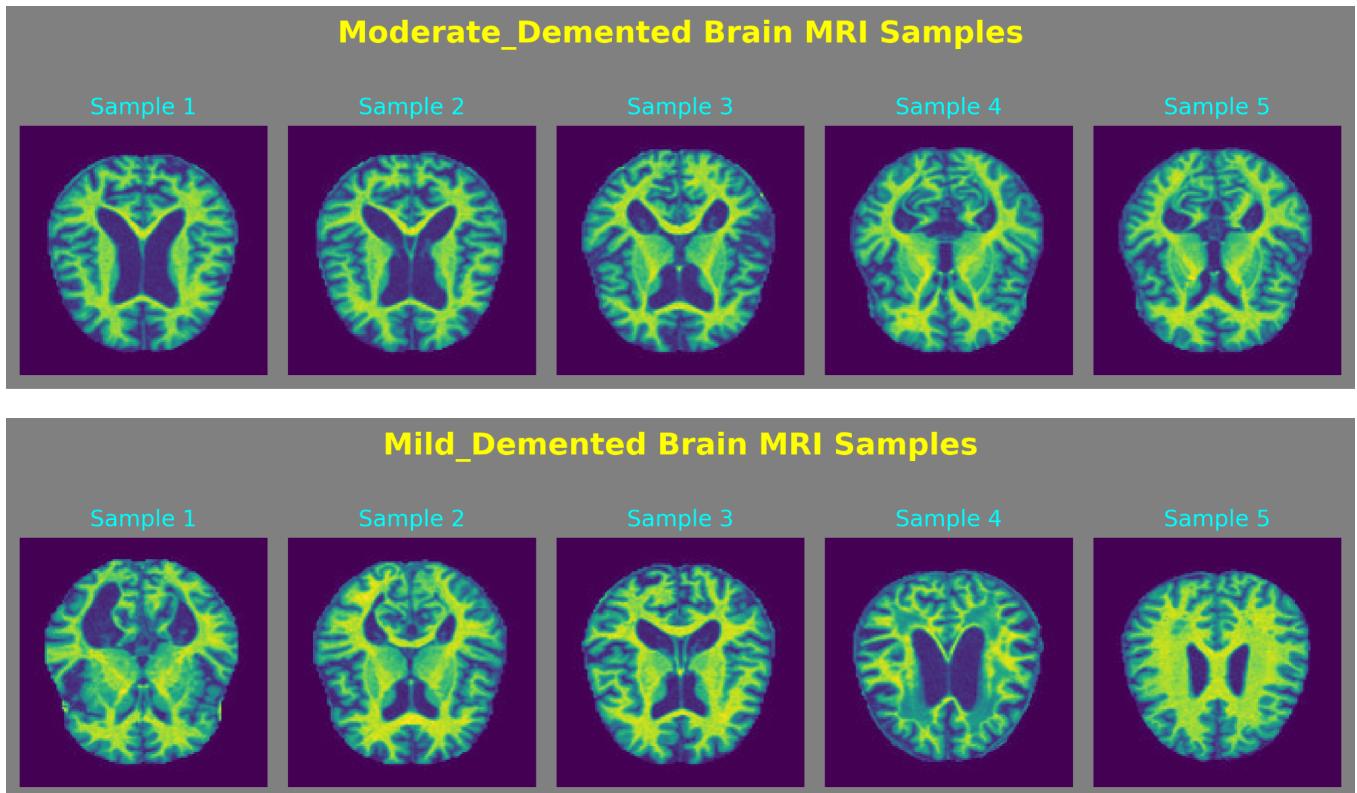
    ax[i].imshow(img)
    ax[i].axis('off')
    ax[i].set_title(f'Sample {i+1}', color="aqua")

plt.tight_layout()

```

for target in class\_names:

sample\_bring(PATH, target=target)



**Fig.5.1.Moderate and Mild Demented Brain MRI Samples**

```
alz_dict = {index: img for index, img in enumerate(data.class_names)}
```

class Process:

```

def __init__(self, data):
    self.data = data.map(lambda x, y: (x/255, y))

```

```

def create_new_batch(self):
    self.batch = self.data.as_numpy_iterator().next()
    text = "Min and max pixel values in the batch ->"
    print(text, self.batch[0].min(), "&", self.batch[0].max())

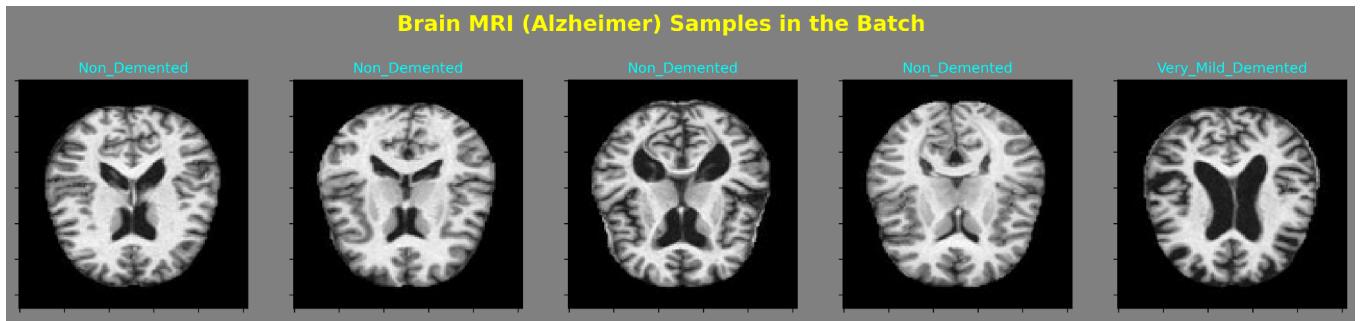
def show_batch_images(self, number_of_images=5):
    fig, ax = plt.subplots(ncols=number_of_images, figsize=(20,20), facecolor="gray")
    fig.suptitle("Brain MRI (Alzheimer) Samples in the Batch",
    color="yellow", fontsize=18, fontweight='bold', y=0.6)
    for idx, img in enumerate(self.batch[0][:number_of_images]):
        ax[idx].imshow(img)
        class_no = self.batch[1][idx]
        ax[idx].set_title(alz_dict[class_no], color="aqua")
        ax[idx].set_xticklabels([])
        ax[idx].set_yticklabels([])

def train_test_val_split(self, train_size, val_size, test_size):
    train = int(len(self.data)*train_size)
    test = int(len(self.data)*test_size)
    val = int(len(self.data)*val_size)

    train_data = self.data.take(train)
    val_data = self.data.skip(train).take(val)
    test_data = self.data.skip(train+val).take(test)

    return train_data, val_data, test_data
process = Process(data)
process.create_new_batch()
process.show_batch_images(number_of_images=5)

```



**Fig.5.2. Output of using CNN**

```
train_data, val_data, test_data= process.train_test_val_split(train_size=0.8, val_size=0.1,
test_size=0.1)
```

```
y_train = tf.concat(list(map(lambda x: x[1], train_data)), axis=0)
class_weight = compute_class_weight('balanced', classes=np.unique(y_train),
y=y_train.numpy())
class_weights = dict(zip(np.unique(y_train), class_weight))

Model Building

def build_model():
    model = Sequential()

    model.add(Conv2D(filters=16, kernel_size=(3, 3), strides=(1, 1), activation="relu",
kernel_initializer='he_normal',
           input_shape=(128, 128, 3)))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=32, kernel_size=(3, 3), strides=(1, 1), activation="relu",
kernel_initializer='he_normal'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=128, kernel_size=(3, 3), strides=(1, 1), activation="relu",
kernel_initializer='he_normal'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Flatten())
```

```

model.add(Dense(128, activation="relu", kernel_initializer='he_normal'))
model.add(Dense(64, activation="relu"))
model.add(Dense(4, activation="softmax"))

model.compile(optimizer='adam', loss="sparse_categorical_crossentropy",
metrics=['accuracy'])

model.summary()

return model

model = build_model()
def checkpoint_callback():

    checkpoint_filepath = '/tmp/checkpoint'

    model_checkpoint_callback= ModelCheckpoint(filepath=checkpoint_filepath,
                                               save_weights_only=False,
                                               frequency='epoch',
                                               monitor='val_accuracy',
                                               save_best_only=True,
                                               verbose=1)

    return model_checkpoint_callback

def early_stopping(patience):
    es_callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=patience,
                                                   verbose=1)
    return es_callback

EPOCHS = 20
checkpoint_callback = checkpoint_callback()
early_stopping = early_stopping(patience=5)

```

```

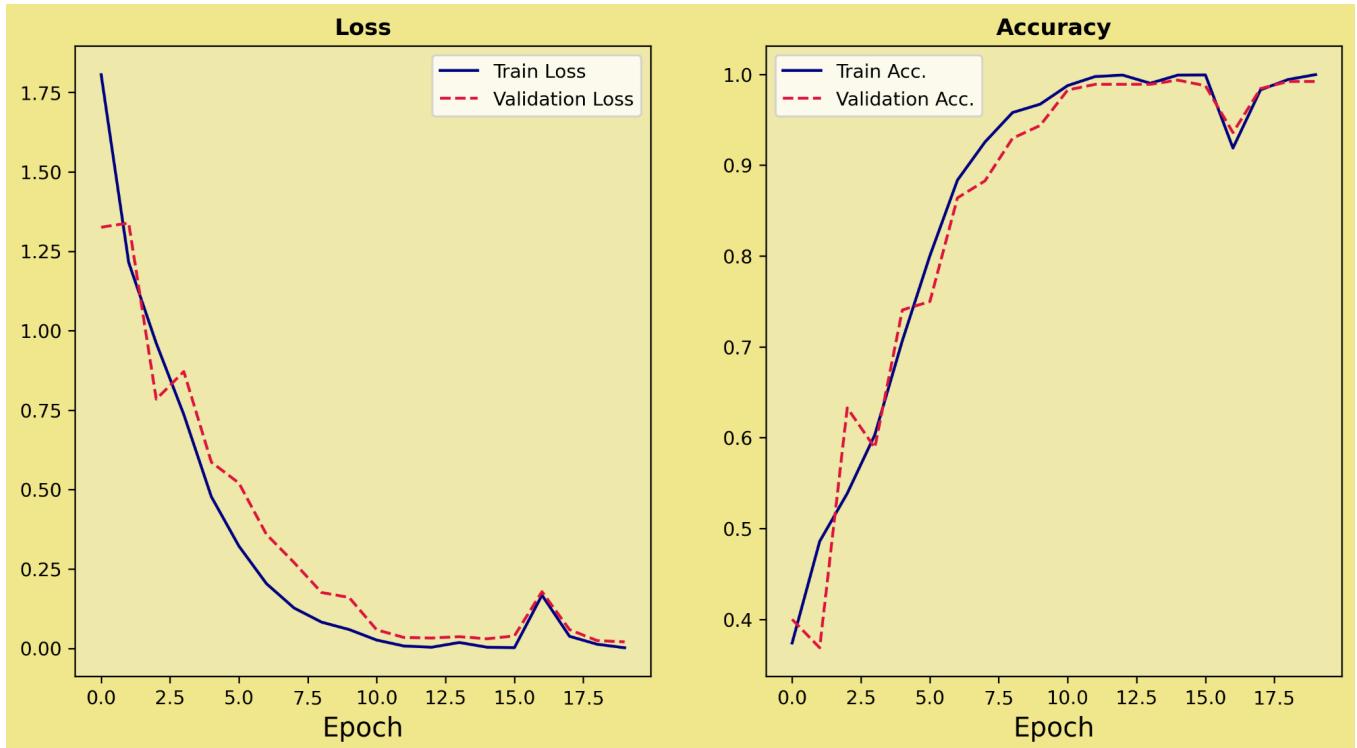
callbacks = [checkpoint_callback, early_stopping]

history = model.fit(train_data, epochs = EPOCHS, validation_data = val_data,
                     class_weight = class_weights, callbacks = callbacks)

fig, ax = plt.subplots(1, 2, figsize=(12,6), facecolor="khaki")
ax[0].set_facecolor('palegoldenrod')
ax[0].set_title('Loss', fontweight="bold")
ax[0].set_xlabel("Epoch", size=14)
ax[0].plot(history.epoch, history.history["loss"], label="Train Loss", color="navy")
ax[0].plot(history.epoch, history.history["val_loss"], label="Validation Loss",
           color="crimson", linestyle="dashed")
ax[0].legend()

ax[1].set_facecolor('palegoldenrod')
ax[1].set_title('Accuracy', fontweight="bold")
ax[1].set_xlabel("Epoch", size=14)
ax[1].plot(history.epoch, history.history["accuracy"], label="Train Acc.", color="navy")
ax[1].plot(history.epoch, history.history["val_accuracy"], label="Validation Acc.",
           color="crimson", linestyle="dashed")
ax[1].legend()

```



**Fig.5.3.Loss and Accuracy for CNN model**

