DEEPWAVEMRI: EARLY ALZHEIMER'S DETECTION

A Project Report submitted in the partial fulfilment of the Requirements for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

M. Yuva Sravani (21471A05H6)

P. Priyanka (21471A05I9)

G. Sirisha (21471A05F2)

Under the esteemed guidance of

Dr. Sireesha Moturi, M.Tech , Ph.D.

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Tire -1 NIRF rank in the band of 201-300 and an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET-522601

2024-2025

NARASARAOPETA ENGINEERING COLLEGE

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name **"DEEPWAVEMRI: EARLY ALZHEIMER'S DETECTION"** is a bonafide work done by the team M. Yuva Sravani(21471A05H6), P. Priyanka(21471A05I9), G. Sirisha(21471A05F2) in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2024-2025.

PROJECT GUIDE

Dr.Sireesha Moturi, M.Tech., Ph.D.,

Associate Professor

PROJECT CO-ORDINATOR

Dr.Sireesha Moturi, M.Tech., Ph.D.,

Associate Professor

HEAD OF THE DEPARTMENT

Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,

Professor & HOD

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled **“DEEPWAVEMRI: EARLY ALZHEIMER’S DETECTION”** is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has not been submitted for any other degree or professional qualification except as specified.

M. Yuva Sravani (21471A05H6)
P. Priyanka (21471A05I9)
G. Sirisha (21471A05F2)

ACKNOWLEDGEMENT

We wish to express our thanks to various personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M. V.Koteswara Rao**, **B.Sc.,** who took keen interest in us in every effort throughout this course.We owe our sincere gratitude to our beloved principal **Dr. S. Venkateswarlu, Ph.D.,** for showing his kind attention and valuable guidance throughout the course.

We express our deep-felt gratitude towards **Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,** HOD of CSE department and also to our guide **Dr.Sireesha Moturi M.Tech., Ph.D.,** of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks to **Dr.Sireesha Moturi M.Tech., Ph.D.,** Associate professor& Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant sourceof inspiration for us throughout this project work.

We extend our sincere thanks to all the other teaching and non-teaching staff in the department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who were involved in giving valuable suggestions clarified our doubts, which really helped us in successfully completing our project.

By

M. Yuva Sravani (21471A05H6)

P. Priyanka (21471A05I9)

G. Sirisha (21471A05F2)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as researchfor the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problem

NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern softwaretools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-basedsystems.

PSO2: Acquaint module knowledge on emerging trends of the modern era inComputer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur,environmental and social issues.

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineeringfundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems relatedto academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Program Outcomes

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, andsynthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineeringactivities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO421.1: Analyze the System of Examinations and identify the problem.

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		√											√		
C421.2	√		√		√								√		
C421.3				√		√	√	√					√		
C421.4			√			√	√	√					√	√	
C421.5					√	√	√	√	√	√	√	√	√	√	√
C421.6									√	√	√		√	√	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO’s and PO’s:

1. Low level
2. Medium level
3. High level

Project mapping with various courses of Curriculum with Attained PO’s:

Name of the course from which principles are applied in this project	**Description of the device**	**Attained PO**
C2204.2, C22L3.2	Gathering the requirements and defining the problem, plan to developa model for classifying images using CNN	PO1, PO3
CC421.1, C2204.3, C22L3.2	Each and every requirement is critically analyzed, the process modelis identified	PO2, PO3
CC421.2, C2204.2, C22L3.3	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done by all our three members in the form of a group	PO10
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation is done and the project will be classifying the 4 stages of Alzheimer’s disease	PO4, PO7
C32SC4.3	The physical design includes website to check whether the MRI is valid or not along with the stage prediction	PO5, PO6

ABSTRACT

Alzheimer's disease (AD) is the common type of dementia, which is a decline in cognition with significant memory loss that cannot be reversed causing the loss of independent functionality. Early detection is thus important for proper management because the current diagnostic methods, among them being cognitive testing, behavioural assessments, brain imaging, and history, are both unreliable and insufficient for the early stage diagnosis. The paper will propose a novel approach for early-stage AD detection based on MRI capability with enhanced image processing, using convolutional neural networks in combination with Wavelet Transform, Random Forest, and Support Vector Machine techniques. Our approach applies the Discrete Wavelet Transform of the MRI images to decompose them into multiple frequency frames, and further features are extracted by processing the wavelet coefficients with kurtosis-based thresholding for de-noising enhanced representations. Then, the findings are used to train on a broad data set offered by Kaggle with CNN, Random Forest, and SVM models which can classify different stages of Alzheimer's diseases. The proposed approach improves the accuracy of detection significantly, which provides a more reliable solution for early diagnosis.

INDEX

LIST OF FIGURES

1. INTRODUCTION

Alzheimer’s disease is a neurodegenerative disease that initiates slowly and successively worsens and is the cause of 60–70% of cases of dementia. The early symptom is difficulty in remembering recent events. As the disease move along, symptoms can involve problems with language, disorientation, mood swings, loss of motivation, self-neglect, and some mental issues.

Step-by-step, physical functions will be lost, and then leads to death. As of October 2023, you are being trained on data. 55 million people across the globe are said to have dementia with AD being the leading sort which accounts for 60-70% of cases. The tempo of succession may differ; however, the life expectancy post analysis is around three to twelve years [1].

Globally, Alzheimer's disease affects over 55 million people, with cases projected to triple by 2050 due to aging populations. In India, over 4 million people live with Alzheimer's, with rising numbers driven by an aging population and changing lifestyle patterns.

The cause of AD is not completely grasped. There are several risk factors associated with its onset including a history of head trauma, depression in clinical settings and hypertension or high cholesterol levels. The usual diagnosis comes from knowing about how long it has lasted and conducting cognitive tests while ignoring any other possible causes with unnecessary blood tests and advanced medical imaging techniques. The brain-imaging technologies that mostly we use are: Magnetic resonance imaging (MRI), Computerized tomography (CT), Positron emission tomography (PET).

MRI provides high-resolution images of brain structures and is particularly effective in detecting early signs of brain atrophy. CT scans are useful for ruling out other potential causes of cognitive impairment, such as strokes or brain tumours. PET imaging, on the other hand, enables the visualization of metabolic activity and the presence of amyloid-beta plaques, a hallmark feature of Alzheimer’s pathology. Blood tests and cerebrospinal fluid analysis are also being increasingly utilized to detect biomarkers like amyloid-beta and tau proteins, which are indicative of Alzheimer’s disease.

Treatment for Alzheimer's disease remains symptomatic, as no cure currently exists. Some medications aim to alleviate symptoms and improve the quality of life for patients [2]. Additionally, non-pharmacological interventions, including cognitive stimulation therapy, physical exercise, and dietary modifications, have shown promise in slowing disease progression and enhancing overall well-being. Magnetic resonance imaging (MRI), Computerized tomography (CT), Positron emission tomography (PET) brain imaging are shown in Fig 1.1.

Fig 1.2 illustrates the projected increase in Alzheimer's disease cases from 1995 to 2050, highlighting a significant rise from 377,000 to 959,000 cases (in thousands) over time.

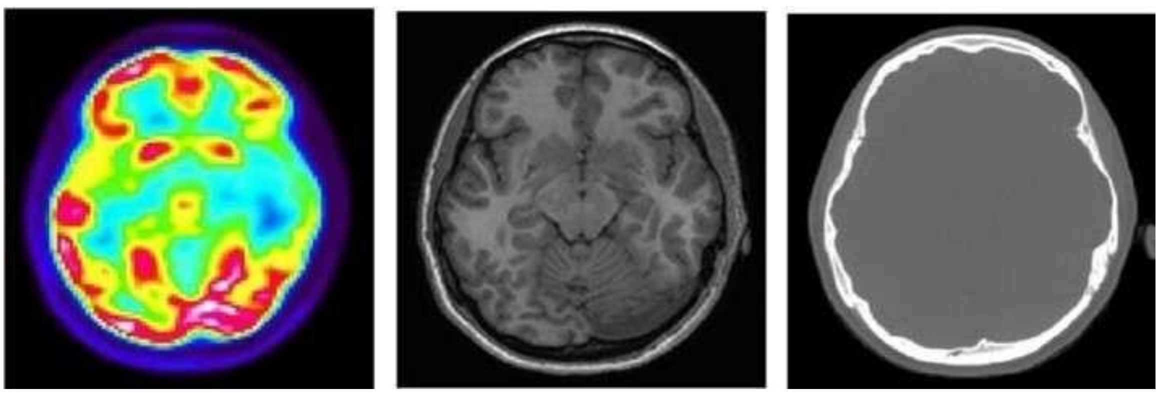

Fig 1.1 PET scan, MRI scan, CT scan of brain

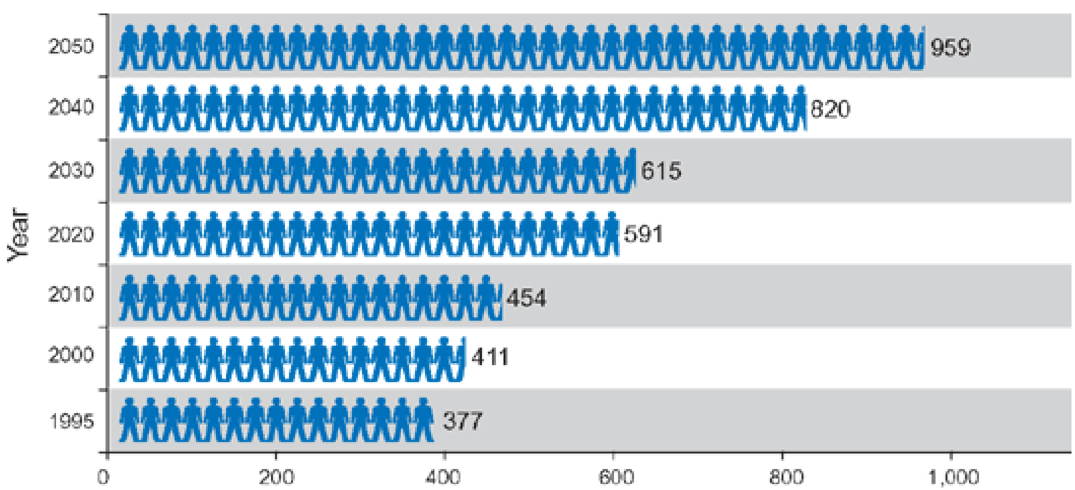

Fig 1.2 Projected Increase in Alzheimer's Disease Cases (1995–2050)

2. LITERATURE SURVEY

Alzheimer’s disease is a global epidemic impacting millions, causing memory loss, cognitive decline, and behavioural changes. Early detection is crucial but often limited by traditional, costly diagnostic methods. Machine Learning innovations, such as Deep Learning models, speech analysis, and graph-based algorithms, now offer scalable, non-invasive, and highly accurate solutions for early diagnosis and management.

Several methods have been explored for the early diagnosis and classification of Alzheimer’s disease using MRI scans. Bansal et al. [3] utilized a Convolutional Neural Network (CNN) to detect Alzheimer’s at an early stage, though the approach required large datasets and had a high computational cost. Kumar et al. [4] employed a Support Vector Machine (SVM) for classification, but the method struggled with nonlinear separability without proper kernel functions.

Sharma and Gupta [5] implemented a Random Forest algorithm for feature selection and classification; however, it was less effective with high-dimensional data and required parameter tuning. Lee et al. [6] introduced a deep Learning model incorporating Curvelet Transform to enhance feature extraction, yet its complexity made it challenging to implement without specialized hardware. In contrast, the proposed method, DeepWaveMRI, integrates CNN with Wavelet Transform to improve early detection accuracy, though it demands robust pre-processing and comes with higher model complexity.

Recent advancements in predictive models have significantly improved Alzheimer's disease detection. Traditional methods like Random Forests and Decision Trees achieve 93.69% accuracy, while deep learning models offer even greater precision. DeepCurvMRI, combining Curvelet Transforms with CNNs, outperforms traditional models with 98.71% accuracy. Graph Convolutional Networks (GCNs) and the Second-Generation Curvelet Transform (SGCT) further enhance classification performance. Additionally, Natural Language Processing (NLP) enables early detection through speech analysis, and Reservoir Computing (RC) models like Echo State Networks (ESNs) effectively analyze brain activity from EEG signals. These innovations provide more accurate and interpretable AD detection methods.

2.1 Deep Learning

Deep Learning has emerged as a powerful tool in the early detection and diagnosis of Alzheimer's disease (AD), leveraging its ability to analyze complex patterns in large datasets, such as brain imaging and genetic data. By using convolutional neural networks (CNNs), Deep Learning models can identify subtle changes in brain structure, such as atrophy in the hippocampus, which is a hallmark of Alzheimer's. These models are trained on MRI or PET scan images, learning to recognize patterns that are often imperceptible to the human eye [8] . Additionally, Deep Learning techniques are being applied to analyze genetic data and biomarkers, aiding in the prediction of disease progression and helping clinicians develop personalized treatment plans. As a result, Deep Learning holds the potential to revolutionize Alzheimer's research by enabling earlier diagnosis, improving prognostic accuracy, and ultimately enhancing the understanding of the disease at a molecular level.

2.2 Some Deep Learning methods

Deep Learning has revolutionized medical imaging by enabling automated diagnosis through advanced neural networks. Some of the widely used Deep Learning methods for Alzheimer's detection include Convolutional Neural Networks (CNNs) for feature extraction, Recurrent Neural Networks (RNNs) for sequential data analysis, and Auto encoders for anomaly detection. Additionally, models such as Transformers and Graph Neural Networks (GNNs) are emerging in medical image analysis, improving disease classification and early detection.

2.3 Applications of Deep Learning

Deep Learning plays a crucial role in healthcare and medical diagnostics, particularly in analysing MRI scans for neurodegenerative diseases like Alzheimer's. It is used in disease classification, image segmentation, feature extraction, and anomaly detection. Apart from medical imaging, Deep Learning is applied in drug discovery, genomics, personalized treatment recommendations, and predictive analytics in patient care. The ability of Deep Learning models to analyze large datasets has significantly improved the accuracy and efficiency of clinical decision-making.

2.4 Prevalence of MRI-Based AD

Alzheimer's disease is one of the most common neurodegenerative disorders, affecting millions worldwide. Traditional diagnostic methods rely on cognitive assessments and clinical evaluations, which often fail to detect early-stage Alzheimer's. MRI-based detection, powered by Deep Learning, has gained prominence as it enables early identification of brain abnormalities. The use of Deep Learning models such as CNNs and Wavelet Transform enhances the precision of Alzheimer's classification by analyzing subtle changes in brain structure.