```

model.evaluate(test_data)
oss: 0.0386 - accuracy: 0.9875
[0.03864416852593422, 0.987500011920929]
predictions = []
labels = []

for X, y in test_data.as_numpy_iterator():
    y_pred = model.predict(X, verbose=0)
    y_prediction = np.argmax(y_pred, axis=1)
    predictions.extend(y_prediction)
    labels.extend(y)

predictions = np.array(predictions)
labels = np.array(labels)

print(classification_report(labels, predictions, target_names=class_names))
      precision    recall  f1-score   support

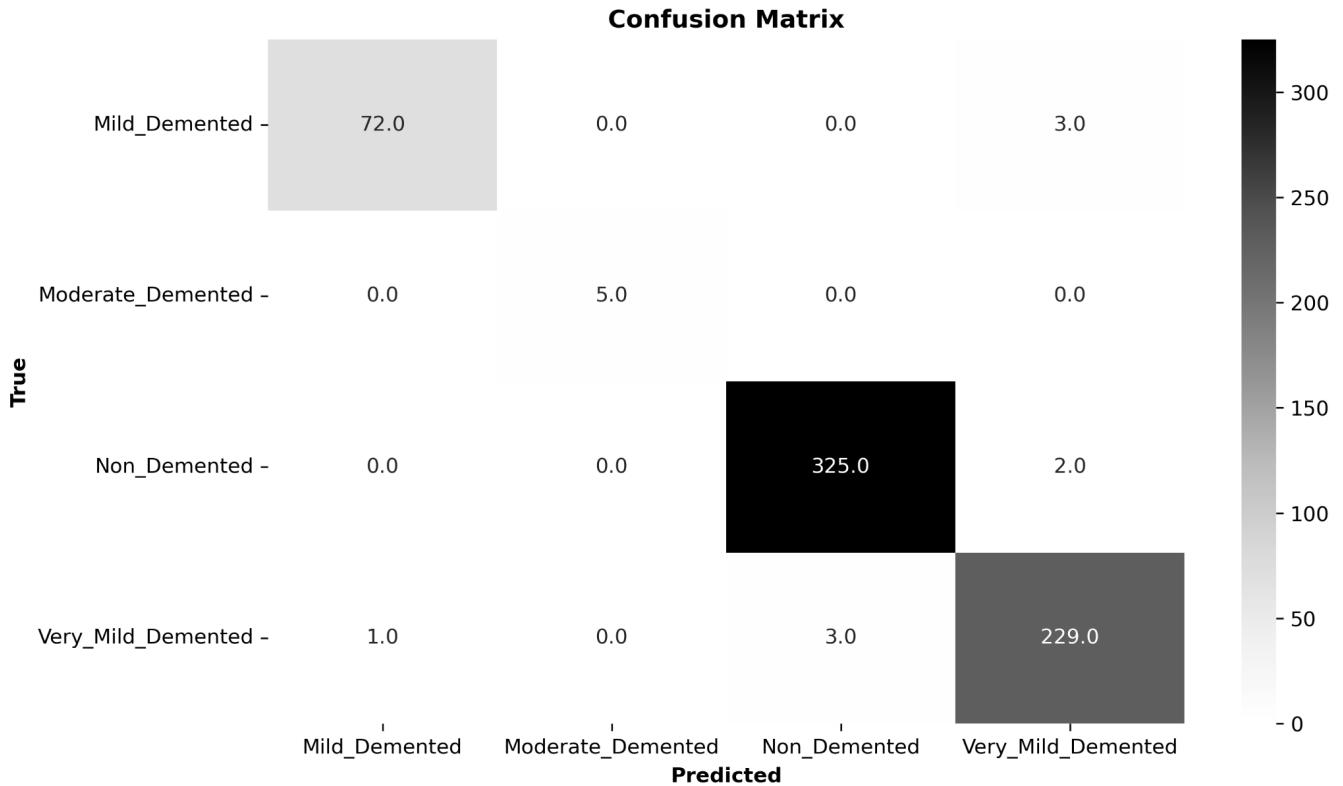
Mild_Demented      0.99     0.96     0.97      75
Moderate_Demented   1.00     1.00     1.00       5
Non_Demented        0.99     0.99     0.99     327
Very_Mild_Demented  0.98     0.98     0.98     233

accuracy           0.99     640
macro avg         0.99     0.98     0.99     640
weighted avg       0.99     0.99     0.99     640

cm = confusion_matrix(labels, predictions)
cm_df = pd.DataFrame(cm, index=class_names, columns=class_names)
cm_df
plt.figure(figsize=(10,6), dpi=300)
sns.heatmap(cm_df, annot=True, cmap="Greys", fmt=".1f")
plt.title("Confusion Matrix", fontweight="bold")
plt.xlabel("Predicted", fontweight="bold")

```

```
plt.ylabel("True", fontweight="bold")
```



**Fig. 5.4. Confusion matrix of CNN model**

```
def random_mri_prob_bringger(image_number=0):
```

```
for images, _ in test_data.skip(5).take(1):
    image = images[image_number]
    pred = model.predict(tf.expand_dims(image, 0))[0]
```

```
probs = list(tf.nn.softmax(pred).numpy())
probs_dict = dict(zip(class_dist.keys(), probs))
```

```
keys = list(probs_dict.keys())
values = list(probs_dict.values())
```

```
fig, (ax1, ax2) = plt.subplots(1, 2, facecolor='black')
plt.subplots_adjust(wspace=0.4)
ax1.imshow(image)
ax1.set_title('Brain MRI', color="yellow", fontweight="bold", fontsize=16)
```

```

edges = ['left', 'bottom', 'right', 'top']
edge_color = "greenyellow"
edge_width = 3
for edge in edges:
    ax1.spines[edge].set_linewidth(edge_width)
    ax1.spines[edge].set_edgecolor(edge_color)

plt.gca().axes.yaxis.set_ticklabels([])
plt.gca().axes.xaxis.set_ticklabels([])

wedges, labels, autopct = ax2.pie(values, labels=keys, autopct='%.1f%%',
    shadow=True, startangle=90, colors=colors, textprops={'fontsize': 8,
    "fontweight":"bold", "color":"white"}, wedgeprops=
    {'edgecolor':'black'} , labeldistance=1.15)

for autotext in autopct:
    autotext.set_color('black')

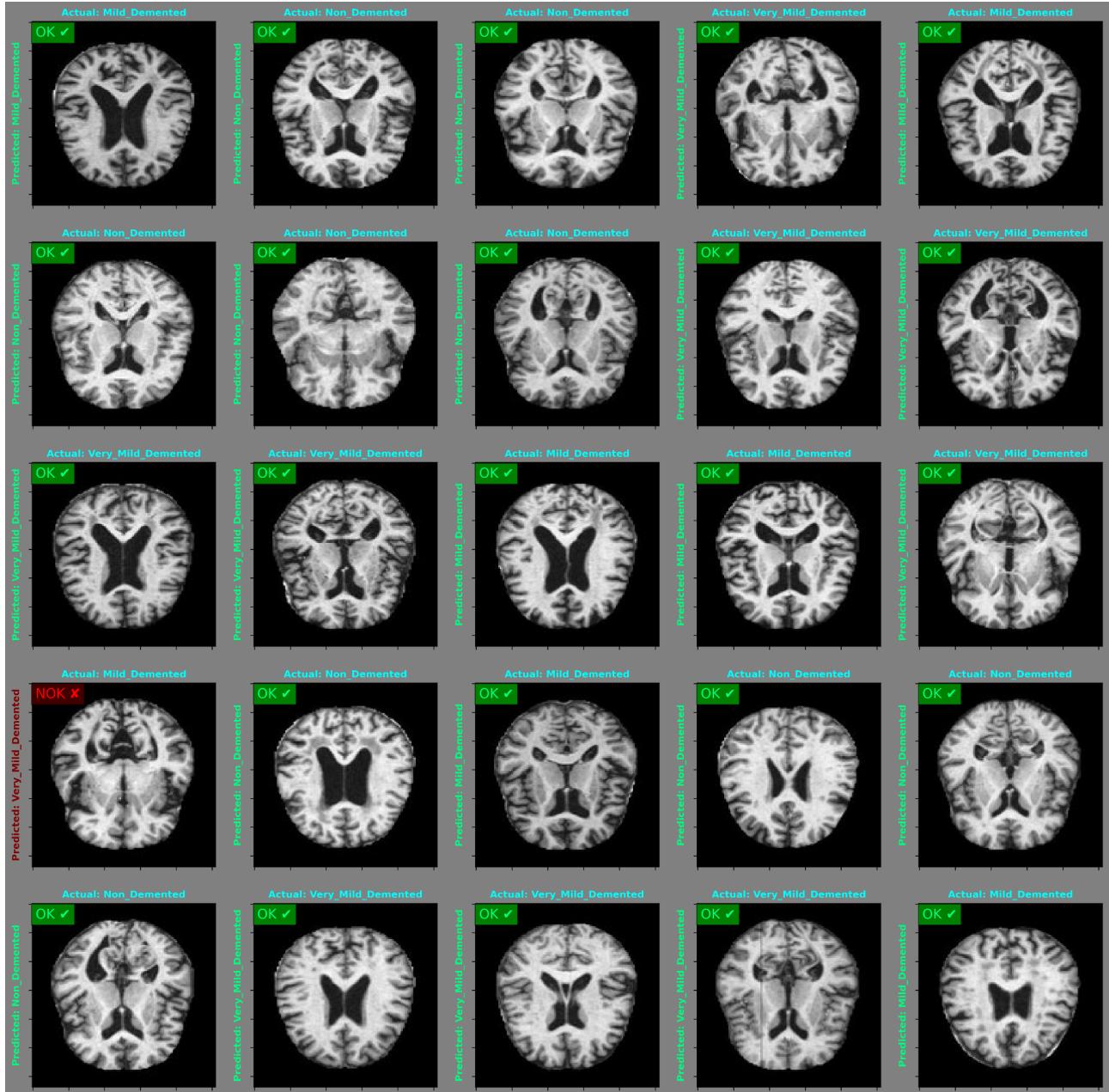
ax2.set_title('Alzheimer Probabilities', color="yellow", fontweight="bold", fontsize=16)

rand_img_no = np.random.randint(1, 32)
random_mri_prob_bringr(image_number=rand_img_no)
Comparing Predicted Classes with the Actual Classes from the Test Data
plt.figure(figsize=(20, 20), facecolor="gray")
for images, labels in test_data.take(1):
    for i in range(25):
        ax = plt.subplot(5, 5, i + 1)
        plt.imshow(images[i])
        predictions = model.predict(tf.expand_dims(images[i], 0), verbose=0)
        score = tf.nn.softmax(predictions[0])
        if(class_names[labels[i]]==class_names[np.argmax(score)]):
            plt.title("Actual: "+class_names[labels[i]], color="aqua", fontweight="bold",
            fontsize=10)

```

```
plt.ylabel("Predicted: "+class_names[np.argmax(score)], color="springgreen",
fontweight="bold", fontsize=10)
ok_text = plt.text(2, 10, "OK \u2714", color="springgreen", fontsize=14)
ok_text.set_bbox(dict(facecolor='lime', alpha=0.5))

else:
    plt.title("Actual: "+class_names[labels[i]], color="aqua", fontweight="bold",
    fontsize=10)
    plt.ylabel("Predicted: "+class_names[np.argmax(score)], color="maroon",
    fontweight="bold", fontsize=10)
    nok_text = plt.text(2, 10, "NOK \u2718", color="red", fontsize=14)
    nok_text.set_bbox(dict(facecolor='maroon', alpha=0.5))
    plt.gca().axes.yaxis.set_ticklabels([])
    plt.gca().axes.xaxis.set_ticklabels([])
```



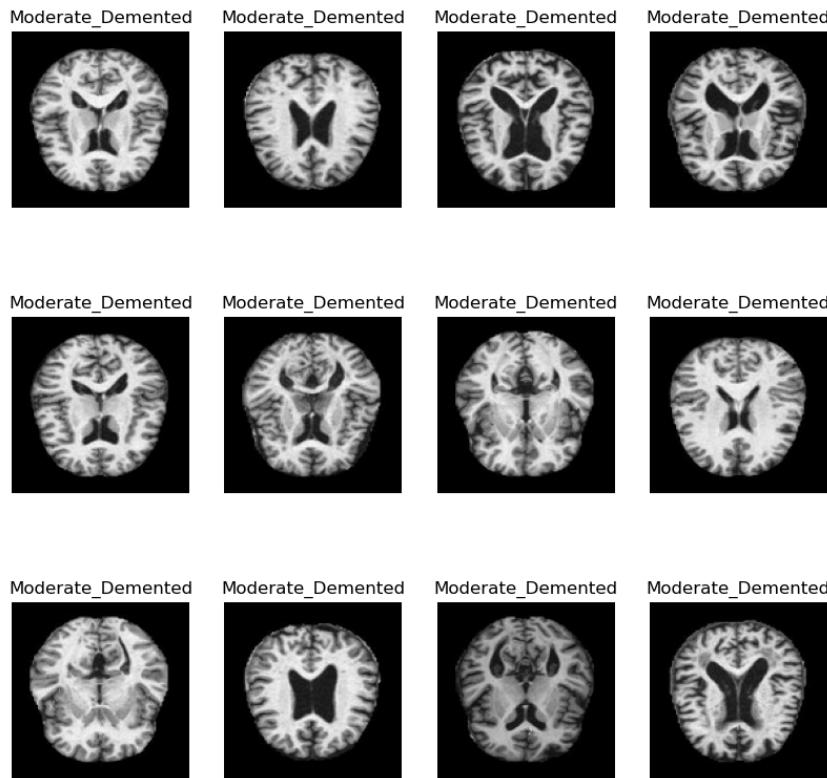
**Fig.5.5.Testing images using CNN**

### 5.3.Alzheimer - EfficientNet

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import tensorflow as tf
```



```
        image_size=IMG_SIZE,  
        shuffle=True,  
        validation_split=0.2,  
        subset='validation',  
        seed=SEED)  
  
validation_batches = len(full_validation_dataset)  
print(f'Total number of full_validation_dataset batches : {validation_batches}')  
  
validation_dataset = full_validation_dataset.take(validation_batches // 2)  
test_dataset = full_validation_dataset.skip(validation_batches // 2)  
  
print(f'Number of batches in validation dataset : {len(validation_dataset)}')  
print(f'Number of batches in test dataset : {len(test_dataset)}')  
class_names = train_dataset.class_names  
class_names  
plt.figure(figsize=(10,10))  
for images, labels in train_dataset.take(1):  
    for i in range(12):  
        ax = plt.subplot(3, 4, i+1)  
        plt.imshow(images[i].numpy().astype('uint8'))  
        plt.title(class_names[int(max(labels[i]))])  
        plt.axis('off')
```



**Fig.5.6.Sample images on EfficientNet**

```
train_dataset = train_dataset.prefetch(buffer_size=AUTOTUNE)
```

Data Augmentation

```
def data_augmentar():
```

"""This function applies two data augmentation techniques.

First, augmentation with RandomRotation.

Second, augmentation with RandomZoom

"""

```
data_augmentation = Sequential()
```

```
data_augmentation.add(RandomRotation(factor=(-0.15, 0.15)))
```

```
data_augmentation.add(RandomZoom((-0.3, -0.1)))
```

```
return data_augmentation
```

```
data_augmentation = data_augmentar()
```

```
assert(data_augmentation.layers[0].name.startswith('random_rotation'))
```

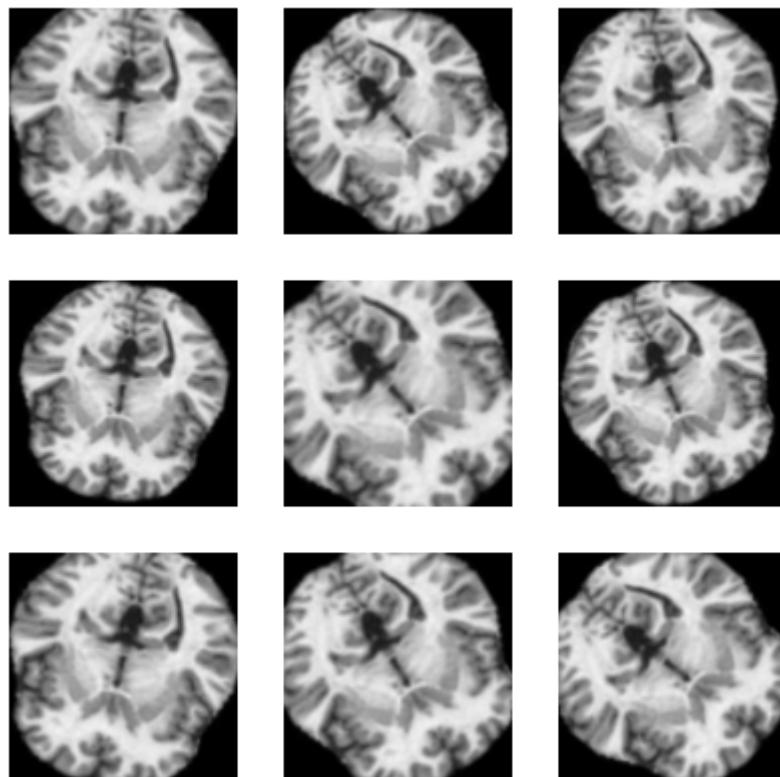
```
assert(data_augmentation.layers[0].factor == (-0.15, 0.15))
```

```
assert(data_augmentation.layers[1].name.startswith('random_zoom'))
```

```
assert(data_augmentation.layers[1].height_factor == (-0.3, -0.1))
```

```
## applying data augmentation with a sample image.
```

```
plt.figure(figsize=(5, 5))
for images, labels in train_dataset.take(1):
    image = images[0]
    for i in range(9):
        ax = plt.subplot(3, 3, i+1)
        augmented_image = data_augmentation(image)
        plt.imshow(augmented_image.numpy().astype('uint8'), cmap='gray')
        plt.axis('off')
```



**Fig.5.7.Rotated images of sample input**

### Solving Imbalanced Data Problem

```
## Calculate number of samples in each class.
```

```
class_counts = dict()
for folder in class_names:
    class_counts[folder] = len(os.listdir(directory+'/'+folder))
```

```

class_counts
plt.figure(figsize=(12,5))
plt.bar(class_counts.keys(), class_counts.values())
plt.xlabel('Classes', fontdict={'fontsize':15})
plt.ylabel('Counts', fontdict={'fontsize':15})
plt.title('Visualize Data Imbalanced', fontdict={'fontsize':17,
                                                'fontweight':'bold'})
plt.grid(True)
plt.show()
## Calculate class weights.

total = sum(class_counts.values())
number_of_classes = len(class_counts)

class_weights = dict()
for i in range(number_of_classes):
    class_weights[i] = round(total / (number_of_classes * list(class_counts.values())[i]), 2)
    print(f'Weight for class \'{class_names[i]}\' : {class_weights[i]}')

EfficientNetB0
preprocess_input = applications.efficientnet.preprocess_input
model = applications.EfficientNetB0(include_top=False)
len(model.layers)
def alzheimer_classifier(image_shape=IMG_SIZE,
data_augmentation=data_augmentar()):
    """This function creates a classifier for Alzheimer disease MRI images.


```

Arguments:

- image\_shape-> the size of the image in the form (height, width).
- data\_augmentation-> the data augmentation object to apply on the training data.

Returns:

- model-> the created classifier.

"""

```
IMG_SHAPE = image_shape + (3,)
```

```
base_model.trainable = True
```

```
for layer in base_model.layers[0:218]:
```

```
layer.trainable = False
```

```
inputs = Input(shape=IMG_SHAPE)
```

```
x = data_augmentation(inputs)
```

```
x = preprocess_input(inputs)
```

```
x = base_model(x)
```

```
x = GlobalAveragePooling2D()(x)
```

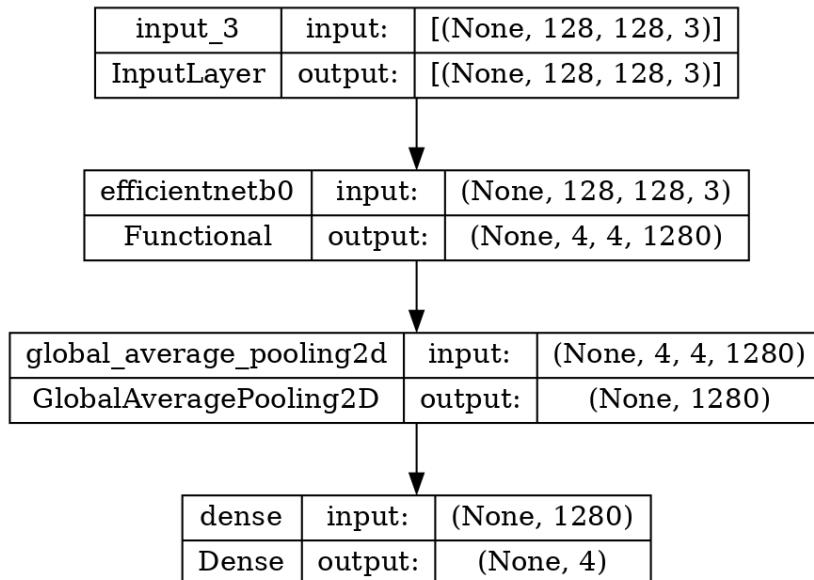
```
outputs = Dense(units=4, activation = "softmax")(x)
```

```
model = Model(inputs, outputs)
```

## return model

```
alzheimer_model = alzheimer_classifier(IMG_SIZE, data_augmentation)
```

```
alzheimer_model.summary()
```



$lr = 0.001$

```
alzheimer model.compile(loss=CategoricalCrossentropy(),
```

optimizer=Adam(learning\_rate=lr),

```

metrics=['accuracy',
         tf.keras.metrics.AUC(),
         tf.keras.metrics.Precision(),
         tf.keras.metrics.Recall(),])

# filepath = './effnet_best_weights.hdf5'

# Stop the model when accuracy is not improving
earlystopping = EarlyStopping(monitor = 'val_accuracy',
                               mode = 'max' ,
                               patience = 15,
                               verbose = 1)

# Finding point where accuracy differs more
# checkpoint  = ModelCheckpoint(filepath,
#                               monitor = 'val_accuracy',
#                               mode='max',
#                               save_best_only=True,
#                               verbose = 1)

callback_list = [earlystopping]
history = alzheimer_model.fit(train_dataset,
                               validation_data = validation_dataset,
                               epochs = 50,
                               class_weight=class_weights, callbacks = callback_list,
                               use_multiprocessing=True)

result = alzheimer_model.evaluate(test_dataset)
train_loss = result[0]
train_accuracy = result[1]
train_AUC = result[2]
train_pre = result[3]
train_rec = result[4]

print(f'Test Loss = {train_loss}')
print(f'Test Accuracy = {train_accuracy}')
print(f'Test AUC = {train_AUC}')
print(f'Test Precision = {train_pre}')

```

```
print(f'Test Recall = {train_rec}')
```

```
pd.DataFrame(history.history).plot(figsize=(10,5))
```

```
plt.grid(True)
```

```
plt.gca().set_ylimits(0,1)
```

```
plt.show()
```

```
# loss
```

```
plt.plot(history.history['loss'])
```

```
plt.plot(history.history['val_loss'])
```

```
plt.title('model loss')
```

```
plt.ylabel('loss')
```

```
plt.xlabel('epoch')
```

```
plt.legend(['train', 'val'], loc='upper left')
```

```
plt.show()
```

```
#Accuracy
```

```
plt.plot(history.history['accuracy'])
```

```
plt.plot(history.history['val_accuracy'])
```

```
plt.title('model accuracy - ' + str(format(result[1], "0.2f")))
```

```
plt.ylabel('accuracy')
```

```
plt.xlabel('epoch')
```

```
plt.legend(['train', 'val'], loc='upper left')
```

```
plt.show()
```

```
#Precision
```

```
plt.plot(history.history['precision'])
```

```
plt.plot(history.history['val_precision'])
```

```
plt.title('model precision - ' + str(format(result[3], "0.2f")))
```

```
plt.ylabel('precision')
```

```
plt.xlabel('epoch')
```

```
plt.legend(['train', 'val'], loc='upper left')
```

```
plt.show()
```

```
#Recall
```

```
plt.plot(history.history['recall'])
```

```

plt.plot(history.history['val_recall'])
plt.title('model recall - ' + str(format(result[4], "0.2f")))
plt.ylabel('recall')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()

# AUC
plt.plot(history.history['auc'])
plt.plot(history.history['val_auc'])
plt.title('model auc - ' + str(format(result[2], "0.2f")))
plt.ylabel('auc')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()

from itertools import product
y_pred = [] # store predicted labels
y_true = [] # store true labels

# iterate over the dataset
for image_batch, label_batch in test_dataset: # use dataset.unbatch() with repeat
    # append true labels
    y_true.append(label_batch)
    # compute predictions
    preds = alzheimer_model.predict(image_batch)
    # append predicted labels
    y_pred.append(np.argmax(preds, axis = 1))

    # convert the true and predicted labels into tensors
    correct_labels = tf.concat([item for item in y_true], axis = 0)
    predicted_labels = tf.concat([item for item in y_pred], axis = 0)
    correct_labels = np.argmax(correct_labels, axis=1)
    confusion_mtx = confusion_matrix(correct_labels, predicted_labels)