2.5 Importance of Deep Learning in Alzheimer's Detection

Deep Learning significantly improves the accuracy of Alzheimer's disease diagnosis by automating feature extraction and classification from MRI scans. Traditional Machine Learning models require manual feature selection, whereas Deep Learning models automatically learn patterns from brain images, reducing human error. By integrating CNNs, FDWT, and kurtosis-based de noising, Deep Learning models can detect Alzheimer's in its early stages, aiding in timely intervention and better patient care.

3. EXISTING SYSTEM

- **ADNI (Alzheimer's Disease Neuroimaging Initiative)**

The ADNI is one of the largest and most well-known research initiatives aimed at studying Alzheimer's disease and related neurodegenerative disorders. It provides a vast collection of neuroimaging data, genetic data, and clinical assessments. Many Machine Learning and Deep Learning models have been trained on this dataset to identify biomarkers and predict the onset and progression of Alzheimer's disease. Models such as Random Forest, Support Vector Machines (SVM), and Deep Learning techniques like Convolutional Neural Networks (CNNs) have been applied to MRI, PET scan, and genetic data to classify patients into different stages of Alzheimer's (e.g., mild cognitive impairment, moderate, or severe) [9] .

- **Alzheimer's Prediction with Machine Learning (APML)**

The APML system uses Machine Learning techniques such as SVMs, decision trees, and ensemble methods to predict the likelihood of a person developing Alzheimer's disease based on neuroimaging data (MRI and PET scans), genetic information, and clinical variables. The system aims to provide an early diagnosis by leveraging a combination of biomarkers and clinical features, improving predictive accuracy and helping healthcare professionals plan treatment options more effectively.

- **MRI-Based Alzheimer's Classification System**

Various Deep Learning models have been developed to detect Alzheimer's disease from MRI data. These systems typically use CNNs to automatically extract features from brain scans, which are then used to classify the images into categories such as "healthy" or "Alzheimer's". Some systems also incorporate techniques such as transfer learning, where pre-trained models are fine-tuned on domain-specific data to improve performance and reduce the need for large amounts of labeled data.

- **Alzheimer's Disease Prediction using Neuroimaging and Clinical Data**

Some systems combine neuroimaging data (MRI, PET scans) with clinical data, including cognitive test results and demographic information, to create a more holistic model for Alzheimer's disease prediction. These hybrid models use both Machine Learning algorithms and Deep Learning approaches, such as CNNs and Long Short-Term Memory

(LSTM) [10] networks, to process structured and unstructured data. This multi-modal approach helps improve the accuracy and robustness of predictions, as it integrates various data types that reflect different aspects of the disease.

- **Alzheimer's Disease Classification with Ensemble Learning**

Ensemble Learning approaches, such as Random Forest and Gradient Boosting Machines (GBM), have been applied to Alzheimer's disease classification. These systems typically combine multiple models to improve performance by reducing the likelihood of over fitting and increasing generalization. These methods have been particularly useful in combining different types of data, such as MRI, PET scans, and clinical records, to make accurate predictions regarding Alzheimer's onset and progression.

- **AI-Based Early Detection Systems (AI-ED)**

Many AI-driven systems have been developed for early detection of Alzheimer's disease using Machine Learning models. These systems often combine CNNs with other techniques, such as recurrent neural networks (RNNs) [11] and LSTMs, to analyze longitudinal data, detect subtle changes over time, and make early predictions. These AI systems have the potential to assist clinicians in identifying Alzheimer's at a much earlier stage, allowing for timely intervention and improved patient outcomes.

4. PROPOSED SYSTEM

The proposed system, “DeepWaveMRI”, leverages advanced pre-processing techniques and Machine Learning models to detect Alzheimer’s disease from MRI scans. The system begins with resizing the images to 208x208 pixels, followed by applying Discrete Wavelet Transform (DWT) [12] to capture key frequency components. A kurtosis-based de noising technique is then used to reduce noise while preserving important image features. The pre-processed images are then fed into a Convolutional Neural Network (CNN), which automatically extracts complex patterns for classification. The system categorizes patients into four classes: Mild Demented, Moderate Demented, Non-Demented, and Very Mild Demented, offering a robust approach to early diagnosis as shown in Fig 4.1.

The advantages of the “DeepWaveMRI” system are manifold. By combining wavelet transform with CNN, it enhances the ability to detect subtle features in medical images, improving classification accuracy. The kurtosis-based de noising ensures that the model focuses on relevant image features without being affected by noise, resulting in more reliable predictions. Moreover, the integration of Machine Learning models like SVM and Random Forest further boosts the system's robustness and accuracy, making it highly suitable for real-world clinical applications where early and precise detection of Alzheimer’s disease is critical.

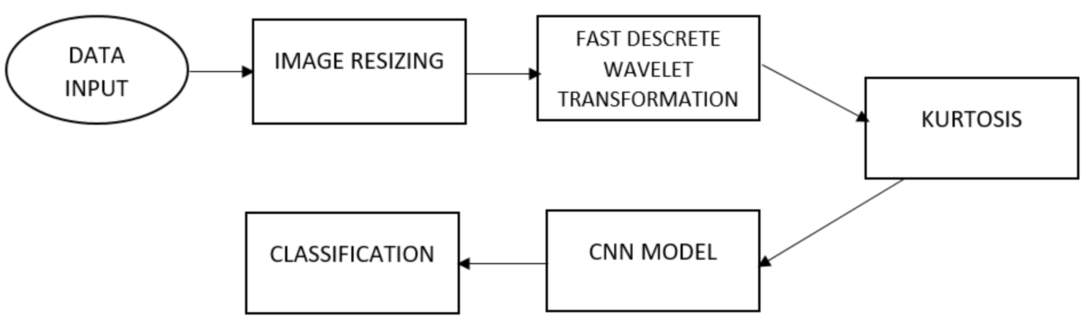

Fig 4.1 Overview of proposed system

5. SYSTEM REQUIREMENT

5.1 Hardware Requirements

- System Type : intel®core™i3-7500UCPU@2.40gh
- Cache memory : 4MB(Megabyte)
- RAM : 8GB (gigabyte)
- Hard Disk : 4GB

5.2 Software Requirements

- Operating System : Windows 11, 64-bit Operating System
- Coding Language : Python
- Python distribution : Google Colab, Flask
- Browser : Any Latest Browser like Chrome

6. SYSTEM ANALYSIS

6.1 Scope of the project

Medical Imaging Focus:

This research primarily focuses on MRI-based medical imaging for the early diagnosis of Alzheimer's disease. While the methodologies developed may have broader applications in medical diagnostics, MRI scans serve as a specific case study due to their effectiveness in detecting neurological disorders.

Integration of Science and Technology:

The project integrates Deep Learning and wavelet transforms (DeepWaveMRI) to enhance MRI image analysis. The combination of computer vision and medical science aims to improve diagnostic accuracy, making it a vital contribution to the field of healthcare technology.

Enhanced Diagnostic Accuracy:

The study aims to improve early detection and classification of Alzheimer's stages through advanced Machine Learning techniques. By leveraging CNN-based models, the research seeks to minimize misdiagnosis, enabling more effective patient management and treatment planning.

Impact on Healthcare and Research:

The scope extends to evaluating the implications of AI-assisted diagnostics in healthcare. The project aims to enhance the reliability of automated detection systems, providing valuable insights for clinicians, radiologists, and researchers in the medical domain.

6.2 Analysis

The proposed approach utilizes a Deep Learning-based Convolutional Neural Network (CNN) model integrated with Wavelet Transform (DeepWaveMRI) to classify Alzheimer’s disease stages using MRI scans. The dataset, sourced from Kaggle and titled "MRI dataset," comprises 6,400 MRI images categorized into four classes: Non-Demented (ND), Very Mild Demented (VMD), Mild Demented (MID), and Moderate Demented (MOD) as shown in Fig 6.1. The Fig 6.2 shows the distribution of classes in the given categories. These images, obtained from 200 subjects, include 32 horizontal brain slices per subject and are resized to a uniform resolution of 208×208 pixels for consistency. The dataset is pre-processed using normalization and augmentation techniques to enhance model generalization. The CNN model is trained and evaluated with accuracy and loss metrics, while a confusion matrix provides insights into misclassifications. The combination of CNN and wavelet transformation improves feature extraction, leading to higher classification accuracy.

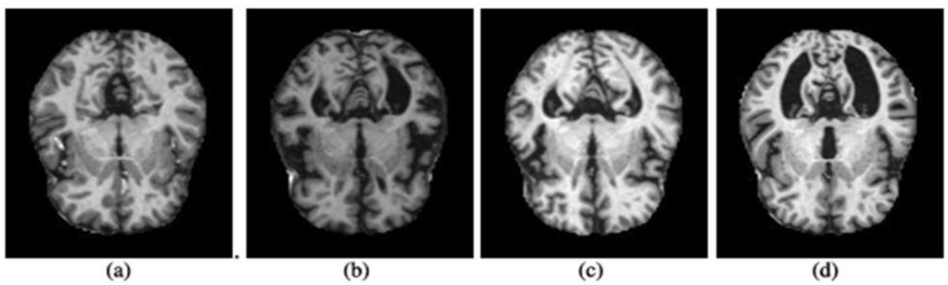

Fig: 6.1 Examples of brain MRI images from four different classes, i.e.: (a) ND, (b) VMD, (c) MID, and (d) MOD

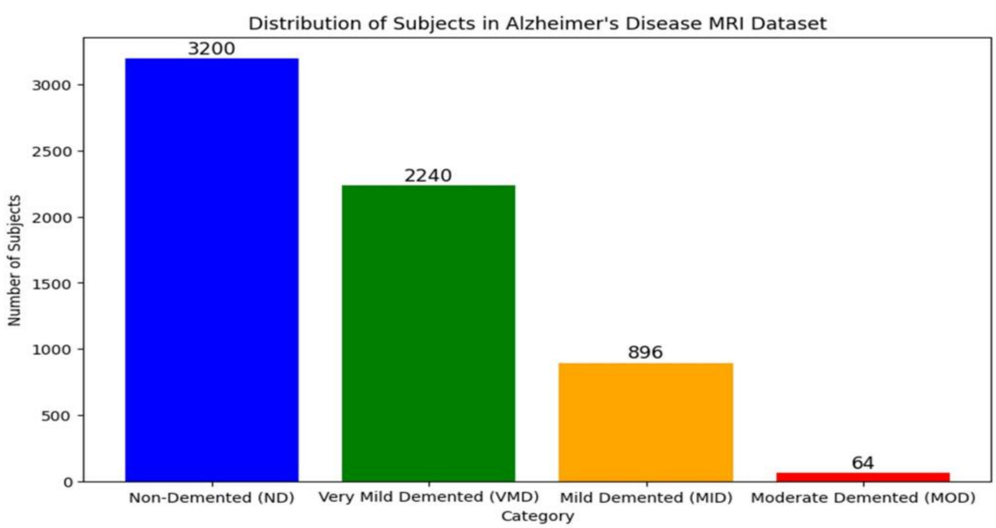

Fig 6.2 Graphical representation of dataset categories

6.3 Data Pre-processing

Image Resizing:

To ensure uniformity across the dataset, all MRI scans are resized to a standard resolution of 208×208 pixels. Since medical images can vary in size and aspect ratio, resizing helps maintain consistency and reduces computational complexity. This step ensures that the Convolutional Neural Network (CNN) receives images of the same dimensions, allowing for efficient processing and better feature extraction.

Normalization:

After resizing, normalization is applied to scale pixel values to the range [0,1] by dividing each pixel intensity by 255. This transformation ensures that the model receives input data in a consistent format, preventing numerical instability and improving convergence during training. Normalization also helps mitigate variations in image intensity caused by differences in MRI scan acquisition settings.

Fast Discrete Wavelet Transform (FDWT):

To enhance feature extraction, FDWT is applied to decompose MRI images into multiple frequency sub-bands. This technique captures fine structural details by separating low-frequency components (which represent the overall structure) from high-frequency components (which highlight edges and textures). By breaking down the images into different resolution levels, FDWT allows the model to focus on significant features relevant to Alzheimer's disease classification as shown in Fig 6.3.

Kurtosis-Based Denoising:

Kurtosis-based denoising is used to remove high-frequency noise while preserving essential brain structures in MRI scans. This technique measures the distribution of pixel intensities and helps identify outliers that may indicate noise as shown in Fig 6.4. By filtering out irrelevant variations, kurtosis-based denoising enhances image clarity, ensuring that the model learns meaningful patterns rather than noise-induced artifacts.

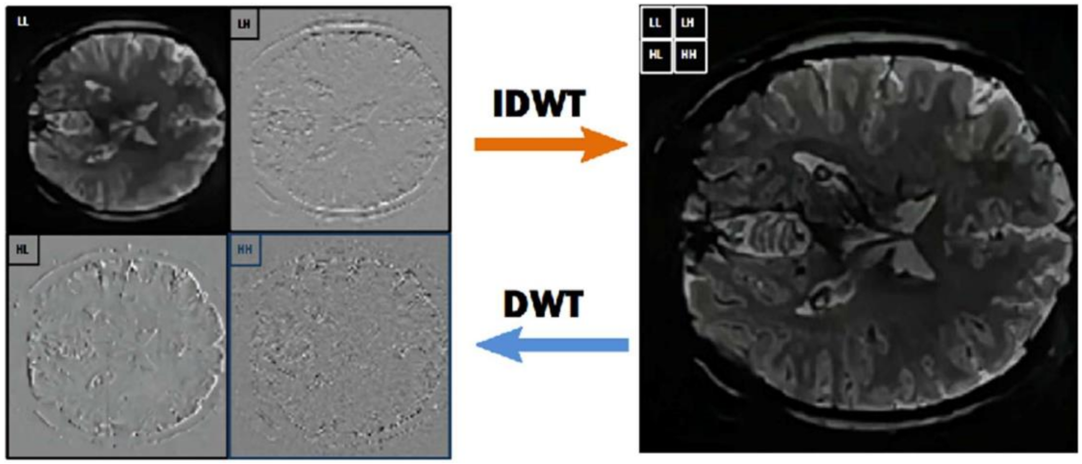

Fig 6.3 Applying FDWT on the pre-processed images

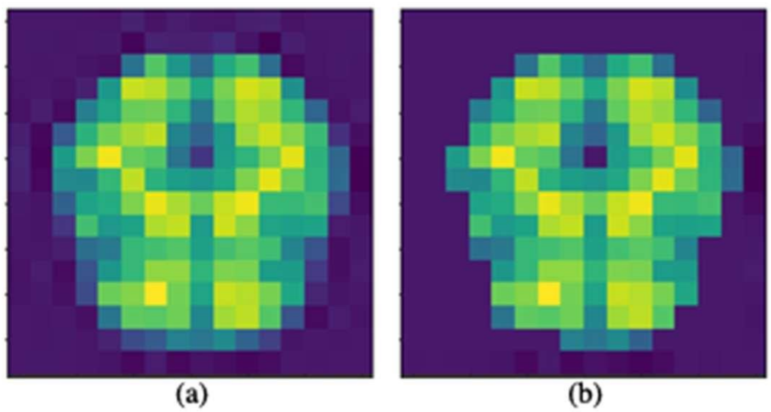

Fig 6.4 (a) Before kurtosis thresholding
(b) After kurtosis thresholding

6.4 Feature Extraction

Wavelet-Based Features

The Fast Discrete Wavelet Transform (FDWT) decomposes MRI images into multiple frequency sub-bands, capturing fine details in different orientations—horizontal, vertical, and diagonal. This process enhances the ability to detect subtle changes in brain structure, such as early signs of atrophy, which are often missed by traditional feature extraction techniques.

Intensity and Contrast Analysis

Alzheimer's disease progression is frequently associated with changes in brain texture and contrast levels. The model analyses pixel intensity variations across MRI scans to detect abnormalities in gray and white matter distribution. Identifying these variations helps in the early detection of structural degeneration, which is crucial for accurate diagnosis.

Edge and Boundary Detection

The feature extraction process also focuses on edges and boundaries within brain structures, identifying irregularities in key regions such as the hippocampus and cerebral cortex. These areas are critical in Alzheimer's diagnosis, as early neurodegeneration often begins here. Detecting these changes aids in distinguishing between different stages of Alzheimer's disease.

6.5 Model Building

Model building in the context of Deep Learning refers to the process of designing and constructing neural network architectures to solve specific tasks such as classification, regression, or generation. Deep Learning models typically consist of multiple layers of neurons organized in a hierarchical fashion, enabling the model to learn intricate patterns and representations from the data given in the Fig 6.5.

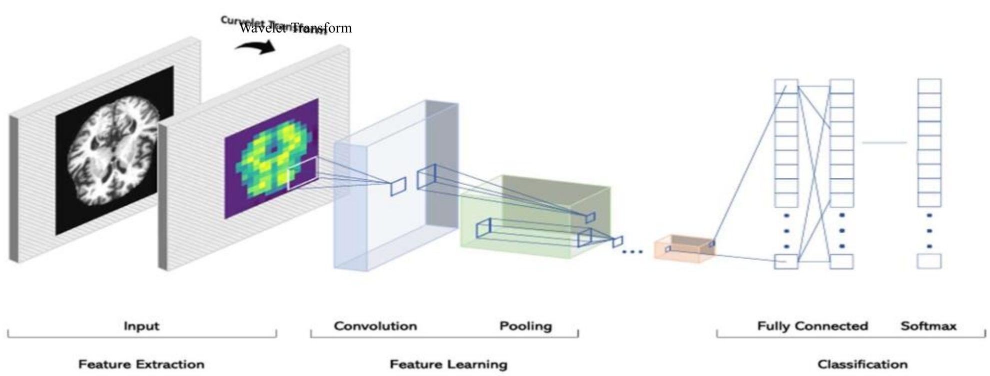

Fig 6.5 Overview of the model

CNN:

A Convolutional Neural Network (CNN) [16] is a type of Deep Learning model designed to process and analyze visual data, particularly images as shown in Fig 6.6. CNNs are inspired by the structure and function of the human visual system and are widely used in computer vision tasks such as image classification, object detection, facial recognition, and more.

CNNs consist of several layers that automatically detect and learn features from images, such as edges, textures, and patterns. These features are learned at different levels of abstraction as the image passes through multiple layers.

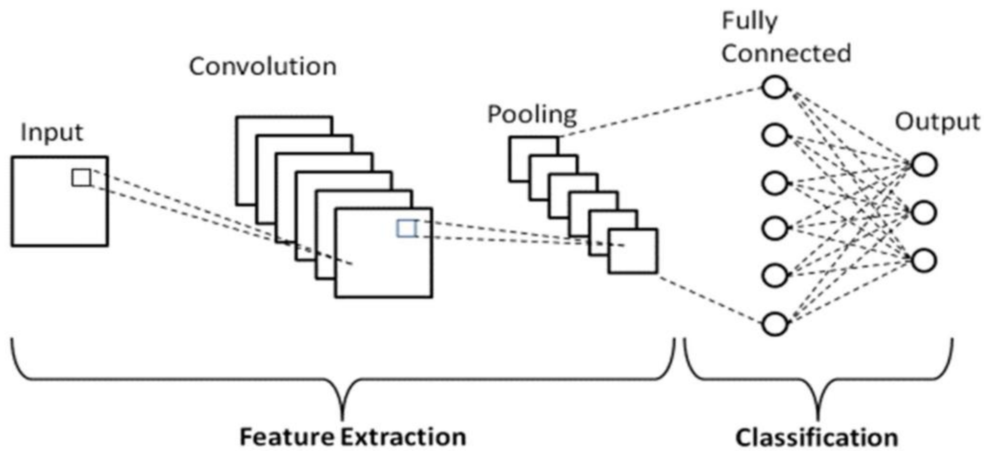

Fig 6.6 Model Architecture of CNN

Layer Types of CNN:

- Input Layer: The Input Layer is the first layer of a neural network. It is responsible for receiving the raw data, such as an image, text, or any other type of input, and passing it on to the next layers for further processing.
- Convolutional Layers: Apply convolution operations to extract features (e.g. edges, shapes) as given in Fig 6.7. The convolutional layer is the fundamental building block of CNN. It applies a set of learnable filters to the input data, typically an image, to extract various features. Each filter convolves across the input, computing dot products between the filter weights and the input at every position.

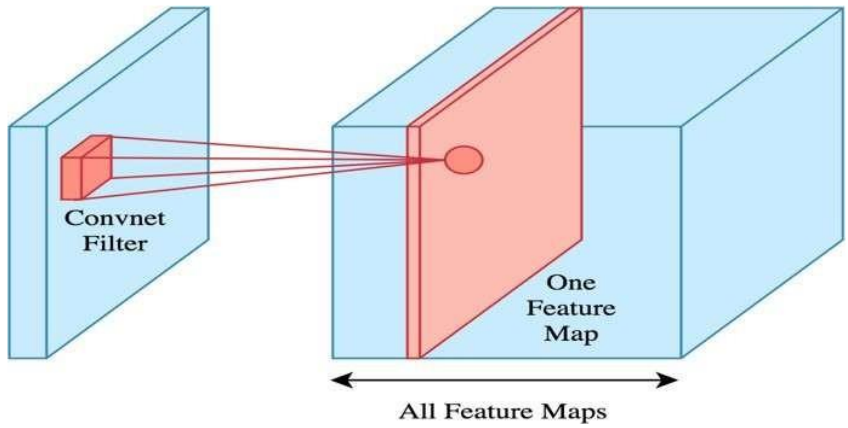

Fig: 6.7 Convolutional layer

- Max Pooling Layers: Max pooling is used to reduce the spatial size of the representation, improving computation efficiency and reducing over fitting as shown in Fig 6.8.

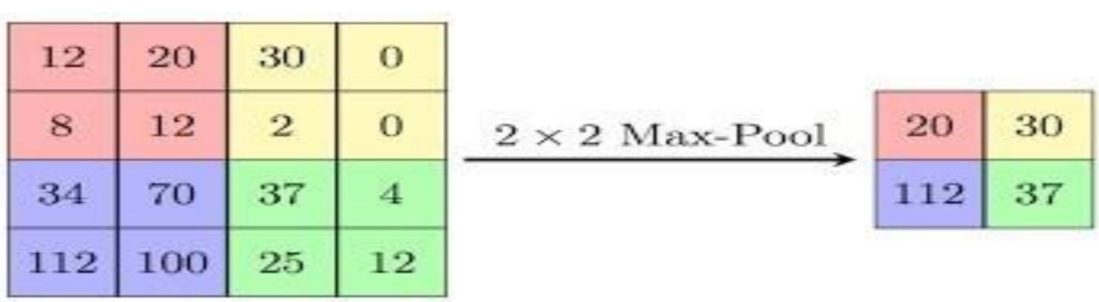

Fig: 6.8 Max-Pooling layer

- Dropout Layers: Often used to prevent over fitting by randomly dropping a proportion of neurons during training. A Dropout Layer is a regularization technique used in neural networks to prevent over fitting. Over fitting occurs when a model learns the noise and fluctuations in the training data rather than the underlying patterns, resulting in poor generalization to new, unseen data. The dropout layer helps mitigate this by randomly "dropping" or deactivating a proportion of neurons during training.
- Activation layer: An Activation Layer is a crucial component of neural networks, responsible for introducing non-linearity into the network. Without activation functions, the neural network would essentially be equivalent to a linear regression model, regardless of the number of layers. Activation functions allow the network to

learn and approximate complex patterns in the data, enabling it to solve tasks such as classification, regression, and more as shown in Fig 6.9.

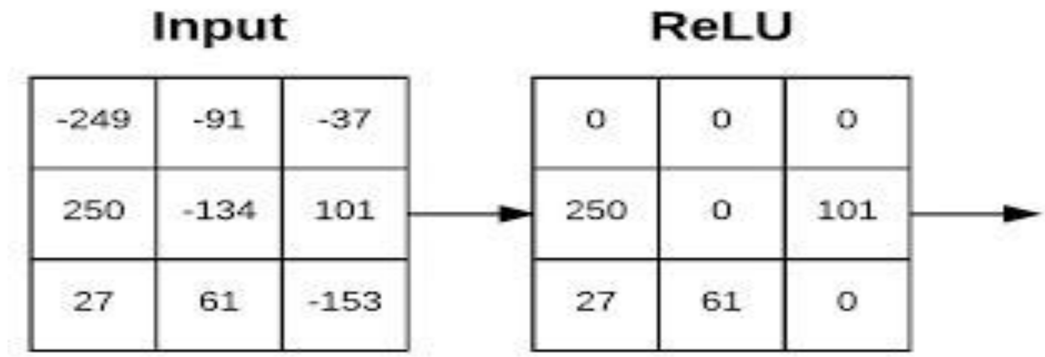

Fig: 6.9 Activation Layer

- Fully Connected Layer: A Fully Connected Layer, also known as a Dense Layer, is a type of neural network layer where each neuron is connected to every neuron in the previous layer. It is one of the most fundamental layers in Deep Learning models and is typically used in the final stages of a network for tasks like classification or regression.
- Output Layer: The Output Layer is the final layer in a neural network that produces the results based on the processed data. The output layer is responsible for delivering the prediction or decision made by the model after processing the input data through multiple hidden layers. The main purpose of the output layer is to produce the network's final prediction. The design of the output layer directly depends on the task the network is designed for, such as classification, regression, or others.

6.6 CLASSIFICATION REPORT

In classification tasks, a classification report provides a detailed evaluation of the performance of a model. It typically includes several key metrics that assess how well the model [17] is performing across different classes. The classification report is especially useful for understanding how a model is handling each class individually, which can be crucial when dealing with imbalanced datasets or multi-class problems.

- Precision: Measures the proportion of correctly predicted positive instances out of all predicted positive instances.
- Recall: Measures the proportion of correctly predicted positive instances out of all actual positive instances.
- F1-Score: The harmonic mean of precision and recall, providing a balanced measure between the two.
- Accuracy: The overall proportion of correct predictions (both true positives and true negatives) out of all predictions.
- Support: Support is the number of actual occurrences of each class in the dataset. It helps you understand how many instances there are for each class in the dataset.

6.7 CONFUSION MATRIX

Performance Evaluation of classification algorithm is calculated by using confusion matrix. Confusion matrix is a table describes performance based on set of test data for which true values are known as shown in Fig 6.10. Performance is calculated by considering actual and predicted class [18] . A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which true values are known.

A true positive (tp) is a result where the model predicts the positive class correctly. Similarly, a true negative (tn) is an outcome where the model correctly predicts the negative class. A false positive (fp) is an outcome where the model incorrectly predicts the positive class. Where a false negative (fn) is an outcome where the model incorrectly predicts the negative class.

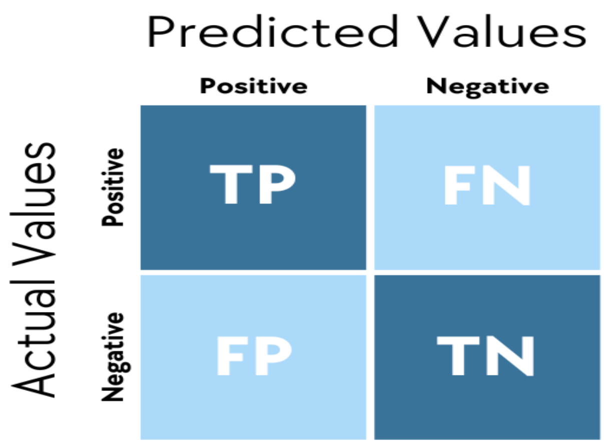

Fig 6.10 Confusion Matrix

7. DESIGN

The proposed system shown in Fig 7.1 aims to classify the four stages of Alzheimer's disease Non-Demented, Moderate Demented, Mild Demented, and Very Mild Demented using a Convolutional Neural Network (CNN). The workflow begins with a well-curated dataset, divided into training and testing subsets, where the training data is used to teach the CNN model patterns and features, and the testing data evaluates its performance. The CNN model processes the input images to extract important features and classify them into the respective stages of Alzheimer's disease. The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score to ensure its reliability. This system offers a time-efficient and accurate solution for automated Alzheimer's detection, with the potential to improve through larger datasets, advanced augmentation techniques, and multi-modal data [19] integration.

Future developments could include deploying the model as a web application or mobile tool for real-time diagnostics, enabling scalable and accessible healthcare solutions the proposed system leverages the power of Deep Learning to address the critical challenge of early and accurate detection of Alzheimer’s disease. By automating the classification process, it minimizes the need for manual intervention, thus reducing the likelihood [20] of human error.

The use of CNNs ensures that intricate patterns and subtle differences in medical images are effectively identified. Furthermore, this approach not only enhances diagnostic efficiency but also provides a scalable solution that can be adapted to other medical imaging tasks. As the system evolves, it holds the potential to revolutionize healthcare diagnostics by making advanced tools accessible to a broader audience, including rural and underserved regions.