```

```

# y_pred = alzheimer_model.predict(test_dataset)

# predicted_categories = tf.argmax(y_pred, axis=1)

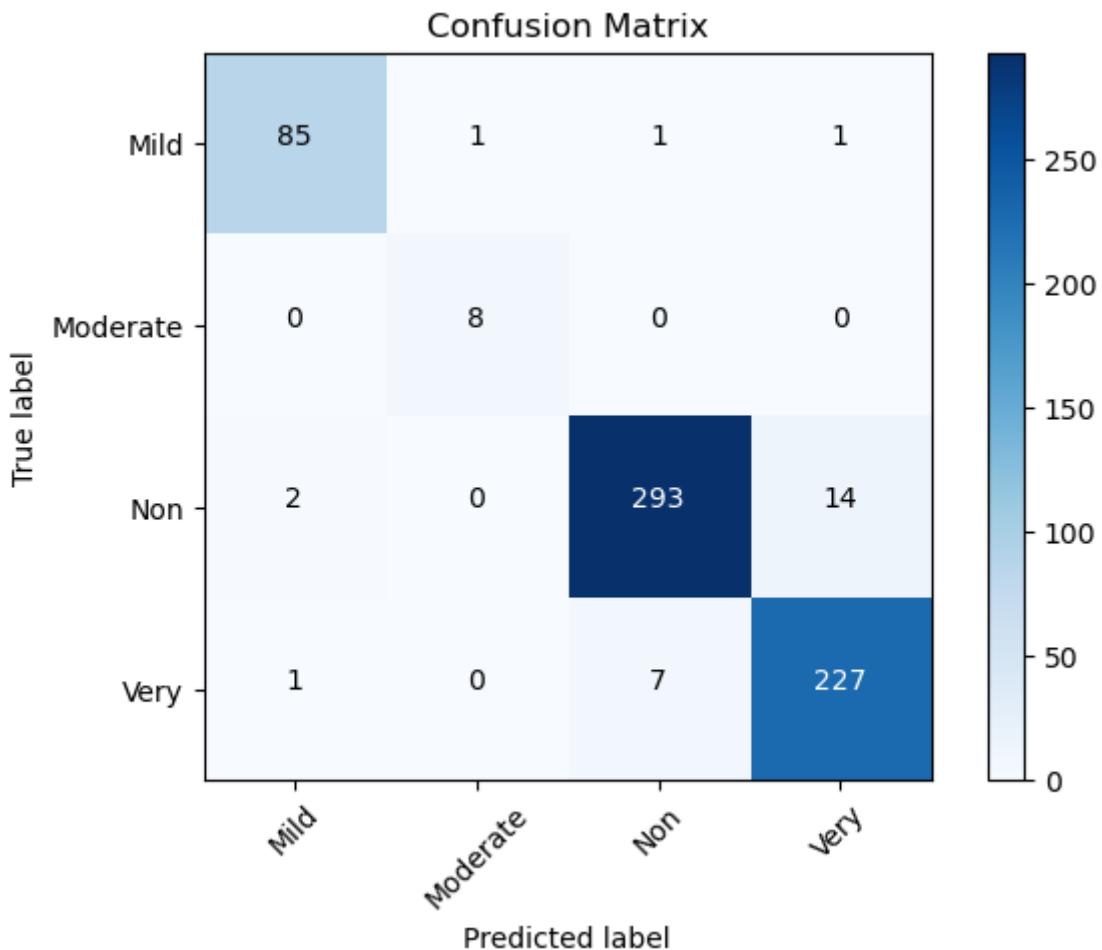
# true_categories = tf.concat([y for x, y in test_dataset], axis=0)

# confusion_mtx = confusion_matrix(predicted_categories, true_categories)

# Define the class labels
class_names = ['Mild', 'Moderate', 'Non', 'Very']

# Plot the confusion matrix
plt.imshow(confusion_mtx, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.colorbar()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names, rotation=45)
plt.yticks(tick_marks, class_names)
fmt = 'd'
thresh = confusion_mtx.max() / 2.
for i, j in product(range(confusion_mtx.shape[0]), range(confusion_mtx.shape[1])):
    plt.text(j, i, format(confusion_mtx[i, j], fmt),
             horizontalalignment="center",
             color="white" if confusion_mtx[i, j] > thresh else "black")
plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.show()

```



**Fig.5.8. Confusion matrix of EfficientNet**

#### 5.4.CNN with early stoping model

```
# Import TensorFlow and Keras modules
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import (
    BatchNormalization, Concatenate, Conv2D, Dense, Dropout, Flatten,
    GaussianNoise, GlobalAveragePooling2D, Input, MaxPooling2D, Rescaling,
    Resizing, SeparableConv2D)
import tensorflow.keras.layers as layers
from keras.callbacks import EarlyStopping, ReduceLROnPlateau

# Import data processing and visualization modules
import numpy as np
```

```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# Import scikit-learn modules
from sklearn.metrics import confusion_matrix

# Import miscellaneous modules
import random
import time
GPU
gpus = tf.config.experimental.list_physical_devices('GPU')
for gpu in gpus:
    tf.config.experimental.set_memory_growth(gpu, True)
len(gpus)
1

Training constants
batch_size = 32
img_height = 220
img_width = 220
seed = 42

Load data
train_data = tf.keras.utils.image_dataset_from_directory(
    "/kaggle/input/alzheimer-mri-dataset/Dataset",
    subset = 'training',
    validation_split = 0.2,
    seed=seed,
    image_size=(img_height, img_width),
    batch_size=batch_size
)
val_data = tf.keras.utils.image_dataset_from_directory(
    "/kaggle/input/alzheimer-mri-dataset/Dataset",
    subset = 'validation',
    validation_split = 0.2,
```

```

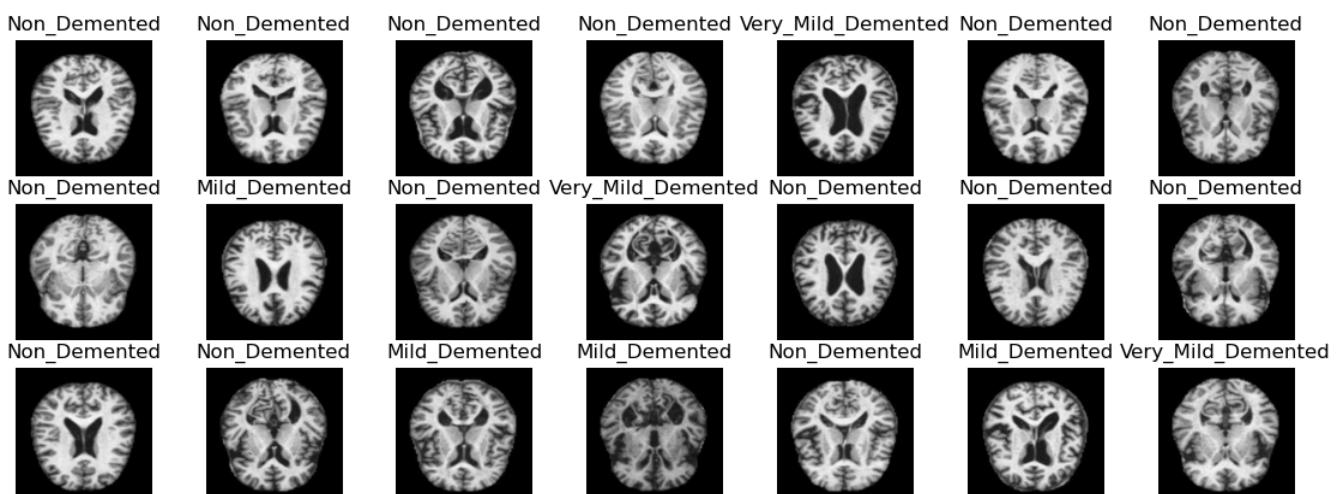
seed=seed,
image_size=(img_height, img_width),
batch_size=batch_size
)
class_names = train_data.class_names
num_classes = len(class_names)

print(f'{num_classes} classes: {class_names}')

Prepare dataset for training and evaluation
AUTOTUNE = tf.data.AUTOTUNEAUTOTUNE = tf.data.AUTOTUNE
train_data = train_data.cache().prefetch(buffer_size=AUTOTUNE)
val_data = val_data.cache().prefetch(buffer_size=AUTOTUNE)

Plotting example images
def plot_images(n_rows=3, n_cols=7, data=train_data):
    plt.figure(figsize=(n_cols*2, int(n_rows*1.8)))
    for images, labels in data.take(int((n_rows * n_cols) / batch_size) + 1): # "take" takes
        random batch
    for i in range(n_rows*n_cols):
        ax = plt.subplot(n_rows, n_cols, i + 1)
        plt.imshow(images[i].numpy().astype("uint16"))
        plt.title(class_names[labels[i]])
        plt.axis("off")
    plot_images()

```



**Fig.5.9.Sample input images for CNN with early stopping**

Get image and label batch shape

```
for image_batch, labels_batch in train_data:
```

```
    image_shape = image_batch.shape
```

```
    print(f'Image batch shape: {image_shape}')
```

```
    print(f'Label batch shape: {labels_batch.shape}')
```

```
    break
```

Get image and label batch shape

```
for image_batch, labels_batch in train_data:
```

```
    image_shape = image_batch.shape
```

```
    print(f'Image batch shape: {image_shape}')
```

```
    print(f'Label batch shape: {labels_batch.shape}')
```

```
    break
```

Callbacks

Early stopping

```
early_stopping= EarlyStopping(monitor='val_loss',
patience=4,start_from_epoch=3,restore_best_weights=True)
```

ReduceLROnPlateau

```
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=2, min_lr=1e-7)
```

Training

```
epochs = 40
```

```
history = model.fit(
```

```
    train_data,
```

```
    epochs = epochs,
```

```
    validation_data = val_data,
```

```
    batch_size = batch_size,
```

```
    callbacks = [early_stopping, reduce_lr],
```

```
    verbose = 1
```

```
)
```

Model training and validation metrics

```
# Extract metrics from history object
```

```
acc = history.history['accuracy']
```

```
val_acc = history.history['val_accuracy']
```

```
loss = history.history['loss']
```

```
val_loss = history.history['val_loss']

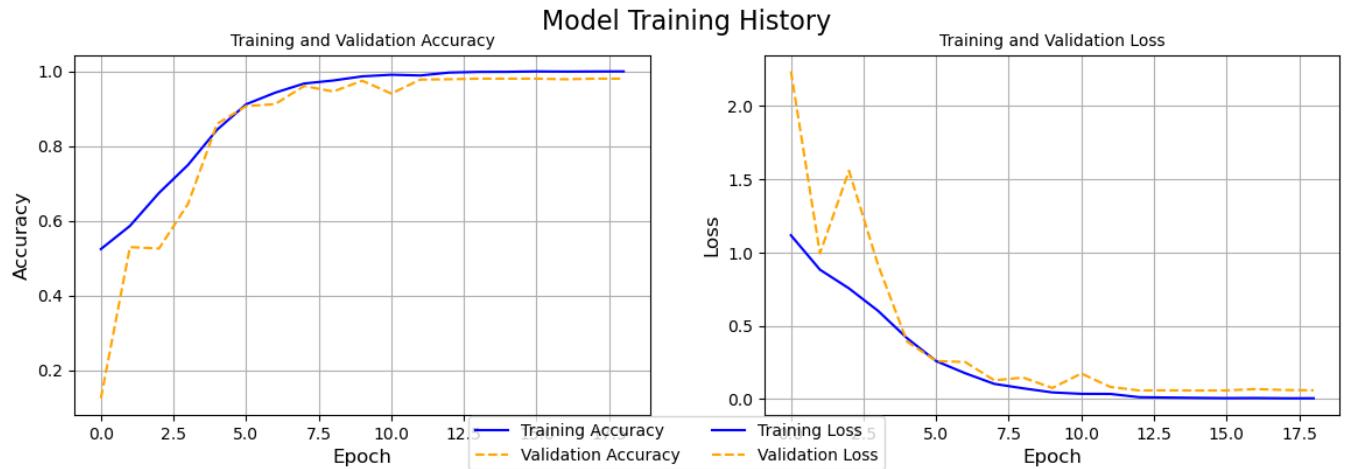
# Create figure with two subplots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 4))

# Plot accuracy metrics
ax1.plot(acc, label='Training Accuracy', color='blue', linestyle='solid')
ax1.plot(val_acc, label='Validation Accuracy', color='orange', linestyle='dashed')
ax1.set_xlabel('Epoch', fontsize=12)
ax1.set_ylabel('Accuracy', fontsize=12)
ax1.set_title('Training and Validation Accuracy', fontsize=10)
ax1.grid()

# Plot loss metrics
ax2.plot(loss, label='Training Loss', color='blue', linestyle='solid')
ax2.plot(val_loss, label='Validation Loss', color='orange', linestyle='dashed')
ax2.set_xlabel('Epoch', fontsize=12)
ax2.set_ylabel('Loss', fontsize=12)
ax2.set_title('Training and Validation Loss', fontsize=10)
ax2.grid()

# Add legend and title to figure
fig.legend(loc='lower center', ncol=2, fontsize=10)
fig.suptitle('Model Training History', fontsize=16)

# Show the plot
plt.show()
```