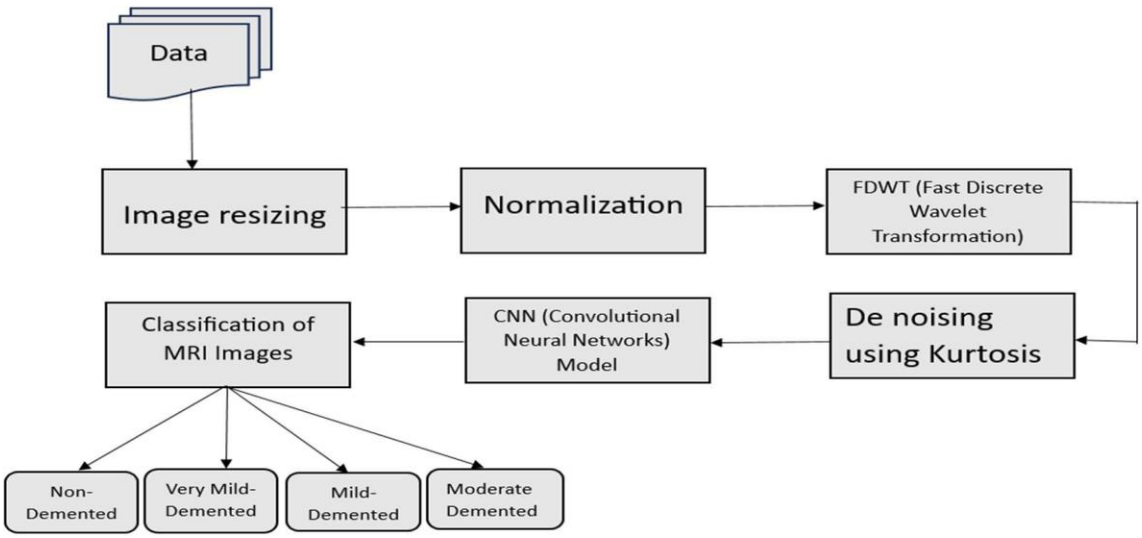

Fig 7.1 Design overview

8. IMPLEMENTATION

#mounting google drive
from google.colab import drive drive.mount("/content/drive")

 # Pre-Processing the images

```
!pip install pillow
import os from PIL
import Image
```

 # resize images in folder

```
import os

from PIL import Image

def resize_images_in_folder(folder_path, target_size):

    if not os.path.exists(folder_path):

        print(f"Error: Folder '{folder_path}' does not exist.")

        return

    print(f"Processing folder: {folder_path}")

    for root, dirs, files in os.walk(folder_path):  # Recursively walk through subdirectories

        for filename in files:

            file_path = os.path.join(root, filename)

            try:

                with Image.open(file_path) as img:

                    print(f"Original size of {filename}: {img.size}")

                    # Resize the image using the updated LANCZOS method

                    img = img.resize(target_size, Image.Resampling.LANCZOS)

                    img.save(file_path, format=img.format)  # Save in the same format
```

```
            print(f"Resized {filename} to {target_size}")
        except Exception as e:
            print(f"Error processing file {file_path}: {e}")
```

 # Example usage

```
base_folder = "/content/drive/MyDrive/MRIdataset/Dataset"
target_size = (208, 208)  # Set your target size
resize_images_in_folder(base_folder, target_size)
import os
from PIL import Image
# check_folder_confidence
def check_folder_confidence(folder_path, target_size):
    folder_confidence = {}
    for root, dirs, files in os.walk(folder_path):  # Traverse all subdirectories
        for dir_name in dirs:
            # Skip folders that are not part of the dataset, e.g., .ipynb_checkpoints
            if dir_name.startswith("."):
                continue
            folder_dir_path = os.path.join(root, dir_name)
            total_images = 0
            correctly_resized = 0
            for filename in os.listdir(folder_dir_path):
                file_path = os.path.join(folder_dir_path, filename)
                if os.path.isfile(file_path):
                    total_images += 1
```

```
            try:
                with Image.open(file_path) as img:
                    # Check if the image is correctly resized
                    if img.size == target_size:
                        correctly_resized += 1
            except Exception as e:
                print(f"Error processing file {file_path}: {e}")
        # Calculate the confidence for this folder
        if total_images > 0:
            confidence = (correctly_resized / total_images) * 100
        else:
            confidence = 0
        folder_confidence[dir_name] = confidence
        print(f"Folder: {dir_name}, Confidence: {confidence:.2f}%")
    return folder_confidence
# Example usage
base_folder = "/content/drive/MyDrive/MRIdataset/Dataset"
target_size = (208, 208)
folder_confidence = check_folder_confidence(base_folder, target_size)
```

Applying fdwt

```
!pip install PyWavelets
import os
import pywt
```

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
def apply_fdwt_and_display(folder_path, wavelet='haar', max_images_to_display=3):
    for root, dirs, files in os.walk(folder_path):  # Traverse folders
        for dir_name in dirs:
            folder_dir_path = os.path.join(root, dir_name)
            processed_images = 0  # Track number of displayed images
            for filename in os.listdir(folder_dir_path):
                file_path = os.path.join(folder_dir_path, filename)
                # Skip images that already have been transformed (contain 'transformed_')
                if 'transformed_' in filename:
                    continue
                if os.path.isfile(file_path):
                    try:
                        with Image.open(file_path) as img:
                            # Resize and process image
                            img = img.resize(target_size).convert("L")
                            img_array = np.array(img)
                            # Apply FDWT
                            coeffs2 = pywt.dwt2(img_array, wavelet)
                            LL, (LH, HL, HH) = coeffs2
                            # Save transformed image (using LL coefficients)
                            LL = np.clip(LL, 0, 255).astype(np.uint8)
```

```
                transformed_image = Image.fromarray(LL, mode='L')
                transformed_image.save(os.path.join(folder_dir_path,
f"transformed_{filename}"))
                if processed_images < max_images_to_display:
                    # Display the first few images
                    fig, axes = plt.subplots(2, 2, figsize=(10, 10))
                    axes[0, 0].imshow(LL, cmap='gray')
                    axes[0, 0].set_title('LL (Approximation)')
                    axes[0, 1].imshow(LH, cmap='gray')
                    axes[0, 1].set_title('LH (Horizontal Detail)')
                    axes[1, 0].imshow(HL, cmap='gray')
                    axes[1, 0].set_title('HL (Vertical Detail)')
                    axes[1, 1].imshow(HH, cmap='gray')
                    axes[1, 1].set_title('HH (Diagonal Detail)')
                    plt.show()
                    processed_images += 1
                else:
                    # Print for other images
                    print(f"FDWT applied to {filename}")
            except Exception as e:
                print(f"Error processing file {file_path}: {e}")
# Example usage
base_folder = "/content/drive/MyDrive/MRIdataset/Dataset"
apply_fdwt_and_display(base_folder, wavelet='haar', max_images_to_display=3)
```

```
import os
import pywt
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
def apply_fdwt_and_display(folder_path, wavelet='haar', max_images_to_display=3):
    folder_confidences = {}  # To store confidence for each folder
    for root, dirs, files in os.walk(folder_path):  # Traverse folders
        for dir_name in dirs:
            folder_dir_path = os.path.join(root, dir_name)
            total_images = 0
            transformed_images = 0
            processed_images = 0  # Track number of displayed images
            for filename in os.listdir(folder_dir_path):
                file_path = os.path.join(folder_dir_path, filename)
                # Skip images that already have been transformed (contain 'transformed_')
                if 'transformed_' in filename:
                    continue
                if os.path.isfile(file_path):
                    total_images += 1  # Increment total images in folder
                    try:
                        with Image.open(file_path) as img:
                            # Resize and process image
                            img = img.resize((208, 208)).convert("L")
```

```
                img_array = np.array(img)
                # Apply FDWT
                coeffs2 = pywt.dwt2(img_array, wavelet)
                LL, (LH, HL, HH) = coeffs2
                # Save transformed image (using LL coefficients)
                LL = np.clip(LL, 0, 255).astype(np.uint8)  # Clip to valid range
                transformed_image = Image.fromarray(LL, mode='L')
                transformed_image.save(os.path.join(folder_dir_path,
f"transformed_{filename}"))
                transformed_images += 1  # Increment transformed images count
                if processed_images < max_images_to_display:
                    # Display the first few images
                    fig, axes = plt.subplots(2, 2, figsize=(10, 10))
                    axes[0, 0].imshow(LL, cmap='gray')
                    axes[0, 0].set_title('LL (Approximation)')
                    axes[0, 1].imshow(LH, cmap='gray')
                    axes[0, 1].set_title('LH (Horizontal Detail)')
                    axes[1, 0].imshow(HL, cmap='gray')
                    axes[1, 0].set_title('HL (Vertical Detail)')
                    axes[1, 1].imshow(HH, cmap='gray')
                    axes[1, 1].set_title('HH (Diagonal Detail)')
                    plt.show()
                    processed_images += 1
                else:
```

```
                print(f"FDWT applied to {filename}")
            except Exception as e:
              print(f"Error processing file {file_path}: {e}")
        # Calculate confidence for this folder
        confidence = (transformed_images / total_images) * 100
        folder_confidences[dir_name] = confidence
        print(f"Folder: {dir_name}, Confidence: {confidence:.2f}%")
    return folder_confidences
# Example usage
base_folder = "/content/drive/MyDrive/MRIdataset/Dataset"
folder_confidences = apply_fdwt_and_display(base_folder, wavelet='haar',
max_images_to_display=3)
# check_fdwt_confidence
import os
def check_fdwt_confidence(folder_path):
    for root, dirs, files in os.walk(folder_path):  # Traverse folders
     for dir_name in dirs:
        folder_dir_path = os.path.join(root, dir_name)
        total_images = 0
        fdwt_images = 0
        for filename in os.listdir(folder_dir_path):
          file_path = os.path.join(folder_dir_path, filename)
          if os.path.isfile(file_path):
            total_images += 1  # Increment total images in folder
```

```
                    # Check if the image is FDWT-applied (starts with 'transformed_')
                    if 'transformed_' in filename:
                        fdwt_images += 1  # Increment FDWT-applied image count
            # Calculate confidence for this folder (FDWT-applied images out of total images)
            confidence = (fdwt_images / total_images) * 100 if total_images > 0 else 0
            folder_confidences[dir_name] = confidence
            print(f"Folder: {dir_name}, FDWT Confidence: {confidence:.2f}%")
    return folder_confidences
# Example usage
base_folder = "/content/drive/MyDrive/MRIdataset/Dataset"  # Replace with your folder path
folder_confidences = check_fdwt_confidence(base_folder)
```

Applying Kurtosis

```
!pip install opencv-python scipy matplotlib
import os
import pywt
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from scipy.stats import kurtosis
def apply_fdwt_and_kurtosis(folder_path, wavelet='haar', max_images_to_display=3):
    for root, dirs, files in os.walk(folder_path):  # Traverse folders
```

```
        for dir_name in dirs:
            folder_dir_path = os.path.join(root, dir_name)
            processed_images = 0  # Track number of displayed images
            for filename in os.listdir(folder_dir_path):
                file_path = os.path.join(folder_dir_path, filename)
                if 'transformed_' in filename:
                    continue
                if os.path.isfile(file_path):
                    try:
                        with Image.open(file_path) as img:
                            # Resize and process image
                            img = img.resize((208, 208)).convert("L")
                            img_array = np.array(img)
                            # Apply FDWT
                            coeffs2 = pywt.dwt2(img_array, wavelet)
                            LL, (LH, HL, HH) = coeffs2
                            # Save transformed image (using LL coefficients)
                            LL = np.clip(LL, 0, 255).astype(np.uint8)  # Clip to valid range
                            transformed_image = Image.fromarray(LL, mode='L')
                            transformed_image.save(os.path.join(folder_dir_path,
f"transformed_{filename}"))
                            # Display the FDWT components for the first few images
                            if processed_images < max_images_to_display:
                                fig, axes = plt.subplots(2, 2, figsize=(10, 10))
```

```
            axes[0, 0].imshow(LL, cmap='gray')
            axes[0, 0].set_title('LL (Approximation)')
            axes[0, 1].imshow(LH, cmap='gray')
            axes[0, 1].set_title('LH (Horizontal Detail)')
            axes[1, 0].imshow(HL, cmap='gray')
            axes[1, 0].set_title('HL (Vertical Detail)')
            axes[1, 1].imshow(HH, cmap='gray')
            axes[1, 1].set_title('HH (Diagonal Detail)')
            plt.show()
            processed_images += 1
          # Calculate and print kurtosis of LL coefficients
          kurtosis_value = kurtosis(LL.flatten())
          print(f"Kurtosis for {filename}: {kurtosis_value:.2f}")
      except Exception as e:
        print(f"Error processing file {file_path}: {e}")
# Example usage
base_folder = "/content/drive/MyDrive/MRIdataset/Dataset"
apply_fdwt_and_kurtosis(base_folder, wavelet='haar', max_images_to_display=3)
```

Using CNN model

```
import os
import cv2
import numpy as np
import tensorflow as tf
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from google.colab import drive
# Mount Google Drive to save and access the model
drive.mount('/content/drive')
# Define the main folder containing the subfolders
main_folder = '/content/drive/MyDrive/MRIdataset/Dataset'
# Define the dataset
image_size = (208, 208)
num_classes = 4
# Initialize lists to hold images and labels
images = []
labels = []
# Load and preprocess images from subfolders
for folder in os.listdir(main_folder):
    folder_path = os.path.join(main_folder, folder)
    if os.path.isdir(folder_path):
        label = folder  # Use folder name as label
        for filename in os.listdir(folder_path):
            file_path = os.path.join(folder_path, filename)
            if os.path.isfile(file_path):
```

```
        try:
            # Read and preprocess the image
            image = cv2.imread(file_path, cv2.IMREAD_GRAYSCALE)
            if image is None:
                continue
            image = cv2.resize(image, image_size)
            image = image.astype('float32') / 255.0  # Normalize pixel values
            # Append image and label to lists
            images.append(image)
            labels.append(label)
        except Exception as e:
            print(f"Failed to process {file_path}: {e}")
# Convert lists to numpy arrays
images = np.array(images)
labels = np.array(labels)
# Encode labels as integers
label_to_index = {label: index for index, label in enumerate(np.unique(labels))}
index_to_label = {index: label for label, index in label_to_index.items()}
labels = np.array([label_to_index[label] for label in labels])
# Reshape images to include the channel dimension (for grayscale: 1 channel)
images = images.reshape(images.shape[0], image_size[0], image_size[1], 1)
# One-hot encode the labels
labels = to_categorical(labels, num_classes)
# Split data into training and testing sets
```

```
X_train, X_test, y_train, y_test = train_test_split(images, labels, test_size=0.06,
random_state=32)  # actual random_state=42

print("Data preprocessing complete.")

print(f"Training data shape: {X_train.shape}")

print(f"Testing data shape: {X_test.shape}")

# Define the CNN model

model = Sequential([

    Conv2D(32, (3, 3), activation='relu', input_shape=(image_size[0], image_size[1], 1)),

    MaxPooling2D(pool_size=(2, 2)),

    Conv2D(64, (3, 3), activation='relu'),

    MaxPooling2D(pool_size=(2, 2)),

    Flatten(),

    Dense(128, activation='relu'),

    Dropout(0.5),

    Dense(num_classes, activation='softmax')

])

# Compile the model

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

# Train the model and save the training history

history = model.fit(X_train, y_train, epochs=50, batch_size=26, validation_data=(X_test,
y_test))#b_s=32,26

# Evaluate the model

test_loss, test_accuracy = model.evaluate(X_test, y_test)

print(f"Test Accuracy: {test_accuracy}")

# Save the trained CNN model to a file in Google Drive
```

```
model_filename = '/content/drive/MyDrive/3test_model.h5'
model.save(model_filename)
print(f"Model saved at {model_filename}")
# Plot training & validation accuracy and loss
plt.figure(figsize=(12, 5))
# Accuracy plot
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Val Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Accuracy over Epochs')
plt.legend()
# Loss plot
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Loss over Epochs')
plt.legend()
plt.tight_layout()
plt.show()
```

Plot some test images with their predicted and true labels

```
plt.figure(figsize=(10, 10))

for i in range(5):

    idx = np.random.randint(0, len(X_test))

    img = X_test[idx].reshape(image_size[0], image_size[1])  # Remove channel
dimension for display

    true_label = np.argmax(y_test[idx])

    pred_label = np.argmax(model.predict(np.expand_dims(X_test[idx], axis=0)))

    plt.subplot(1, 5, i+1)

    plt.imshow(img, cmap='gray')

    plt.title(f"True: {index_to_label[true_label]}\nPred: {index_to_label[pred_label]}")

    plt.axis('off')

plt.tight_layout()

plt.show()
```

Classification Report

```
from sklearn.metrics import classification_report

# Get predictions for the test set

y_pred = model.predict(X_test)

y_pred_classes = np.argmax(y_pred, axis=1)  # Convert predicted probabilities to class
labels

y_true_classes = np.argmax(y_test, axis=1)  # Convert one-hot encoded true labels to
class labels

# Print classification report

print("Classification Report:")
```

```
print(classification_report(y_true_classes, y_pred_classes,
target_names=list(index_to_label.values())))
```

Getting Confusion and Correlation matrix

```
from sklearn.metrics import confusion_matrix

import seaborn as sns

import pandas as pd

# Get predictions for the test set

y_pred = model.predict(X_test)

y_pred_classes = np.argmax(y_pred, axis=1)

y_true_classes = np.argmax(y_test, axis=1)

# Confusion Matrix

conf_matrix = confusion_matrix(y_true_classes, y_pred_classes)

# Plot Confusion Matrix

plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=list(index_to_label.values()), yticklabels=list(index_to_label.values()))

plt.title('Confusion Matrix')

plt.ylabel('True Label')

plt.xlabel('Predicted Label')

plt.show()

# Correlation Matrix

# Convert the predictions and true labels to Data Frame for correlation

df = pd.DataFrame({'True': y_true_classes, 'Predicted': y_pred_classes})
```

```python
corr_matrix = df.corr()


# Plot Correlation Matrix
plt.figure(figsize=(6, 5))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', cbar=True, square=True)
plt.title('Correlation Matrix')
plt.show()
```

#FRONTEND

#form.html

#A symptom checker form that collects user inputs on age, headache frequency, memory issues, confusion, and other symptoms.

```html
<body>
  <div class="container">
    <h1>Symptom Checker</h1>
    <form id="symptomForm" action="/submit_form" method="post">
            <label for="age">Age:</label>
      <input type="number" id="age" name="age" placeholder="Enter your age" required>
      <label for="headache">Do you experience frequent headaches?</label>
      <select id="headache" name="headache">
        <option value="" disabled selected>Choose</option>
        <option value="yes">Yes</option>
        <option value="no">No</option>
        <option value="yes">Sometimes</option>
```

```
</select>

<label for="memory">Do you have memory loss issues?</label>

<select id="memory" name="memory">

  <option value="" disabled selected>Choose</option>

  <option value="yes">Yes</option>

  <option value="no">No</option>

  <option value="yes">Not Often</option>

</select>

<label for="confusion">Do you feel confused often?</label>

<select id="confusion" name="confusion">

  <option value="" disabled selected>Choose</option>

  <option value="yes">Yes</option>

  <option value="no">No</option>

  <option value="yes">Sometimes</option>

</select>

<div class="checkbox-group">

  <label><input type="checkbox" name="symptom" value="Memory Loss">
Memory Loss</label>

<label><input type="checkbox" name="symptom" value="Difficulty Planning">
Difficulty Planning</label>

  <label><input type="checkbox" name="symptom" value="Confusion with Time">
Confusion with Time</label>

  <label><input type="checkbox" name="symptom" value="Misplacing Items">
Misplacing Items</label>

  <label><input type="checkbox" name="symptom" value="Mood Changes"> Mood
Changes</label>
```

```
        <label><input type="checkbox" name="symptom" value="Difficulty Speaking"> Difficulty Speaking</label>

        <label><input type="checkbox" name="symptom" value="Withdrawal from Social Activities"> Withdrawal from Social Activities</label>

      </div>

      <button type="submit" onclick="showPopup(event)">Continue</button>

      <!-- Message will be displayed here -->

      <div id="message"></div>

      <div id="messageDiv" style="display: none; margin-top: 20px; font-size: 16px; color: #333;"></div>

    </form>

  </div>

</body>

</html>
```

 # app.py

#A Flask-based web application for Alzheimer's disease prediction and MRI validation using Deep Learning models.

```
import tensorflow as tf

import numpy as np

from flask import Flask, render_template, request, redirect, url_for, jsonify

from PIL import Image

import io

app = Flask(_name_)
```

 # Load the TensorFlow Models

```
MODEL_PATH = r"C:\Users\LENOVO\flask_project\model\alzheimer_cnn_model.h5"
```

```
VALIDATION_MODEL_PATH =
r"C:\Users\LENOVO\flask_project\model\validation_model_new.h5"

model = tf.keras.models.load_model(MODEL_PATH)

validation_model = tf.keras.models.load_model(VALIDATION_MODEL_PATH)

# Define class labels

CLASS_LABELS = ["Non-Demented", "Very-Mild Demented", "Mild Demented",
"Moderate Demented"]

#  Function to preprocess the image

def preprocess_image(image, target_size=(208, 104)):

    image = image.resize(target_size)  # Resize to model input size

    image = np.array(image.convert("L")) / 255.0  # Convert to grayscale & normalize

    image = np.expand_dims(image, axis=-1)  # Add channel dimension

    image = np.expand_dims(image, axis=0)  # Add batch dimension

    return image

#  Function to preprocess image for validation

def preprocess_validation_image(image, target_size=(128, 128)):

    image = image.resize(target_size)  # Resize to validation model input size

    image = np.array(image) / 255.0  # Normalize

image = np.expand_dims(image, axis=0)  # Add batch dimension

    return image

# API route to predict image classification

@app.route("/predict", methods=["POST"])

def predict():

    try:

        file = request.files["file"]  # Get uploaded image
```

```
        image = Image.open(io.BytesIO(file.read()))  # Open image
        processed_image = preprocess_image(image)  # Preprocess the image
        # Get model prediction
        output = model.predict(processed_image)
        predicted_class = np.argmax(output)  # Get index of max probability
        confidence = np.max(output)  # Get highest confidence score
          return jsonify({
            "prediction": CLASS_LABELS[predicted_class],
            "confidence": float(confidence)  # Convert NumPy float to Python float
        })
        except Exception as e:
        return jsonify({"error": str(e)})
 # Function to preprocess image for validation
def preprocess_validation_image(image, target_size=(255, 255)):
    image = image.resize(target_size)  # Resize to validation model input size
    image = image.convert("RGB")  # Ensure 3-channel RGB format
    image = np.array(image) / 255.0  # Normalize
    image = np.expand_dims(image, axis=0)  # Add batch dimension
    return image
@app.route("/validate_mri", methods=["POST"])
def validate_mri():
    try:
        file = request.files["file"]
        image = Image.open(io.BytesIO(file.read()))
```

```
        processed_image = preprocess_validation_image(image)


        output = validation_model.predict(processed_image)

        is_brain_mri = output[0][0] < 0.5  # Assuming binary classification

        return jsonify({"is_mri": bool(is_brain_mri)})

    except Exception as e:

        return jsonify({"error": str(e)})
```

 # Route for the symptom checker form

```
@app.route("/")

def symptom_form():

    return render_template("form.html")  # Ensure "form.html" is the symptom checker page
```

 # Route to handle symptom form submission

```
@app.route("/submit_form", methods=["POST"])

def submit_form():

    symptoms = request.form.getlist("symptom")  # Get checked symptom

    if len(symptoms) > 3:

        # More than 3 symptoms → Redirect to validation page

        return redirect(url_for("validate_page"))

    else:

        # 3 or fewer symptoms → Show alert and stay on the form page

        return '''

        <script>

            alert("You're alright! No need to upload an image. Just take care of your health with:\n\n" +
```

```
            "- Staying mentally active\n" +
            "- Eating a balanced diet\n" +
            "- Getting regular exercise");
        window.location.href = "/";
    </script>
    '''
```

 # Route to display the MRI validation page

```
@app.route("/validate")
def validate_page():
    return render_template("validate.html")  # Ensure "validate.html" is the validation page
```

#Route to display the MRI image upload page

```
@app.route("/upload")
def upload_page():
    return render_template("index.html")  # Ensure "index.html" is the upload page
if _name_ == "_main_":
    app.run(debug=True)
```

 # validate.html
#A validation page for checking if an uploaded image is an MRI before proceeding to the upload stage.

```
<body>
    <div class="container">
        <h1>MRI Image Validator</h1>
        <form id="validateForm">
            <input type="file" id="imageInput" accept="image/*" required>
            <button type="submit">Validate Image</button>
```

```
        </form>
        <div id="validationResult"></div>
        <button id="proceedButton" onclick="location.href='/upload'">Proceed to
Upload</button>
    </div>
    <script>
document.getElementById("validateForm").addEventListener("submit", async (e) => {
    e.preventDefault();
      const formData = new FormData();
    const imageInput = document.getElementById("imageInput");
    if (!imageInput.files.length) {
        alert("Please select an image first.");
        return;
    }
    formData.append("file", imageInput.files[0]);
    try {
        const response = await fetch("/validate_mri", {
            method: "POST",
            body: formData,
  });
        const result = await response.json();
        const resultDiv = document.getElementById("validationResult");
        const proceedButton = document.getElementById("proceedButton");
        if (result.is_mri) {
```

```
        resultDiv.style.color = "lightgreen";
        resultDiv.innerHTML = "✅ This is a valid MRI.";
        proceedButton.style.display = "block";
      } else {
        resultDiv.style.color = "red";
        resultDiv.innerHTML = "❌ This is NOT a valid MRI.";
        proceedButton.style.display = "none";
      }
    } catch (error) {
      const resultDiv = document.getElementById("validationResult");
      resultDiv.style.color = "red";
      resultDiv.innerHTML = Error: ${error.message};
    }
  }
```

9. RESULT ANALYSIS

The proposed model, DeepWaveMRI, with a CNN architecture using these pre-processing methods attained a fantastic accuracy rate of 98.96%, indicating that it can be able to effectively differentiate between different types of MRI images. Additionally, it achieved an F1-score of 99.07%, sensitivity of 98.96%, and specificity rate reaching up to 99.74% as shown in Fig 9.2.

Fig 9.1 is a confusion matrix displaying the classification performance of an Alzheimer's disease detection model, showing true versus predicted labels for different dementia stages. Figure 9.3 shows the results of the CNN model on the Alzheimer MRI pre-processed dataset, comparing true and predicted classifications for different dementia stages. Fig 9.4 presents training and validation accuracy and loss curves over epochs, demonstrating the CNN model's learning performance on the Alzheimer MRI dataset.

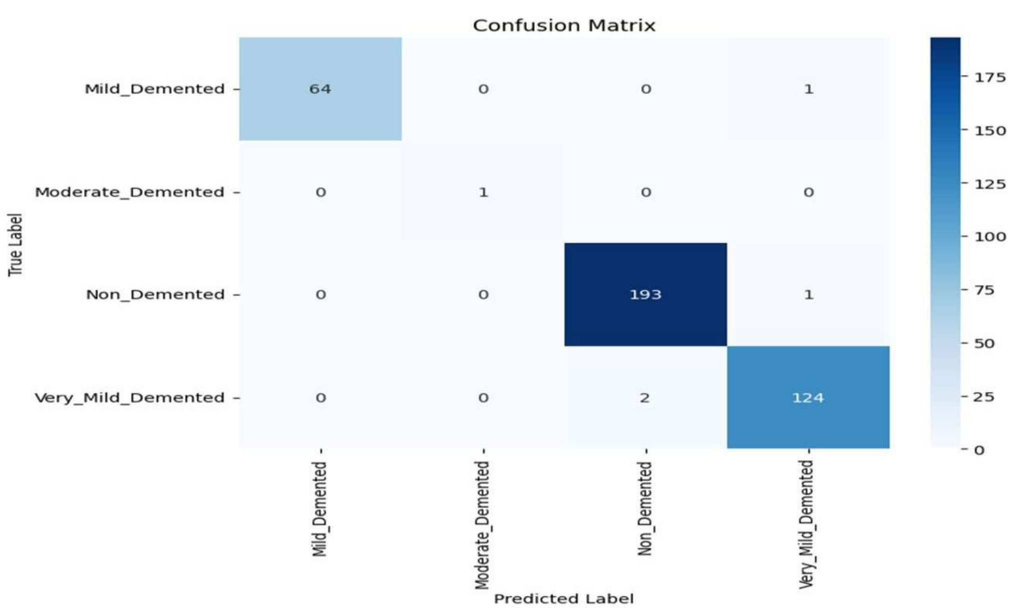

Fig 9.1 Correlation matrix

```
13/13 ━━━━━━━━━━━━━━━━━━━━ 13s 947ms/step
Classification Report:
                    precision    recall  f1-score   support