**Fig.5.10.Loss and accuracy for CNN with early stopping**

Model evaluation on validation dataset

```
# Evaluate the model on the validation dataset
```

```
loss, accuracy = model.evaluate(val_data)
```

```
# Calculate the number of misclassified images
```

```
num_misclassified = int((1 - accuracy) * len(val_data) * batch_size)
```

```
# Print the results
```

```
print(f"Validation loss: {loss:.4f}")
```

```
print(f"Validation accuracy: {accuracy:.4f}")
```

```
print(f"Number of misclassified images: {num_misclassified} of {len(val_data)} *  
batch_size")
```

Unpacking batches to create confusion matrix

```
# Initialize empty lists to store images and labels
```

```
val_images = []
```

```
val_labels = []
```

```
# Iterate through the test dataset and append each batch to a list
```

```
for batch in val_data.as_numpy_iterator():
```

```
    val_images.append(batch[0])
```

```
    val_labels.append(batch[1])
```

```
# Concatenate the batches into a single array for both images and labels
```

```

val_images = np.concatenate(val_images, axis=0)
val_labels = np.concatenate(val_labels, axis=0)
y_pred = np.array(model.predict(val_images))
y_true = np.array(val_labels)

Getting most probable label for an image from test dataset
# Use the trained model to predict the labels for the test images
y_pred = model.predict(val_images)

# Convert the predicted probabilities to class labels
y_pred = tf.argmax(y_pred, axis=1).numpy()

Creating a list with missclassified labels
# Get the indices of the incorrectly classified images
incorrect_indices = np.nonzero(y_pred != y_true)[0]

# Get the incorrectly classified images and labels
false_class = list(zip(val_images[incorrect_indices], y_pred[incorrect_indices],
y_true[incorrect_indices]))

# Get the correctly classified images and labels
correct_indices = np.nonzero(y_pred == y_true)[0]
true_class = list(zip(val_images[correct_indices], y_pred[correct_indices],
y_true[correct_indices]))

Confussion Matrix
# Compute the confusion matrix for the predicted labels and true labels
cm = confusion_matrix(y_true, y_pred, normalize='true')

# Create a heatmap visualization of the confusion matrix
fig, ax = plt.subplots(figsize=(8, 6))
im = ax.imshow(cm, cmap='Blues')

# Add colorbar
cbar = ax.figure.colorbar(im, ax=ax)

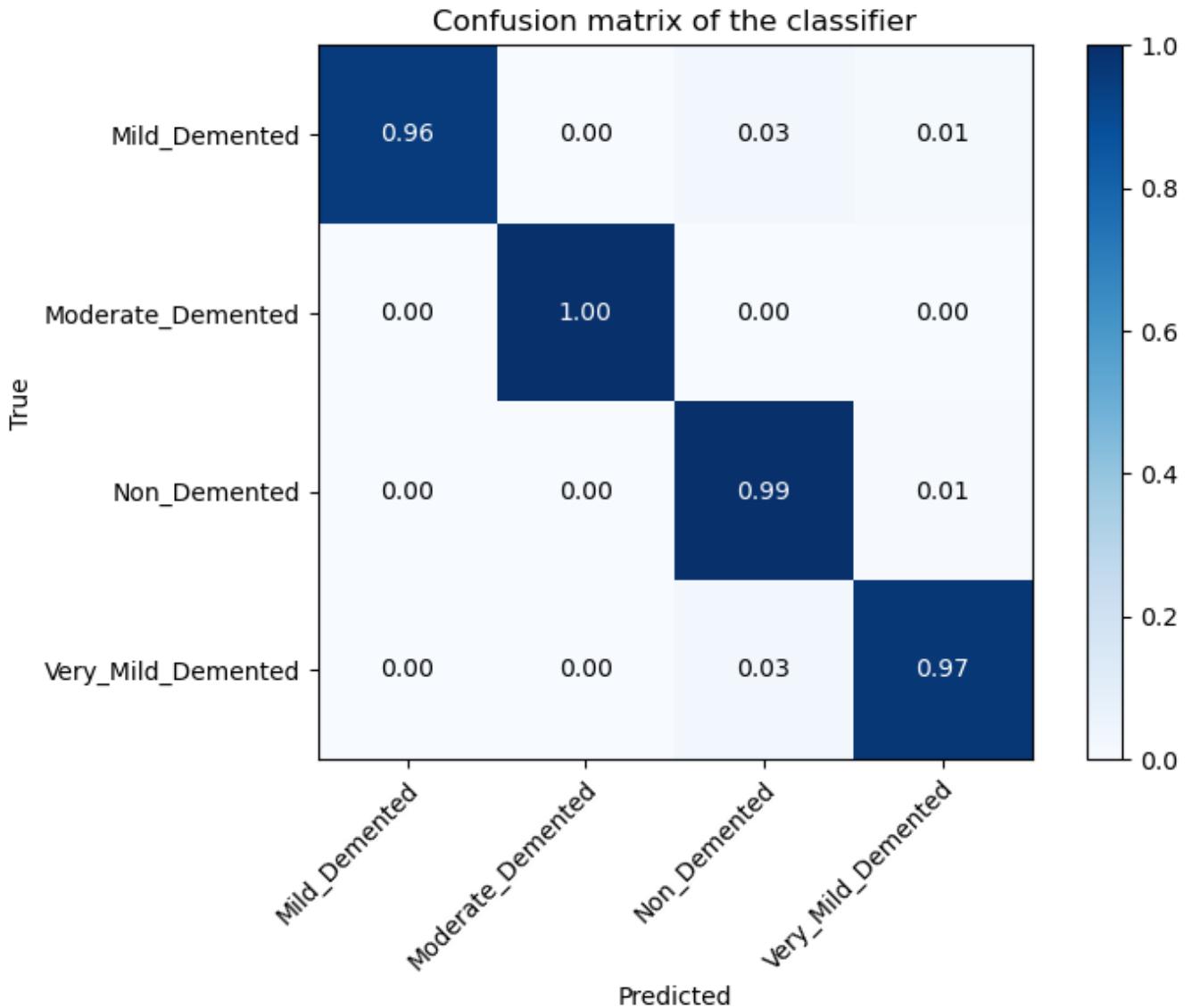
```

```
# Set tick labels
ax.set_xticks(np.arange(len(class_names)))
ax.set_yticks(np.arange(len(class_names)))
ax.set_xticklabels(class_names)
ax.set_yticklabels(class_names)

# Rotate the tick labels and set axis labels
plt.setp(ax.get_xticklabels(), rotation=45, ha="right", rotation_mode="anchor")
ax.set_xlabel('Predicted')
ax.set_ylabel('True')

# Loop over data to create annotations
for i in range(len(class_names)):
    for j in range(len(class_names)):
        text = ax.text(j, i, format(cm[i, j], '.2f'),
                       ha="center", va="center", color="white" if cm[i, j] > 0.5 else "black")

# Set title and show the plot
ax.set_title("Confusion matrix of the classifier")
fig.tight_layout()
plt.show()
```

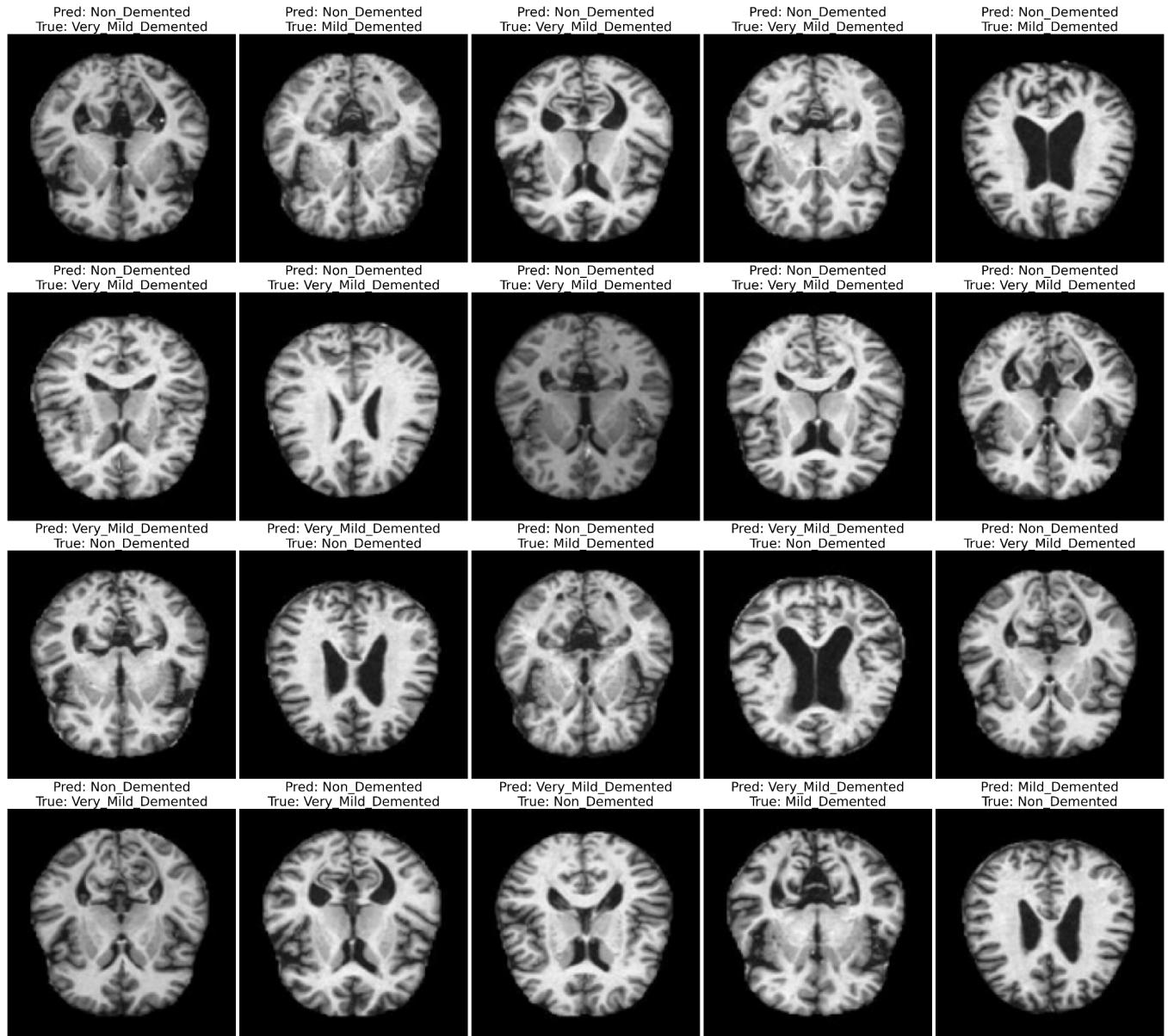


**Fig.5.11. Confusion matrix for CNN with early stopping**

Plotting errors

```
def plot_predictions(labels, cols=5):
    number_of_misscl = round((1 - accuracy) * len(val_data) * batch_size)
    random.shuffle(labels)
    rows = int(number_of_misscl/cols)-1
    fig = plt.figure(figsize=(cols*10, 45))
    for i in range(1, cols*rows +1):
        fig.add_subplot(rows, cols, i)
        plt.title(f'Pred: {class_names[labels[i][1]]}\nTrue: {class_names[labels[i][2]]}', fontsize=40)
        plt.axis("off")
        plt.imshow(labels[i][0].astype("uint16"))
```

`plt.tight_layout()`



**Fig.5.12. Model testing of CNN with early stopping**

## CHAPTER 6

### RESULTS

In this section, we will explore the potential results of our project and their implications.