     Mild_Demented       1.00      0.98      0.99        65
 Moderate_Demented       1.00      1.00      1.00         1
      Non_Demented       0.99      0.99      0.99       194
Very_Mild_Demented       0.98      0.98      0.98       126

          accuracy                           0.99       386
         macro avg       0.99      0.99      0.99       386
      weighted avg       0.99      0.99      0.99       386
```

Fig 9.2 Classification report for CNN model

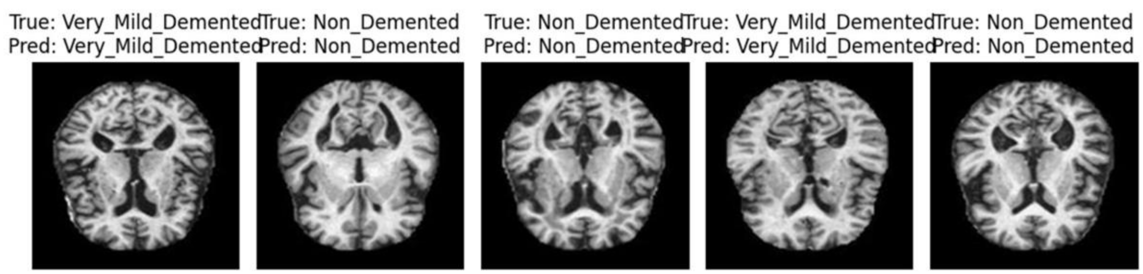

Fig 9.3 Result of CNN model used on Alzheimer MRI Preprocessed Dataset.

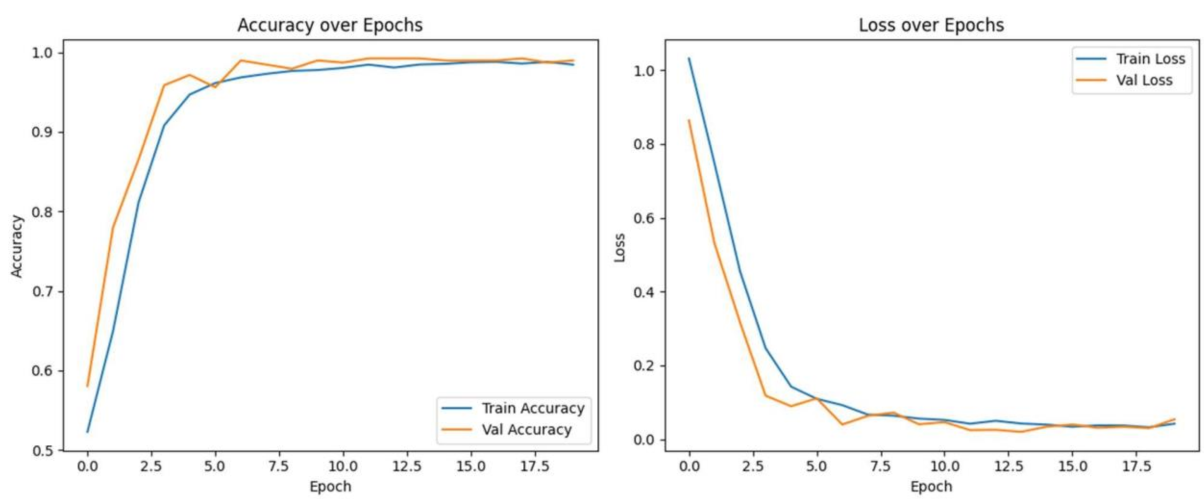

Fig 9.4 Accuracy and Loss over Epochs in CNN

10. TEST CASES

Test case 1: Mild Demented stage

In the Fig 10.1 when the MRI image of the brain is uploaded to predict the Alzheimer's stage, the predicted stage is displaying as "Mild Demented"

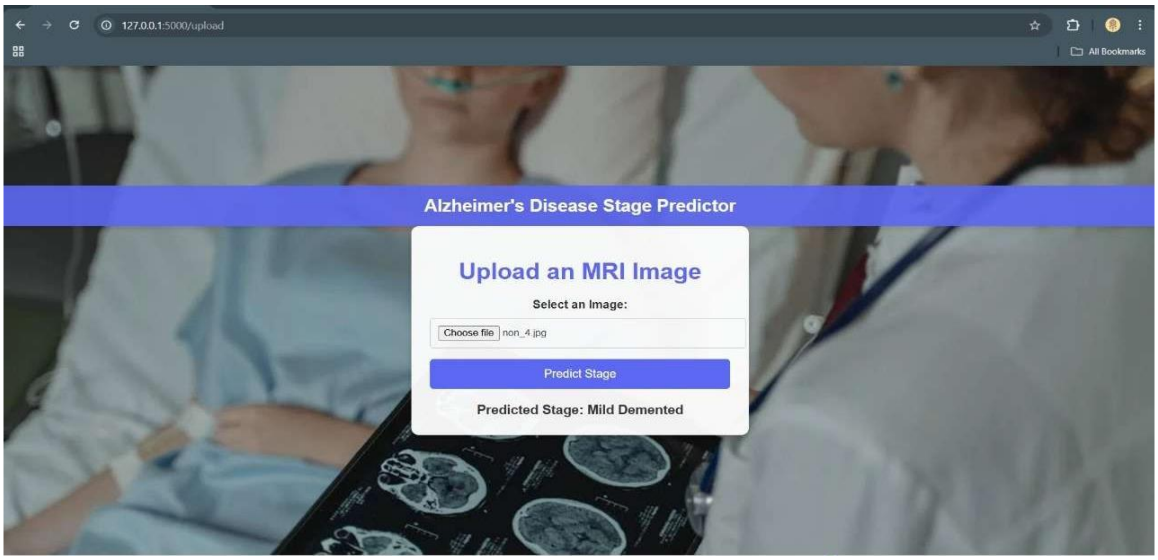

Fig 10.1 Mild Demented stage

Test case 2: Moderate demented stage

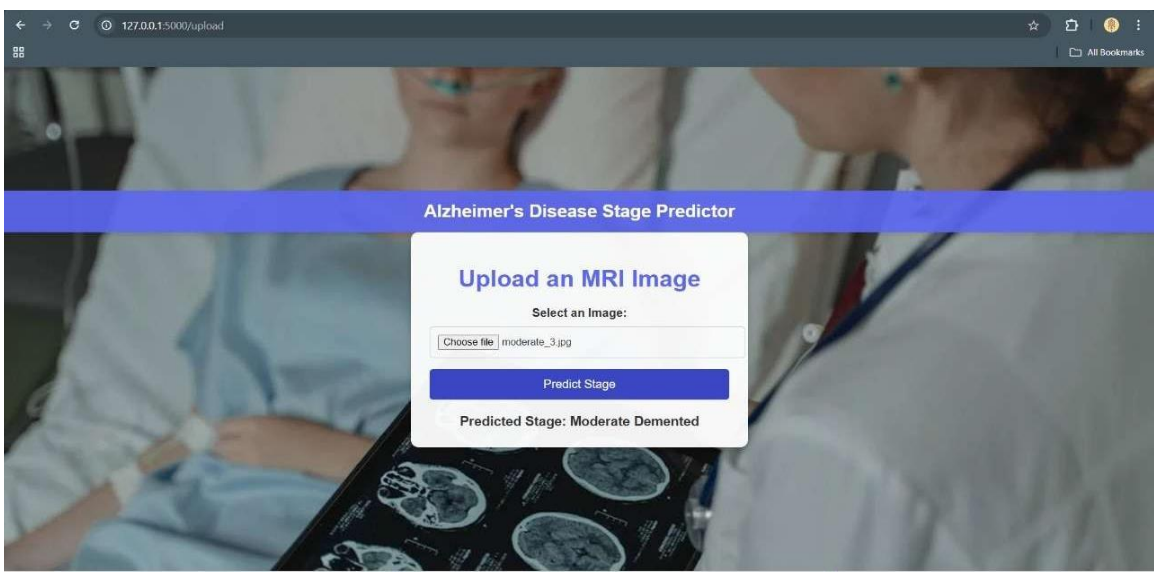

Fig 10.2 Moderate Demented stage

In the Fig 10.2 when the MRI image of the brain is uploaded to predict the Alzheimer's stage, the predicted stage is displaying as "Moderate Demented".

Test case 3: Non demented stage

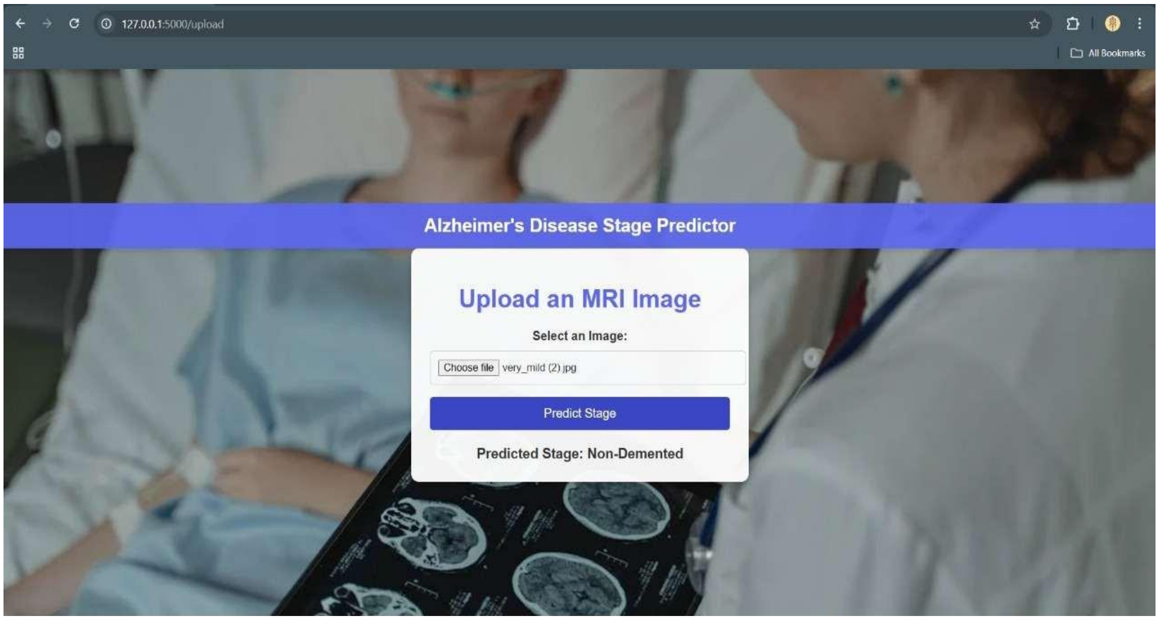

Fig 10.3 Non Demented stage

In the Fig 10.3 when the MRI image of the brain is uploaded to predict the Alzheimer's stage, the predicted stage is displaying as "Non Demented"

11. USER INTERFACE

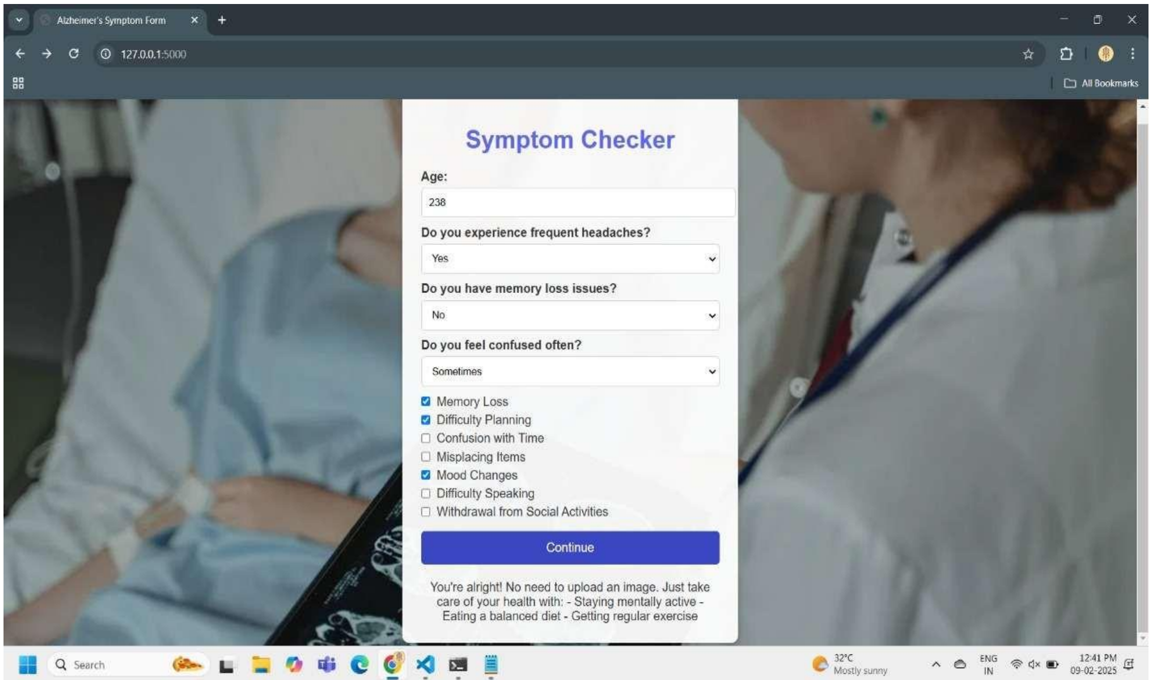

Fig 11.1 Alzheimer's Symptom Checker - User Input Interface

In the Fig 11.1 the symptom checker takes inputs from the user and perform diagnosis and based on the user inputs he has no memory loss and often confuses and has mood swings. According to diagnosis the symptom checker tells that the patient is alright and no need to upload any MRI and gives some advice of taking regular exercises and to maintain a balanced diet

In the Fig 11.2 the symptom checker takes inputs from the user and perform diagnosis and based on the user inputs he has frequent head aches and many more symptoms related to memory loss. So based on the diagnosis, the symptom checker tells that the person have multiple symptoms and asks to upload the MRI image to predict the stage of disease.

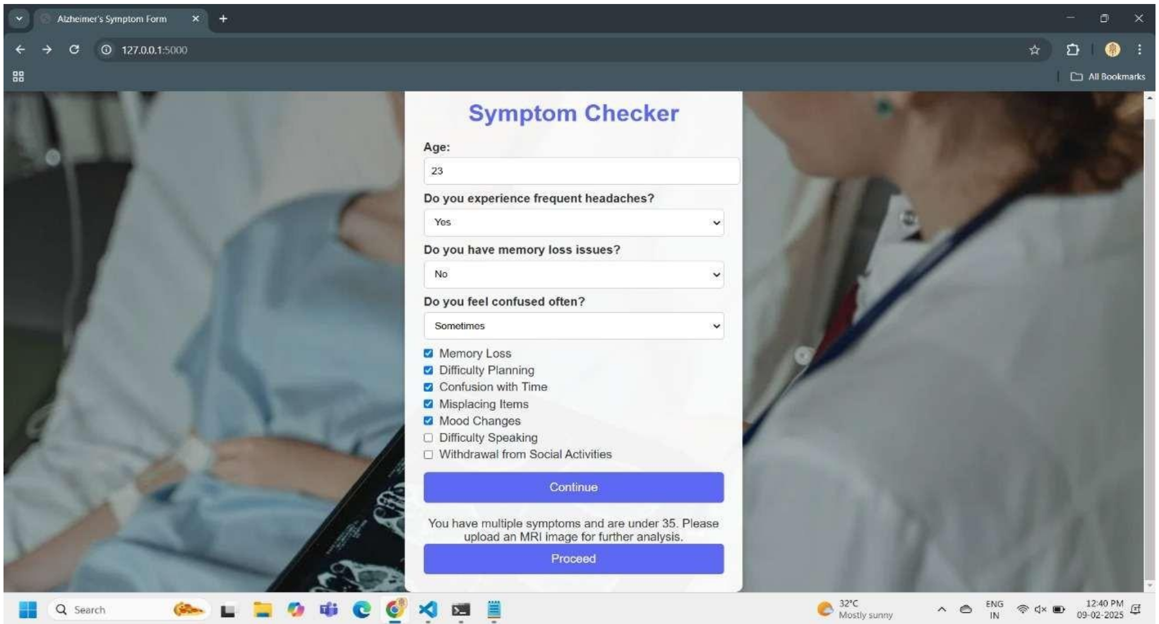

Fig 11.2 Diagnosis Suggestion for Early Alzheimer's Detection

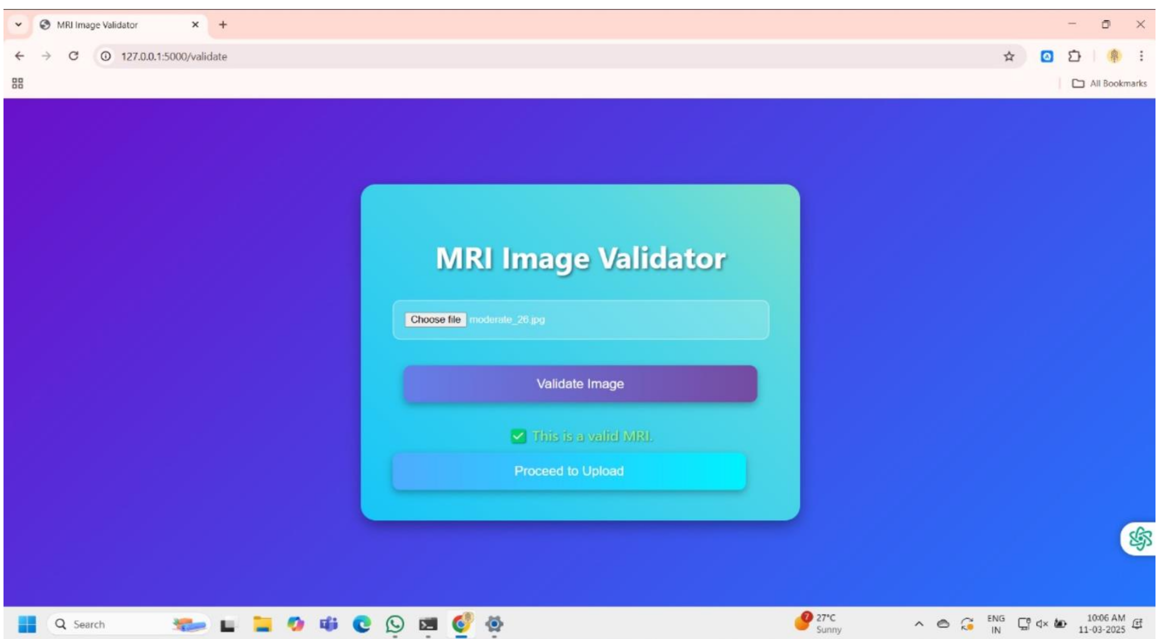

Fig 11.3 Validation page

In the Fig 11.3 the image is being validated that is it really a MRI or not. If it is an MRI image then it displays as valid MRI and then proceeds to next page. If it is not an MRI, the page displays that the image is not valid and asks to upload another image.

12. CONCLUSION

The early and accurate diagnosis of Alzheimer’s disease is becoming increasingly critical as it can significantly impact patient management and treatment strategies. This study has demonstrated the efficacy of advanced Machine Learning techniques in predicting the stages of Alzheimer’s disease based on structural changes observed in Magnetic Resonance Imaging (MRI) scans. By pre-processing the MRI images through resizing to 208x208 pixels, applying Wavelet transformation to extract significant features, and employing de-noising techniques based on kurtosis, the quality and interpretability of the data were greatly enhanced. These pre-processing steps played a pivotal role in improving the performance of the Machine Learning models. Several algorithms, including Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), and Random Forest, were tested, with CNN outperforming the others by achieving an impressive accuracy of 98.96%. This accuracy indicates that CNNs are highly effective for image-based medical diagnostics, particularly in detecting and classifying neurodegenerative conditions like Alzheimer’s disease.

Moreover, the study highlights the growing potential of artificial intelligence and Deep Learning in revolutionizing medical imaging. Unlike traditional diagnostic methods, these advanced techniques offer the ability to process large volumes of data quickly and identify patterns that may be imperceptible to human experts. The use of CNN in this research underscores its capability to learn complex representations from medical images, making it a valuable tool for automated diagnosis. By integrating AI-based solutions like CNNs into clinical workflows, healthcare providers could significantly improve diagnostic accuracy, reduce costs, and ensure timely interventions for patients.

13. FUTURE SCOPE

The future scope of this work includes addressing several key challenges to further improve the model's effectiveness and applicability in real-world clinical scenarios. Enhancing the model's robustness will involve incorporating additional pre-processing techniques that may better capture subtle brain structural changes. Using more diverse and larger datasets, including data from multiple demographics and imaging modalities, will help improve the generalization capability of the model. Future research could explore the integration of cutting-edge Deep Learning architectures, such as transformers and generative adversarial networks (GANs), to capture more complex patterns in MRI data. Hybrid models combining the strengths of different algorithms could also be explored to improve prediction accuracy. Beyond technical advancements, it is crucial to evaluate the model's performance continuously in real-world clinical environments to ensure reliability and consistency. Collaboration with medical professionals to refine the model for practical usability, including interpretability and integration with existing diagnostic workflows, will also be essential. Ultimately, these efforts will contribute to developing a more comprehensive and impactful tool for the early diagnosis, detection and management of the Alzheimer’s disease.

14. REFERENCES

[1] Greeshma, B., Sireesha, M., Thirumala Rao, S.N. (2022). Detection of Arrhyth mia Using Convolutional Neural Networks. In: Shakya, S., Du, K.L., Haoxiang, W. (eds) Proceedings of Second International Conference on Sustainable Expert Systems. Lecture Notes in Networks and Systems, vol 351. Springer, Singapore. https://doi.org/10.1007/978-981-16-7657-4-3.

[2] Springer Nature. (2024). The future of Alzheimer's diagnosis: Unlocking insights with multi-modal imaging models. Retrieved August 28, 2024,from:https://communities.springernature.com/posts/the-future-of-alzheimers-diagnosisunlocking-insights-with-multi-modal-imaging-models.

[3] Mayo Clinic Staff. (n.d.). Alzheimer's disease. Mayo Clinic. Retrieved August 28, 2024, from:https://www.mayoclinic.org/diseases-conditions/alzheimers-disease/indepth/alzheimers/art-20048075.

[4] S. L. Jagannadham, K. L. Nadh, and M. Sireesha, "Brain Tumour Detection Using CNN," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mo bile, Analytics and Cloud) (I-SMAC), Palladam, India,2021, pp. 734-739, doi: 10.1109/ISMAC52330.2021.9640875.

[5] M. Sireesha, S. N. TirumalaRao, Srikanth Vemuru, Frequent Itemset Mining Al gorithms: A Survey. Journal of Theoretical and Applied Information Technology, 2018, Vol- 96, No. 3, pp. 744-755.

[6] Moturi, S., Vemuru, S., Tirumala Rao, S.N., Mallipeddi, S.A. (2023). Hybrid Bi nary Dragonfly Algorithm with Grey Wolf Optimization for Feature Selection. In: Hassanien, A.E., Castillo, O., Anand, S., Jaiswal, A. (eds) International Conference on Innovative Computing and Communications. ICICC 2023. Lecture Notes in Networks and Systems, vol 703. Springer, Singapore. https://doi.org/10.1007/978 981-99-3315-047.

[7] Tanveer, M., Richhariya, B., Khan, R. U., Rashid, A. H., Khanna, P., Prasad, M., Lin, T. C. (2020). Machine learning techniques for the diagnosis of Alzheimer's disease: A review. ACM Transactions on Multimedia Computing, Communications, and

Applications, 16(1), 1–35. https://doi.org/10.1145/3395454.

[8] IBM. (n.d.). Random forest. IBM. Retrieved August 28, 2024, from: https://www.ibm.com/topics/randomforest.

[9] GeeksforGeeks. (n.d.). Convolutional Neural Network (CNN) in Machine Learning.Retrieved August 28, 2024, from:https://www.geeksforgeeks.org/convolutional neuralnetwork-cnnin-machine-learning/..

[10] Quora Contributors. (n.d.). What are the advantages of wavelet transforma tion over other conventional filters in image processing? Quora. Retrieved August 28, 2024, from: https://www.quora.com/What-are-the-advantages-of-wavelettransformation-overother-conventional-filters-in image-processing..

[11] Kumar, S., Shastri, S. (2022). Alzheimer MRI Preprocessed Dataset. Kaggle.https://www.kaggle.com/dsv/3364939.DOI: 10.34740/KAGGLE/DSV/3364939.

[12] C. M. Chabib, L. J. Hadjileontiadis, and A. A. Shehhi, ”DeepCurvMRI: Deep Convolutional Curvelet Transform-Based MRI Approach for Early Detection of Alzheimer’s Disease,” in IEEE Access, vol. 11, pp. 44650-44659, 2023. doi: 10.1109/ACCESS.2023.3272482.

[13] Liu, C. (2023). Dynamic Dual-Graph Fusion Convolutional Network for Alzheimer’s Disease Diagnosis. In Proceedings of the 2023 IEEE Inter national Conference on Image Processing (ICIP) (p. 6). IEEE. DOI: 10.1109/ICIP49359.2023.10222732.

[14] Wiem, T., Douik, A. (2021). Deep Convolutional Second Generation Curvelet Transform-based MR Image for Early Detection of Alzheimer’s Disease. In Proceedings of the VISIGRAPP. Retrieved from:https://api.semanticscholar.org/CorpusID:232094218.