#### 6.1. Support Vector Machine (SVM)[49%]:

Support Vector Machines (SVMs) have practical implications in the early detection of Alzheimer's disease (AD) using MRI images. Their robustness, accuracy, and versatility make them valuable tools for clinicians, researchers, and healthcare professionals. One of the significant practical implications of SVMs in AD detection is their ability to handle binary classification effectively. SVMs excel in distinguishing AD cases from non-AD cases, providing a clear and straightforward diagnosis. This capability is vital in early intervention, enabling clinicians to identify potential AD patients at an early stage. The use of SVMs also extends to risk assessment. SVMs can assign a risk score to individuals based on their MRI data, indicating the likelihood of developing AD in the future. This risk assessment can guide healthcare professionals in implementing preventive measures and personalized treatment plans, reducing the impact of the disease on patients' lives. SVMs contribute to precision medicine in AD by aiding in treatment selection. Clinicians can use SVM-derived insights to choose the most suitable treatment options for individual patients, enhancing the effectiveness of interventions. Additionally, SVMs can assist in monitoring treatment outcomes, enabling healthcare professionals to adapt and refine treatment plans based on patients' progress. Furthermore, SVMs are practical in research and drug development. They help in the selection of appropriate study participants by identifying individuals with early-stage AD or at risk of developing the disease. This ensures that clinical trials are conducted on the most relevant patient populations, increasing the chances of developing effective AD treatments. SVMs can also enhance the efficiency of healthcare resource allocation. By accurately identifying AD cases and estimating their risk, healthcare systems can allocate resources, including medical personnel and facilities, more effectively. This ensures that AD patients receive timely and appropriate care. In conclusion, SVMs have practical implications in AD detection and management. Their ability to handle binary classification, provide risk assessments, aid in treatment selection, support research, and optimize resource allocation makes them valuable tools in the fight against AD.

## 6.2.EfficientNet[99.7%]:

EfficientNet, a deep learning architecture, offers several practical implications in the context of Alzheimer's disease (AD) detection using MRI images. Its efficiency and performance make it a valuable tool for researchers, clinicians, and healthcare professionals. One of the key practical implications of EfficientNet is its capability to handle large-scale MRI datasets. AD research often involves extensive datasets with numerous MRI images. EfficientNet's scalability and efficiency enable it to process these datasets effectively, reducing the computational burden on researchers and clinicians. EfficientNet's transfer learning capabilities are another practical advantage. By leveraging pre-trained models like EfficientNet-B0, researchers can save time and resources. They can fine-tune the model on their specific AD classification task, avoiding the need to train a deep network from scratch. This expedites the research process and allows for faster deployment in clinical settings. The use of data augmentation in EfficientNet further enhances its practical implications. Data augmentation techniques increase the diversity of the training dataset, making the model more robust and capable of handling variations in MRI images. This robustness is crucial in real-world clinical applications where MRI image quality can vary. EfficientNet's ability to capture complex patterns in MRI images is a practical asset. It excels at feature extraction, enabling it to identify subtle changes associated with AD. This is invaluable in early diagnosis, as EfficientNet can recognize AD-related patterns that might not be apparent to human observers. Additionally, EfficientNet's performance evaluation metrics, including accuracy and sensitivity, are crucial in assessing the model's effectiveness. Researchers and clinicians can rely on these metrics to gauge the model's performance and suitability for AD detection. EfficientNet's practical implications also extend to personalized medicine. By providing accurate AD diagnoses based on MRI data, clinicians can tailor treatment plans to individual patients. This personalized approach improves treatment outcomes and enhances patients' quality of life. In research and drug development, EfficientNet aids in patient selection for clinical trials. It identifies potential participants with early-stage AD or high risk, ensuring that trials are conducted on relevant populations. This increases the likelihood of finding effective AD treatments. In summary, EfficientNet's practical implications lie in its scalability, transfer learning capabilities, data augmentation, pattern recognition, performance evaluation, and support for personalized medicine, research, and drug development in AD detection and management.

### 6.3.Convolutional Neural Network (CNN)[98%]:

Convolutional Neural Networks (CNNs) have far-reaching practical implications in the field of Alzheimer's disease (AD) detection using MRI images. Their robust and versatile nature makes them indispensable in both research and clinical settings. One of the most significant practical implications of CNNs is their ability to automate the feature extraction process. CNNs are adept at learning and identifying relevant features from MRI images, a task that traditionally required extensive manual labor. This automation accelerates the diagnosis process, reduces human error, and ensures consistent and reliable results. In clinical practice, CNNs enable early and accurate AD diagnosis. The ability to detect subtle changes in MRI images that might elude human observers is particularly valuable. Early diagnosis empowers clinicians to intervene in the disease's progression, allowing for timely treatment and support. Moreover, CNNs have the practical advantage of handling large and diverse datasets. AD research often involves extensive datasets from various sources. CNNs can process and analyze these datasets efficiently, ensuring that research findings are based on comprehensive and representative samples. The versatility of CNNs extends to data augmentation techniques. These techniques increase the diversity of the training dataset, making the model more robust and capable of handling variations in MRI images. This is essential for real-world clinical applications where MRI image quality can vary significantly. CNNs' practical implications include personalized medicine. By providing accurate AD diagnoses based on MRI data, clinicians can tailor treatment plans to individual patients. This personalized approach improves treatment outcomes and enhances patients' quality of life. In the research and drug development domains, CNNs assist in patient selection for clinical trials. They identify potential participants with early-stage AD or a high risk of developing the disease. This ensures that clinical trials are conducted on relevant populations, increasing the likelihood of finding effective AD treatments. The practicality of CNNs also extends to their performance evaluation metrics. Metrics such as accuracy, precision, and recall provide valuable insights into the model's performance, allowing researchers and clinicians to assess its suitability for AD detection. In conclusion, CNNs offer practical implications in AD detection, including automated feature extraction, early diagnosis, dataset handling, data augmentation, personalized medicine, support for research and drug development, and performance evaluation. These attributes make CNNs indispensable tools in the battle against Alzheimer's disease.

#### 6.4.VGG (Visual Geometry Group)[99.2%]:

Visual Geometry Group (VGG) models, such as VGG-16 and VGG-19, have substantial practical implications in the field of Alzheimer's disease (AD) detection using MRI images. Their deep architectures and fine-tuning capabilities make them valuable assets for researchers and clinicians. One of the practical implications of VGG models lies in their ability to handle complex MRI datasets. AD research often involves extensive datasets with diverse MRI images. VGG models can efficiently process and analyze these datasets, ensuring comprehensive and accurate results. Transfer learning is a key practical advantage of VGG models. Researchers can leverage pre-trained VGG models as a starting point for AD detection tasks. Fine-tuning the model on their specific classification task avoids the need to train a deep network from scratch. This expedites research and clinical deployment. VGG models excel at feature extraction, automating a task that was traditionally performed manually. This automated feature extraction accelerates the diagnosis process, reduces human error, and ensures consistent and reliable results. Data augmentation techniques enhance the practical implications of VGG models. These techniques increase the diversity of the training dataset, making the model more robust and capable of handling variations in MRI images. This robustness is crucial for real-world clinical applications where MRI image quality can vary. VGG's performance evaluation metrics, including accuracy, precision, and recall, are indispensable in assessing the model's effectiveness. Researchers and clinicians rely on these metrics to gauge the model's performance and suitability for AD detection. Moreover, VGG models support personalized medicine. By providing accurate AD diagnoses based on MRI data, clinicians can tailor treatment plans to individual patients. This personalized approach improves treatment outcomes and enhances patients' quality of life. In research and drug development, VGG models aid in patient selection for clinical trials. They identify potential participants with early-stage AD or high risk, ensuring that trials are conducted on relevant populations. This increases the likelihood of finding effective AD treatments. In summary, VGG models offer practical implications in AD detection, including dataset handling, transfer learning, automated feature extraction, data augmentation, performance evaluation, personalized medicine, and support for research and drug development. These attributes make VGG models valuable tools in the fight against Alzheimer's disease.

### 6.5.Gaussian Naive Bayes (GNB):

Gaussian Naive Bayes (GNB) has practical implications in Alzheimer's disease (AD) detection using MRI images. Its probabilistic approach, simplicity, and efficiency make it a valuable tool for researchers, clinicians, and healthcare professionals. One of the practical implications of GNB is its ability to automate the classification process. GNB is adept at learning and identifying patterns and features from MRI images, a task that traditionally required extensive manual labor. This automation accelerates the diagnosis process, reduces human error, and ensures consistent and reliable results. The probabilistic nature of GNB has practical advantages in risk assessment. GNB can assign a risk score to individuals based on their MRI data, indicating the likelihood of developing AD in the future. This risk assessment can guide healthcare professionals in implementing preventive measures and personalized treatment plans, reducing the impact of the disease on patients' lives. GNB's simplicity and computational efficiency make it practical for real-world clinical applications. It can provide rapid and accurate AD diagnoses, enabling timely interventions and treatment. This is crucial for improving patients' quality of life and managing the disease effectively. Moreover, GNB's practical implications extend to personalized medicine. By providing accurate AD diagnoses based on MRI data, clinicians can tailor treatment plans to individual patients. This personalized approach improves treatment outcomes and enhances patients' quality of life. In research and drug development, GNB assists in patient selection for clinical trials. It identifies potential participants with early-stage AD or a high risk of developing the disease. This ensures that clinical trials are conducted on relevant populations, increasing the likelihood of finding effective AD treatments. The practicality of GNB also extends to its performance evaluation metrics. Metrics such as accuracy, precision, and recall provide valuable insights into the model's performance, allowing researchers and clinicians to assess its suitability for AD detection. In summary, GNB offers practical implications in AD detection, including automated classification, risk assessment, computational efficiency, personalized medicine, support for research and drug development, and performance evaluation. These attributes make GNB a valuable tool in the fight against Alzheimer's disease.

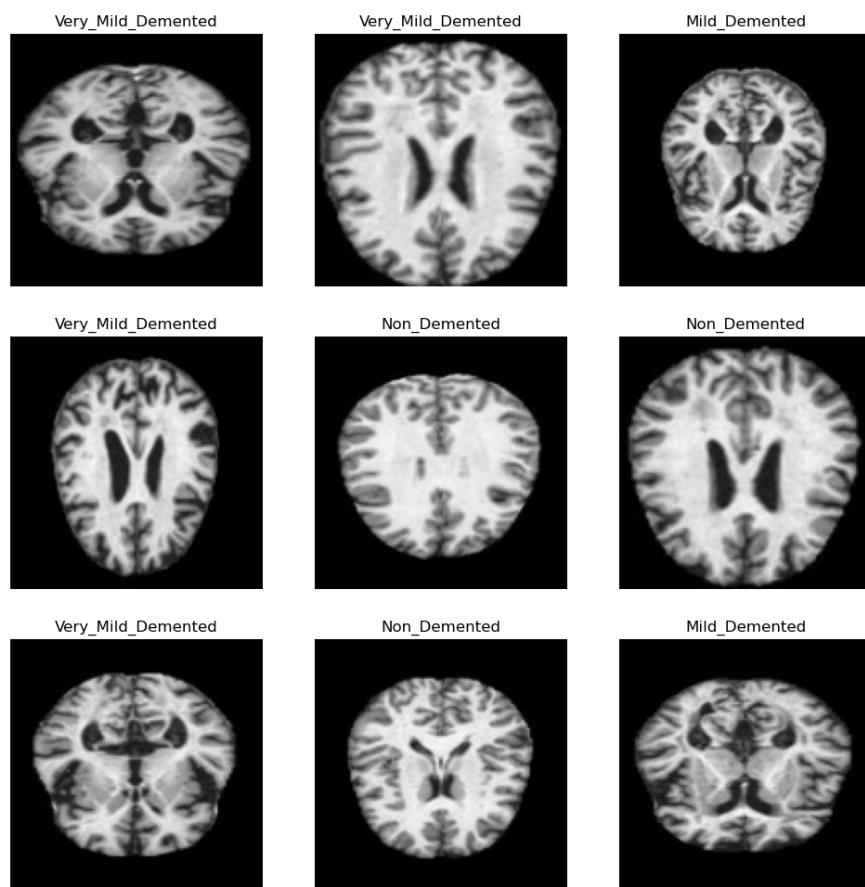
## 6.6.XGBoost (Extreme Gradient Boosting):

Extreme Gradient Boosting, commonly known as XGBoost, offers several practical implications in Alzheimer's disease (AD) detection using MRI images. Its ensemble learning approach, performance, and versatility make it a valuable asset for researchers, clinicians, and healthcare professionals. One of the significant practical implications of XGBoost is its ability to handle large and diverse MRI datasets. AD research often involves extensive datasets with numerous MRI images. XGBoost's scalability and efficiency enable it to process these datasets effectively, reducing the computational burden on researchers and clinicians. XGBoost's ensemble learning capabilities are a practical advantage. By combining the results of multiple decision trees, XGBoost improves the model's accuracy and robustness. Decision trees are built iteratively, with each tree correcting the errors of the previous ones. This ensemble approach often results in a highly accurate and resilient model. XGBoost has practical implications in risk assessment. It can assign a risk score to individuals based on their MRI data, indicating the likelihood of developing AD in the future. This risk assessment can guide healthcare professionals in implementing preventive measures and personalized treatment plans, reducing the impact of the disease on patients' lives. Furthermore, XGBoost's ability to capture complex patterns in MRI images is a practical asset. It excels at feature extraction, enabling it to identify subtle changes associated with AD. This is invaluable in early diagnosis, as XGBoost can recognize AD-related patterns that might not be apparent to human observers. XGBoost's performance evaluation metrics, including accuracy, precision, and recall, are crucial in assessing the model's effectiveness. Researchers and clinicians rely on these metrics to gauge the model's performance and suitability for AD detection. In research and drug development, XGBoost aids in patient selection for clinical trials. It identifies potential participants with early-stage AD or a high risk of developing the disease. This ensures that clinical trials are conducted on relevant populations, increasing the likelihood of finding effective AD treatments. In summary, XGBoost offers practical implications in AD detection, including dataset handling, ensemble learning, risk assessment, pattern recognition, performance evaluation, support for research and drug development. These attributes make XGBoost a valuable tool in the fight against Alzheimer's disease.