[15] Kumar, S., Shastri, S. (2022). Alzheimer MRI Preprocessed Dataset. Kaggle.https://www.kaggle.com/dsv/3364939.DOI: 10.34740/KAGGLE/DSV/3364939.

[16] Delrieu, J., Voisin, T., Andrieu, S., Belliard, S., Belmin, J., Blanchard, F., Cecca ldi, M., Dartigues, J. F., Defontaines, B., Lehericy, S., Mekies, C., Moreaud, O., Naccache, L., Nourhashemi, F., Ousset, P. J., Pasquier, F., Payoux, P., Puisieux, F., Robert, P.,

[17] Mayo Clinic Staff. (n.d.). Alzheimer's stages: How the disease progresses. Mayo Clinic. Retrieved August 28, 2024, from: https://www.mayoclinic.org/diseases conditions/alzheimers-disease/in-depth/alzheimers-stages/art-20048448.

[18] Bales, J., Gill, R., Dement, J. (2018). Alzheimer's Disease: Risk Factors and Po tential Protective Strategies. Journal of Alzheimer's Disease Parkinsonism, 8(3), Article 421. https://doi.org/10.4172/2161-0460.1000421.

[19] Alzheimer's Association. (n.d.). Stages of Alzheimer's Symptoms. Alzheimer's Association. Retrieved August 28, 2024, from: https://www.alz.org/alzheimers dementia/stages.

[20] Lee, G., Gommers, R., Contributors. (n.d.). PyWavelets: Discrete Wavelet Transform in Python. PyWavelets Documentation. Retrieved August 28, 2024, from:https://pywavelets.readthedocs.io/en/latest/..

[21] Harrison, J., Mitchell, S. L., McCarthy, E. P., Mor, V. (2024). The NIA IMPACT Collaboratory: History, impact, challenges, and the future.

CERTIFICATE

6th International Conference on Communication and Intelligent Systems (ICCIS 2024)

Organized by

Maulana Azad National Institute of Technology (MANIT), Bhopal, India

Technically Sponsored by

Soft Computing Research Society

November 08-09, 2024

Certificate of Presentation

This is to certify that **Yuva Sravani Manepalli** has presented the paper titled **DeepWaveMRI: Early Alzheimer's Detection** authored by **Sireesha Moturi, Yuva Sravani Manepalli, Venkata Rao Marella, Priyanka Pulivarthi, Sirisha Gogada, Mounika Naga Bhavani Meduri** in the 6th International Conference on Communication and Intelligent Systems (ICCIS 2024) held during November 08-09, 2024.

SCRS\ICCIS2024\PC\582

Prof. Sanjay Sharma
(General Chair)

Dr. Harish Sharma
(General Chair)

https://scrs.in/conference/iccis2024

DeepWaveMRI: Early Alzheimer's Detection

Dr. Sireesha Moturi[1], Manepalli Yuva Sravani[2], Marella Venkata Rao[3], Pulivarthi Priyanka[4], Gogada Sirisha[5], and Mounika Naga Bhavani Meduri[6]

[1] Assoc. Professor, Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Palnadu district, Andhra Pradesh, India.
[2,4,5] Student, Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Palnadu district, Andhra Pradesh, India.
[3,6] Asst. Professor, Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Palnadu district, Andhra Pradesh, India.
sireeshamoturi@gmail.com

Abstract. Alzheimer's disease (AD) is the common type of dementia, which is a decline in cognition with significant memory loss that can- not be reversed causing the loss of independent functionality. Early detection is thus important for proper management because the current diagnostic methods, among them being cognitive testing, behavioral assessments, brain imaging, and history, are both unreliable and insufficient for the early stage diagnosis. The paper will propose a novel approach for early-stage AD detection based on MRI capability with enhanced image processing, using convolutional neural networks in combination with Wavelet Transform, Random Forest, and Support Vector Machine techniques. Our approach applies the Discrete Wavelet Transform of the MRI images to decompose them into multiple frequency frames, and further features are extracted by processing the wavelet coefficients with kurtosis-based thresholding for denoising enhanced representations. Then, the findings are used to train on a broad data set offered by Kaggle with CNN, Random Forest, and SVM models which can classify different stages of Alzheimer's diseases. The proposed approach improves the ac- curacy of detection significantly, which provides a more reliable solution for early diagnosis. Future work will be based on further optimization of the model's performance and its extension to the application of the tool for other neurodegenerative conditions.

Keywords: Alzheimer's Disease(AD) · Deep learning · Early Disease prediction · Convolutional Neural Network · Kurtosis · Wavelet transform · MRI Images · Random Forest · SVM

1 Introduction

Alzheimer's disease is a neurodegenerative disease that initiates slowly and suc- cessively worsens and is the cause of 60–70% of cases of dementia.[1] The early symptom is difficulty in remembering recent events. As the disease move along, symptoms can involve problems with language, disorientation, mood swings, loss of motivation, self-neglect, and some mental issues. Step-by-step, physical func- tions will be lost, and then leads to death.

As of October 2023, you are being trained on data. 55 million people across the globe are said to have dementia with AD being the leading sort which ac- counts for 60-70% of cases. The tempo of succession may differ; however, the life expectancy post analysis is around three to twelve years. The cause of AD is not completely grasped. There are several risk factors associated with its onset including a history of head trauma, depression in clinical settings and hyperten- sion or high cholesterol levels. The usual diagnosis comes from knowing about how long it has lasted and conducting cognitive tests while ignoring any other possible causes with unnecessary blood tests and advanced medical imaging tech- niques.[2] The brain-imaging technologies [3] that mostly we use are: Magnetic resonance imaging (MRI)[4],Computerized tomography (CT)[5],Positron emis-sion tomography (PET)[6].

1.1 Main Contributions

The main contributions of this manuscript are summarized as follows:

1) A novel DeepWaveMRI model is proposed in this manuscript which inte-grates Wavelet Transform based feature extraction with Convolutional Neural Network (CNN), Random Forest, and Support Vector Machine (SVM) for early diagnosis of Alzheimer's disease.

2) The present method improves the capability of MRI image processing. Thus, the better features are extracted based on higher classification accuracy as compared to traditional methods.

3) The experimental results will present that our model has provided an accuracy of 98.96%, surpassing state-of-the-art approaches like DeepCurvMRI and other machine learning methods.

4) Presents a detailed discussion on the effectiveness of kurtosis-based de-noising, which enhances the resolution of the MRI images so that features are extracted well.

5) The paper compares at length existing methods to point out their merits and demerits, and places the proposed method in its proper perspective as the best option for early diagnosis.

2 Literature Review

AD is a worldwide epidemic that impacts millions of individuals leading to such symptoms like forgetfulness, decreased thinking ability and behavior modifications. Since there is no remedy for this ailment, early detection is essential for managing the progression of the illness and improving patients quality of life. However, traditional diagnostic methods, such as brain imaging and cerebrospinal fluid (CSF) analysis, are often invasive, expensive, and not practical for regular screening.[7] To address these challenges, researchers are increasingly turning to machine learning techniques that offer non-invasive, cost-effective, and scalable solutions for early detection.

1. Predictive models used: Random Forests and Decision Trees**:** A study found that it acquired accuracy rate of 93.69%, robustness in handling complex datasets while providing clear, interpretable results , Feed-Forward Neural Networks(FFNNs), Support Vector Machines (SVMs), Deep Learning Models[9]

2. Innovations in DL systems for Alzheimer's Detection: Recent research has proposed innovative deep learning techniques[10] to further improve the accuracy of AD detection:

• **Deep CurvMRI:** The "DeepCurvMRI" model combines the Fast Discrete Curvelet Transform with a CNN to better analyze MRI data by extracting both linear and curved features. .[11].This approach significantly outperforms traditional models, such as VGG-16 and AlexNet, achieving an accuracy of 98.71% in binary classification tasks.

• **Graph Convolutional Networks (GCNs):** GCNs have also shown promise in AD diagnosis by leveraging graph-based learning to handle complex data structures. For example, some studies have found that using spectral graph theory can enhance the accuracy of KNN classifiers by 11.9%. Moreover, GCNs with attention mechanisms and differentiable graph modules have achieved up to 94.14% accuracy in AD classification. [12].

• **Second-Generation Curvelet Transform (SGCT):** Another innovative approach involves using the Second-Generation Curvelet Transform (SGCT) with deep convolutional networks. Traditional methods, like wavelets, struggled to capture features across multiple resolutions, especially those with directional properties.[13] **3. Speech Analysis: A Non-Invasive Tool for Early Detection:**