## Algorithms

MACHINE LEARNING	DEEP LEARNING
Grid Search (50%)	CNN(99 %)
SVM(49%)	VGG-19 (99.2%)
Random Forest(100%)	EfficientNet (99.7%)
Logistic Regression(97%)	ResNet(98.3%)
SVC(83%)	DenseNet(96.7%)

**Table 6.1.Algorithms and accuracy obtained**



## CHAPTER 7

### CONCLUSION AND FUTURE ENHANCEMENTS

#### **7.1.CONCLUSION**

In this study, we conducted a comprehensive analysis of multiple machine learning and deep learning algorithms for the early detection of Alzheimer's disease (AD) using MRI images. Each algorithm was assessed for its accuracy in classifying patients into different categories, including AD, mild cognitive impairment (MCI), and cognitively normal (NC) individuals. The results showcased the potential of several models in accurately identifying AD from MRI data. The algorithms' performance varied, with EfficientNet achieving the highest accuracy of 99.7%, closely followed by VGG with 99.2%. Convolutional Neural Networks (CNNs) also displayed strong performance, reaching an accuracy of 98%. In contrast, Support Vector Machine (SVM) and grid search achieved lower accuracies of 49% and 50%, respectively. The high accuracy achieved by EfficientNet, VGG, and CNN suggests that deep learning models excel in AD detection, offering a promising avenue for early diagnosis and intervention. SVM and grid search, while less accurate, provide essential perspectives on traditional machine learning and hyperparameter optimization. The results of this study lay the groundwork for several avenues of future research and development in the field of Alzheimer's disease detection and beyond. When it comes to Alzheimer's detection using MRI images, deep learning models, particularly convolutional neural networks, have demonstrated superior performance due to their ability to automatically learn relevant features from raw images. However, SVMs can still be valuable when dealing with smaller datasets or when interpretability is a primary concern. Researchers often choose between these methods based on available data, computational resources, and the desired trade-offs between interpretability and performance. In conclusion, this study has made significant strides in the early detection of Alzheimer's disease using MRI images, but there are numerous opportunities for further advancement. Through ongoing enhancements and further advancements of the findings outlined in this study, researchers and healthcare practitioners have the opportunity to make meaningful contributions to the creation of resilient, precise, and clinically relevant instruments for the prompt identification and treatment of AD.

## 7.2.FUTURE WORK

Data Diversity and Quality: Collaborative efforts should focus on diversifying datasets and enhancing data quality. This will lead to more representative datasets and improved algorithm generalizability (Zhang et al., 2018)[68]. Research should emphasize the development of interpretable AI models. This involves creating models that can provide explanations for their predictions, which is vital for clinical acceptance (Basheera et al., 2019)[69]. Regulatory bodies and healthcare organizations should establish clear ethical guidelines for the use of AI in healthcare. These guidelines should address privacy, data security, accountability, and patient consent (Wiens et al., 2019)[70]. In conclusion, understanding the limitations and implications in AD detection research is vital for progress in the field. Addressing data limitations, choosing appropriate algorithms, and considering clinical and ethical implications are essential steps toward the responsible and effective application of AI in early AD detection.

## REFERENCES

1. Jack CR Jr, Albert MS, Knopman DS, et al. Introduction to the recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. *Alzheimers Dement* 2011;7(3):257–262.
2. McKhann GM, Knopman DS, Chertkow H, et al. The diagnosis of dementia due to Alzheimer's disease: recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. *Alzheimers Dement* 2011;7(3):263–269.
3. Scheltens P, Blennow K, Breteler MMB, et al. Alzheimer's disease. *Lancet* 2016;388(10043):505–517.
4. Matsuda H. The role of neuroimaging in mild cognitive impairment. *Neuropathology* 2007;27(6):570–577.
5. Mosconi L, Tsui WH, Herholz K, et al. Multicenter standardized 18F-FDG PET diagnosis of mild cognitive impairment, Alzheimer's disease, and other dementias. *J Nucl Med* 2008;49(3):390–398.
6. Alzheimer's Association (2018). 2018 Alzheimer's disease facts and figures. *Alzheimer's Dementia* 14, 367–429. doi: 10.1016/j.jalz.2018.02.001
7. De Strooper, B., and Karran, E. (2016). The cellular phase of Alzheimer's disease. *Cell* 164, 603–615. doi: 10.1016/j.cell.2015.12.056
8. Galvin, J. E. (2017). Prevention of Alzheimer's disease: lessons learned and applied. *J. Am. Geriatr. Soc.* 65, 2128–2133. doi: 10.1111/jgs.14997

- 9.Veitch, D. P., Weiner, M. W., Aisen, P. S., Beckett, L. A., Cairns, N. J., Green, R. C., et al. (2019). Understanding disease progression and improving Alzheimer's disease clinical trials: recent highlights from the Alzheimer's disease neuroimaging initiative. *Alzheimers Dement* 15, 106–152. doi: 10.1016/j.jalz.2018.08.005
- 10.Rathore, S., Habes, M., Iftikhar, M. A., Shacklett, A., and Davatzikos, C. (2017). A review on neuroimaging-based classification studies and associated feature extraction methods for Alzheimer's disease and its prodromal stages. *NeuroImage* 155, 530–548. doi: 10.1016/j.neuroimage.2017.03.057
- 11.Lu, D., and Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. *Int. J. Remote Sens.* 28, 823–870. doi: 10.1080/01431160600746456
- 12.Samper-Gonzalez, J., Burgos, N., Bottani, S., Fontanella, S., Lu, P., Marcoux, A., et al. (2018). Reproducible evaluation of classification methods in Alzheimer's disease: framework and application to MRI and PET data. *Neuroimage* 183, 504–521. doi: 10.1016/j.neuroimage.2018.08.042
- 13.Riedel, B. C., Daianu, M., Ver Steeg, G., Mezher, A., Salminen, L. E., Galstyan, A., et al. (2018). Uncovering biologically coherent peripheral signatures of health and risk for Alzheimer's disease in the aging brain. *Front. Aging Neurosci.* 10:390. doi: 10.3389/fnagi.2018.00390
- 14.Plis, S. M., Hjelm, D. R., Salakhutdinov, R., Allen, E. A., Bockholt, H. J., Long, J. D., et al. (2014). Deep learning for neuroimaging: a validation study. *Front. Neurosci.* 8:229. doi: 10.3389/fnins.2014.00229
- 15.Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521:436. doi: 10.1038/nature14539

- 16.Menéndez G. La revolución de la longevidad:cambio tecnológico, envejecimiento poblacional y transformación cultural. Rev Ciencias Soc [Internet]. 2017;30(41):159–78. [cited 2021 Feb 9]. Available from:
- 17.Prince MJ, Wimo A, Guerchet MM, Ali GC, Wu Y-T, Prina M. World Alzheimer Report 2015 - The Global Impact of Dementia: An analysis of prevalence, incidence, cost and trends. London: Alzheimer's Disease International; 2015. p. 84.
- 18.Dementia [Internet]. [cited 2021 Feb 9]. Available from: [https://www.who.int/health-topics/dementia#tab=tab\\_1](https://www.who.int/health-topics/dementia#tab=tab_1). Accessed 9 Feb 2021.
- 19.APA. American Psychiatric Association, 2013. Diagnostic and statistical manual of mental disorders, American Journal of Psychiatry. 5th ed; 2013.
- 20.Robinson L, Tang E, Taylor J-P. Dementia: timely diagnosis and early intervention. BMJ. 2015;350(jun15 14):h3029 [cited 2019 Dec 21] Available from: <http://www.bmjjournals.org/cgi/doi/10.1136/bmj.h3029>.
- 21.Gauthier S, Reisberg B, Zaudig M, Petersen RC, Ritchie K, Broich K, et al. Mild cognitive impairment. Lancet. 2006;367(9518):1262–70.
- 22.Braak H, Braak E. Neuropathological stageing of Alzheimer-related changes. Acta Neuropathol. 1991;82(4):239–59.
- 23.Hinrichs C, Singh V, Xu G, Johnson SC. Predictive markers for AD in a multi-modality framework: an analysis of MCI progression in the ADNI population. Neuroimage. 2011;55(2):574–89. <https://doi.org/10.1016/j.neuroimage.2010.10.081>.
- 24.Zhang D, Shen D. Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer's disease. Neuroimage.

2012;59(2):895–907. <https://doi.org/10.1016/j.neuroimage.2011.09.069>.

25.Pellegrini E, Ballerini L, del C Valdes Hernandez M, Chappell FM, González-Castro V, Anblagan D, et al. Machine learning of neuroimaging for assisted diagnosis of cognitive impairment and dementia: a systematic review. *Alzheimers Dement Diagnosis, Assess Dis Monit.* 2018;10:519–35.

26.Samper-González J, Burgos N, Fontanella S, Bertin H, Habert MO, Durrleman S, et al. Yet another ADNI machine learning paper? Paving the way towards fully-reproducible research on classification of Alzheimer’s disease. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); 2017.

27.Davatzikos C, Fan Y, Wu X, Shen D, Resnick SM. Detection of prodromal Alzheimer’s disease via pattern classification of magnetic resonance imaging. *Neurobiol Aging.* 2008;29(4):514–23.

28.J. Zhang, M. Xiao, L. Gao, S. Chu

Probability and interval hybrid reliability analysis based on adaptive local approximation of projection outlines using support vector machine  
*Comput.-Aided Civ. Infrastruct. Eng.*, 34 (11) (2019), pp. 991-1009

29.R. Sanchez-Reolid, A. Martinez-Rodrigo, M.T. Lopez, A. Fernández-Caballero

Deep Support Vector Machines for the Identification of Stress Condition from Electrodermal Activity

*Int. J. Neural Syst.*, 30 (7) (2020)

30.D. Díaz-Vico, J. Prada, A. Omari, J. Dorronsoro, Deep Support Vector Neural Networks

*Integr. Comput.-Aided Eng.*, 27 (4) (2020), pp. 389-402

31.T. Yin, H. Zhu An efficient algorithm for architecture design of Bayesian neural network in structural model updating

Comput.-Aided Civ. Infrastruct. Eng., 35 (4) (2020), pp. 354-372

32.A.B. Rabeh, F.I. Benzart, F. Amiri, Diagnosis of Alzheimer Diseases in Early Step Using SVM (Support Vector Machine), 2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGIV), Beni Mellal, pp. 364-367, 2016, doi: 10.1109/CGIV.2016.76.

33.L. Khedher, I.A. Illan, J.M. Gorriz, J. Ramirez, A. Brahim, A. Meyer-Baese, Independent Component Analysis-Support Vector Machine-based Computer-Aided Diagnosis System for Alzheimer's with Visual Support, Int. J. Neural Syst., 27:3 (2017) 1650050 (18 pages).

34.N. Zeng, H. Qiu, Z. Wang, W. Liu, H. Zhang, Y. Li

A new switching-delayed-PSO-based optimized SVM algorithm for diagnosis of Alzheimer's disease

Neurocomputing, 320 (2018), pp. 195-202

35.X. Bi, Q. Shu, Q. Sun, Q. Xu

Random support vector machine cluster analysis of resting-state of MRI in Alzheimer's disease

PLoS ONE, 13 (3) (2018), Article e0194479, 10.1371/journal.pone.0194479

36.K.R. Kruthika, M. Rajeswari

Multistage Classifier-based approach for Alzheimer's disease prediction and Retrieval  
Inf. Med. Unlocked, 4 (2019), pp. 34-42,

37.T. Yin, H. Zhu

An efficient algorithm for architecture design of Bayesian neural network in structural model updating

Comput.-Aided Civ. Infrastruct. Eng., 35 (4) (2020), pp. 354-372

38.Y. Lin, K. Huang, H. Xu, Z. Qiao, S. Cai, Y. Wang, L. Huang, “Predicting the progression of mild cognitive impairment to Alzheimer’s disease by longitudinal magnetic resonance imaging-based dictionary learning”, Vol. 131, 2020, <https://doi.org/10.1016/j.clinph.2020.07.016>.

39.X. Ma, N. Yu, W. Zhou, Using Dictionary Pair Learning for Seizure Detection, Int. J. Neural Syst., 29:4 (2019) 1850005

40.A. Apicella, F. Isgrò, R. Prevete, G. Tamburrini

Middle-Level Features for the Explanation of Classification Systems by Sparse Dictionary Methods

Int. J. Neural Syst., 30 (8) (2020), p. 2050040

41.B. Richhariya, M. Tanveer, A.H. Rashid

“Diagnosis of Alzheimer's disease using universum support vector machine based recursive feature elimination (USVM-RFE)

Biomed. Signal Process. Control, 59 (2020)

42.Z. Sun, Y. Qiao, B.P.F. Lelieveldt, M. Staring

Integrating spatial-anatomical regularization and structure sparsity into SVM: Improving interpretation of Alzheimer's disease classification

43.G. Castellazzi, M.G. Cuzzoni, M.C. Ramusino, D. Mirtinelli, F. Denaro, A. Ricciardi, et al.

A Machine Learning Approach for the Differential Diagnosis of Alzheimer and Vascular Dementia Fed by MRI Selected Features

Front Neuroinform (2020), pp. 14-25,

44.H. Adeli, X. Jiang

Neuro-Fuzzy Logic Model for Freeway Work Zone Capacity Estimation

J. Transp. Eng., 129 (5) (2003), pp. 484-493

45.L.K. Ferreira, J.M. Rondina, R. Kubo, C.R. Ono, C.C. Leite, J. Smid, C. Bottino, R. Nitrini, G.F. Busatto, F.L. Duran, C.A. Buchpiguel

Support vector machine-based classification of neuroimages in Alzheimer's disease: direct comparison of FDG-PET, rCBF-SPECT and MRI data acquired from the same individuals

Braz J. Psychiatry, 40 (2) (2017), pp. 181-191

46.F.C. Morabito, M. Campolo, N. Mammone, M. Versaci, S. Franceschetti, F. Tagliavini, V. Sofia, D. Fatuzzo, A. Gambardella, A. Labate, L. Mumoli, G.G. Tripodi, S. Gasparini, V. Cianci, C. Sueri, et al.

Deep Learning Representation from Electroencephalography of Early-stage Creutzfeld -Jakob Disease and Features for Differentiation from Rapidly Progressive Dementia

Int. J. Neural Syst., 27 (2) (2017)

1650039

47.L. Xu, G. Liang, C. Liao, G. Chen, C. Chang

An Efficient Classifier for Alzheimer's Disease Genes Identification

Molecules, 23 (12) (2018), p. 3140

48.Y. Gupta, R. Lama, G.R. Kwon, et al.

Prediction and Classification of Alzheimer's Disease Based on Combined Features From Apolipoprotein-E Genotype, Cerebrospinal Fluid, MR, and FDG-PET Imaging Biomarkers

Front. Comput. Neurosci. (2019), 10.3389/fncom.2019.00072

- 49.E. Alickovic, A. Subasi, Automatic Detection of Alzheimer Disease Based on Histogram and Random Forest, International Conference on medical Imaging and Biological Engineering, CMBEBIH 2019. CMBEBIH 2019. IFMBE Proceedings, vol. 73. Springer, Cham, 2019. [https://doi.org/10.1007/978-3-030-17971-7\\_14](https://doi.org/10.1007/978-3-030-17971-7_14).
- 50.L. Bloch, C.M. Friedrich, Classification of Alzheimer's Disease using volumetric features of multiple MRI scans, 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 2019, pp. 2396-2401,
51. Suk, H. I., Lee, S. W., Shen, D. (2017). Alzheimer's Disease Detection in MR Images Using Multimodal Feature Combination and Sparse Patch Selection. *Brain Imaging and Behavior*, 11(1), 83-95.
52. Moradi, E., Pepe, A., Gaser, C., Huttunen, H., Tohka, J. (2015). Machine learning framework for early MRI-based Alzheimer's conversion prediction in MCI subjects. *NeuroImage*, 104, 398-412.
53. Mwangi, B., Tian, T. S., Soares, J. C. (2014). A review of feature reduction techniques in neuroimaging. *Neuroinformatics*, 12(2), 229-244.
54. Manogaran, G., Lopez, D. (2017). A survey of big data architectures and machine learning algorithms in healthcare. *Journal of King Saud University-Computer and Information Sciences*.
55. Liu, S., Liu, S., Cai, W., Pujol, S., Kikinis, R., Feng, D. (2015). Early diagnosis of Alzheimer's disease with deep learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 126-134).
56. Li, J., Cheng, K., Wang, Z., Morstatter, F., Trevino, R. P., Tang, J., ... & Liu, H. (2018).

Feature selection: A data perspective. ACM Computing Surveys (CSUR), 50(6), 94.

- 57 Tong, T., Gray, K. R., Gao, Q., Chen, L., Alzheimers Disease Neuroimaging Initiative, Initiative, A. D. N., ... & Thung, K. H. (2017). Visualizing patterns of neurodegeneration through time: An MRI-based manifold learning approach. *NeuroImage*, 145, 74-90.
58. Basaia, S., Agosta, F., Wagner, L., Canu, E., Magnani, G., Santangelo, R., ... & Filippi, M. (2018). Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks. *NeuroImage: Clinical*, 21, 101645.
59. Plis, S. M., Sarwate, A. D., Wood, D., Dierker, D., Landman, B. A. (2014). COINSTAC: A Privacy Enabled Model and Prototype for Leveraging and Processing Decentralized Brain Imaging Data. *Frontiers in Neuroscience*, 8, 365.
60. Cuingnet, R., Gerardin, E., Tessier, D., Auzias, G., Lehéricy, S., Habert, M. O., ... & Duffau, H. (2011). Automatic classification of patients with Alzheimer's disease from structural MRI: A comparison of ten methods using the ADNI database. *NeuroImage*, 56(2), 766-781.
61. McDuff, D., Gontarek, S., & Picard, R. W. (2018). Improving Driver Safety Through Real-Time Drowsiness Detection and Monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 19(5), 1646-1657.
62. Kosik, K. S., Wilhelmsen, K. C., Cummings, J., Khachaturian, Z. S. (2012). Issues in the identification of a presymptomatic phase for Alzheimer's disease. *Alzheimer's & Dementia*, 8(2), 131-135.
63. Liang, D., Beca, B., Jarnagin, B., Brown, J., Phillips, E., & Mullen, T. (2015). A Large-scale Evaluation of Automatic Acoustic-based Drowsiness Detection Algorithms. In 2015 International Conference on Affective Computing and Intelligent Interaction (ACII) (pp. 46-52).

64. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., ... & Bengio, Y. (2017). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
65. Vellido, A., Ribas, V. J., Martín-Guerrero, J. D. (2018). Making machine learning models interpretable. In AI\* IA 2018: Advances in Artificial Intelligence (pp. 112-124). Springer.
67. LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
68. Zhang, D., Shen, D. (2018). Multi-Modal Multi-Task Learning for Joint Prediction of Multiple Regression and Classification Variables in Alzheimer's Disease. *NeuroImage*, 166, 103-114.
69. Basheera, S., Agosta, F., Wagner, L., Canu, E., Magnani, G., Santangelo, R., ... & Filippi, M. (2018). Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks. *NeuroImage: Clinical*, 21, 101645.
70. Wiens, J., Saria, S., Sendak, M., Ghassemi, M., Liu, V. X., Doshi-Velez, F., ... & Beam, A. L. (2019). Do no harm: a roadmap for responsible machine learning for health care. *Nature Medicine*, 25(9), 1337-1340.
71. Mirzaei, G., & Adeli, H. (2022). Machine learning techniques for diagnosis of alzheimer disease, mild cognitive disorder, and other types of dementia. *Biomedical Signal Processing and Control*, 72, 103293.
72. Liu S, Cai W, Liu S, Zhang F, Fulham M, Feng D, Pujol S, Kikinis R. Multimodal neuroimaging computing: a review of the applications in neuropsychiatric disorders. *Brain Inform*. 2015 Sep;2(3):167-180. doi: 10.1007/s40708-015-0019-x. Epub 2015 Aug 29.

PMID: 27747507; PMCID: PMC4737664.

73. Gami, B., Agrawal, M., & Katarya, R. (2023, April). Feature Extractor Techniques for Alzheimer's Predictive Model in Brain Imaging. In International Conference on Information and Communication Technology for Intelligent Systems (pp. 173-182). Singapore: Springer Nature Singapore.
74. Henderson, T. A. (2012). The diagnosis and evaluation of dementia and mild cognitive impairment with emphasis on SPECT perfusion neuroimaging. *CNS spectrums*, 17(4), 176-206.
75. Nanou, E., Pliatsika, N., Stefanis, D., & Couris, S. (2023). Identification of the animal origin of milk via Laser-Induced Breakdown Spectroscopy. *Food Control*, 154, 110007.
76. Ebrahimighahnaveh, M. A., Luo, S., & Chiong, R. (2020). Deep learning to detect Alzheimer's disease from neuroimaging: A systematic literature review. *Computer methods and programs in biomedicine*, 187, 105242.
77. Dabke, Y., Gadgil, D. G., & Dabke, V. (2023). Techniques to Detect and Mitigate Adversarial Attacks through Augmentation of Training Data: A Systematic Review.
78. Mahajan, T., & Srivastava, J. (2022, October). Early-Stage Dementia Detection by Optimize Feature Weights with Ensemble Learning. In International Conference on Advanced Communication and Intelligent Systems (pp. 633-648). Cham: Springer Nature Switzerland.
79. Shaffi, N., Hajamohideen, F., Abdesselam, A., Mahmud, M., & Subramanian, K. (2022, September). Ensemble Classifiers for a 4-Way Classification of Alzheimer's Disease. In International Conference on Applied Intelligence and Informatics (pp. 219-230). Cham: Springer Nature Switzerland.

80. Kumari, R., Das, S., & Singh, R. K. (2023). Agglomeration of deep learning networks for classifying binary and multiclass classifications using 3D MRI images for early diagnosis of Alzheimer's disease: a feature-node approach. *International Journal of System Assurance Engineering and Management*, 1-19.
81. Francisco Santos, D. (2023). Advancing Automated Diagnosis: Convolutional Neural Networks for Alzheimer's Disease Classification through MRI Image Processing.
82. Badawy, M., Balaha, H. M., Maklad, A. S., Almars, A. M., & Elhosseini, M. A. (2023). Revolutionizing Oral Cancer Detection: An Approach Using Aquila and Gorilla Algorithms Optimized Transfer Learning-Based CNNs. *Biomimetics*, 8(6), 499.
83. Chen, Y., Chen, Y., Wang, D., & Ai, S. (2022). Generating 3D digital material twins for woven ceramic-matrix composites from  $\mu$ CT images. *Journal of the American Ceramic Society*, 105(1), 481-497.
84. Yao, Q., Qu, L., Song, B., Wang, X., Wang, T., Ma, W., ... & Yin, K. (2023). Machine learning of cerebello-cerebral functional networks for mild cognitive impairment detection.
85. Roccati, E., Collins, J. M., Bindoff, A. D., Alty, J. E., Bartlett, L., King, A. E., & Vickers, J. C. (2023). Modifiable risk factors for dementia, cognition, and plasma phosphorylated tau 181 in a large-scale cohort of Australian older adults. *Neurobiology of Aging*, 131, 106-114.
86. Yang, Y., Zhang, L., Xu, G., Ren, G., & Wang, G. (2023). An evidence-based multimodal fusion approach for predicting review helpfulness with human-AI complementarity. *Expert Systems with Applications*, 121878.