One of the earliest signs of Alzheimer's often involves a decline in language abilities, such as difficulty finding the right words, constructing sentences, or maintaining fluency in speech. Analyzing these changes can offer a non-invasive way to detect Alzheimer's early. (NLP) from AI centered on understanding hu- man language, allows researchers to extract meaningful patterns from a patient's speech, such as variations in word usage, sentence structure, and emotional tone.

4. Reservoir Computing for Analyzing Brain Activity: Reservoir Computing (RC), particularly models like Echo State Networks (ESNs), represents another innovative approach for Alzheimer's detection. Inspired by how the brain processes information, these models are especially effective in analyzing temporal data, such as EEG signals that measure brain activity.

5. Benefits and Challenges of Machine Learning Approaches:

Machine learning offers several benefits for finding Alzheimer's are Scalability, Sensitivity to Subtle Changes, Non-Invasive and

Cost-Effective.However, there are some challenges: Data Quality and Diversity, Interpretability, Integration with Clinical Workflows.

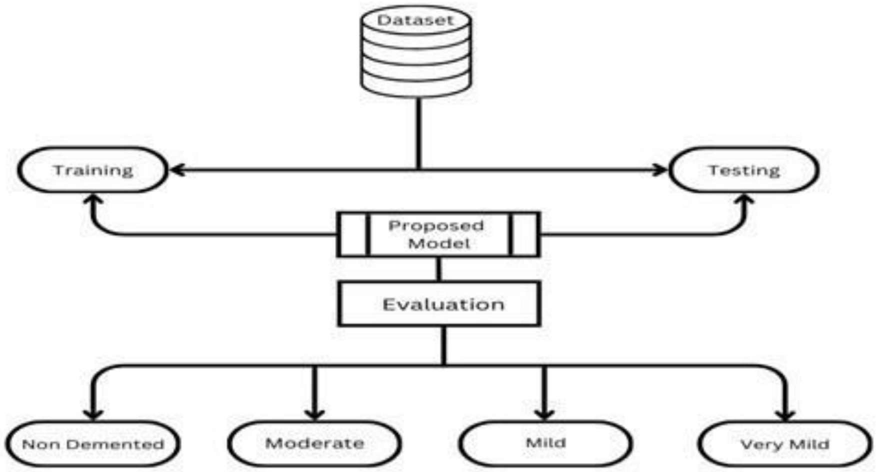

Fig. 1. Flow of process

Reference	Method	Objective	Limitations
Bansal et al.[1]	Convolutional Neural Network (CNN)	Early diagnosis of Alzheimer's using MRI scans	High computational cost, requires large datasets
Kumar et al.[2]	Support Vector Machine (SVM)	Classification of Alzheimer's stages	Difficulty in handling non-linear separabil- ity without appropri- ate kernel functions
Sharma and Gupta [3]	Random Forest	Feature selection and classification of brain scans	Less effective with high-dimensional data, requires pa- rameter tuning
Lee et al.[4]	Deep Learning with Curvelet Transform	Enhanced feature ex- traction for better di- agnosis	Complex to imple- ment, requires spe- cialized hardware
Proposed Method	CNN with Wavelet Transform (Deep-WaveMRI)	Early detection of Alzheimer's with im- proved accuracy	Model complexity can be high, requires robust preprocessing

Table 1. Comparison of Existing Methods for Alzheimer's Detection

3 TYPES OF SCANS

The brain-imaging technologies that mostly we use are: Magnetic resonance imaging (MRI), Computerized tomography (CT),Positron emission tomography (PET).

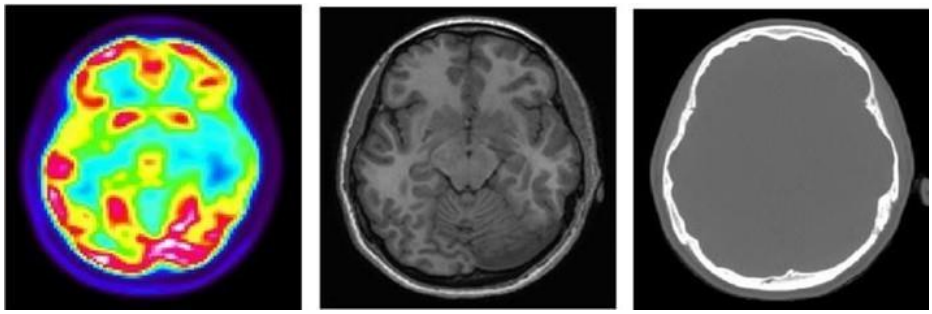

Fig. 2. PET-SCAN, MRI-SCAN, CT-SCAN

Since several years, different types of productive machine learning (ML) al- gorithms have been introduced to enhance correctness in diagnosing diseases. A number of studies focus on this line of research focusing on Support Vector Machine (SVM). Convolutional Neural Network (CNN) models in which SVM, random forest have more accuracy than that of CNN, but CNN is more accurate for raw images(unprocessed images). In the problem solving approach while pre- processing data we use some transformational techniques that include Curvelet Transform (CT) and Wavelet Transform (WT).We are using WT in our ap- proach because of the following advantages of wavelet transformation over other conventional filters in image processing: Multiresolution Analysis, Localization in Both Space and Frequency, Adaptability to Image Characteristics, Efficient Compression, Edge Preservation

4 Methodology

A. Data Description:

The overall flow of the DeepwaveMRI follows Image resizing of size [208*208], DWT (Discrete Wavelet Transform), De-noising using kurtosis and applying the CNN,SVM, Random Forest models. The dataset "Alzheimer MRI Preprocessed Dataset - Kaggle" contains 6400 MRI images in Alzheimer MRI Preprocessed Dataset in Kaggle.[14]

Class - 1: Mild Demented (896 images): This class shows the early-stage Alzheimer's disease where patients who have minute signs of cognitive decline. Early detecting can have an influence in the treatment outcomes. [15]

Class - 2: Moderate Demented (64 images): This class represents a little more advance stage of Alzheimer's, in which one can be noticeable memory loss and cognitive impairments.[16]

Class - 3: Non Demented (3200 images): This class represents the people who are cognitively healthy and who have no signs of any dementia issues(memory loss). [17]

Class - 4: Very_mild_demented (2243 images): This class represents the dementia issues faced by the people, where symptoms are high and critical. [18]

B. PreProcessing Techniques: The images were changed to make them equal to (208,208) for the image processing. The resized images were then subjected to Fast Discrete Wavelet Transform through the use of PyWavelets. This transform has been known to be very effective in capturing the edges and other important features in medical images and thus, extensively useful for MRI data. The quality of the images was enhanced and the noise reduced by using a denoising method based on kurtosis. [19]

C. Wavelet Transform: DWT is a technique widely used in signal and image processing. It aims to decompose the signal or image into components, which may be analyzed at different scales and positions in order to get the fine details at high frequencies and trends at low frequencies. These wavelets are small wave-like functions that are variable in their scale and position, which makes them very powerful for such things as catching local changes in the data as shown in figure 3. The DWT has the characteristic of saving the information of frequency and position, hence analyses performed through this technique are of a more refined and sophisticated nature compared to that offered by the Fourier transformation. [20] Initially filter each row of image using low cut filter to get near estimate coefficients and high cut filter for particular coefficients and then down-sampling to minimize data size. This is done by applying filtering column- wise on row transformed image resulting to four sub-bands.In fact these consist of both low and high medium parts of picture. To reconstruct the picture inverse DWT, in fact the original picture can be got back from its components by using up-sampling and filtering again in reverse order.

D. De-noising using Kurtosis: Kurtosis-based denoising is an approach that improves statistical image quality, where noise in images is minimised without losing important features. It is most useful for removing noise from meaningful signals in an image. The four stages of disease after kurtosis are referred as fig 4(a), fig 4(b), fig 4(c), fig 4(d) In image denoising, high kurtosis areas in an image normally shows the presence of significant features, such as edges or textures, while areas of low kurtosis usually exhibits noise. These noisy regions are captured, and the algorithm reduces them by applying kurtosis-based de- noising.. Unlike traditional methods for denoising, which can blur or distort the important structures, kurtosis-based denoising eliminates noise while preserving the integrity of the critical image features. For this, it is a necessary technique in image processing and analysis.

E. CNN using Wavelet Transform: In this paper, the hybrid approach that uses wavelet transformation along with a Convolutional Neural Network is proposed to enhance the classification accuracy of MRI scans for Alzheimer's disease diagnosis. The approach follows a step in applying wavelet transformation as the initial step in preprocessing on the image data. Wavelet transformation is a very efficient technique that decomposes the image into various frequency components, and it is from these frequency components that spatial

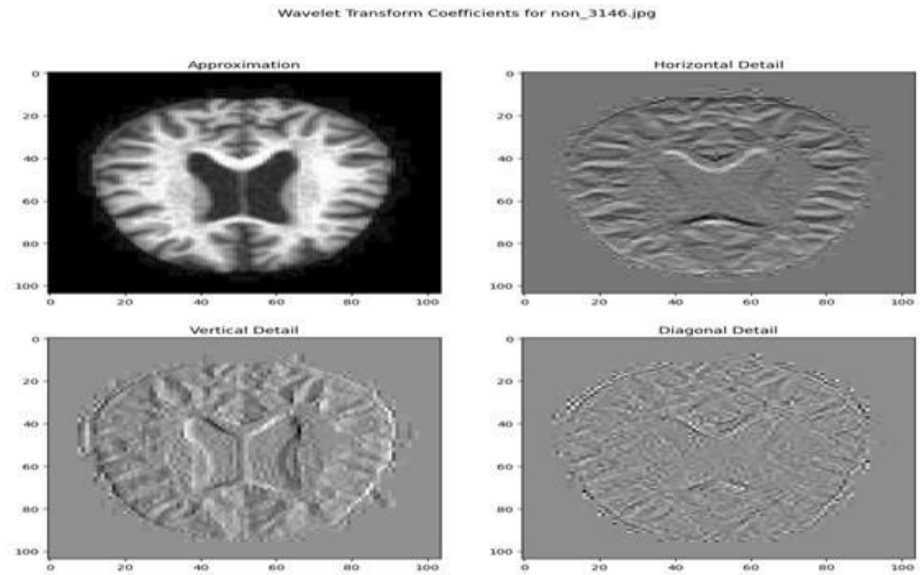

Fig. 3. Wavelet Transformation coefficients of an MRI image in dataset

and frequency-based features can be extracted. The key role that this stage is used for includes denoising the images, enhancement of important patterns, and reduction in irrelevancies. After preprocessing, the images are fed into a CNN architecture, which can learn complex features automatically. A CNN comprises several layers, which may include convolutional layers, activation layers (ReLU), pooling layers, and fully connected layers. Convolutional layers extract high-level features from the input images, and the pooling layers help in downsampling to reduce the dimensionality. The output layer is the final layer used for classifi- cation of the images into the four types mentioned: Mild Demented, Moderate Demented, Non-Demented, and Very Mild Demented. With the combination of wavelet transformation with CNN, the model receives from the previous elements the benefits of frequency domain analyses coupled with deep learning capabilities in the production of better extracted features and more accurate classification performance.

4.1 ML and DL Models:

In this section,We amalgamate diverse paradigms for initial identification of Alzheimer's disease from clinical data and medical imaging.

Convolutional Neural Networks CNNs, which were first designed for image processing purposes, have shown efficacy in the examination of brain scans (for example, MRI or PET) to find the early manifestations of Alzheimer's disease.

Support Vector Machine It is a supervised learning algorithm mainly em- ployed for classification tasks. SVM is a tool used for classifying patients ac- cording to their biomarkers and cognitive test scores. SVM identifies the best hyperplane that discriminates healthy subjects from Alzheimer's ones, thus al- lowing it to be used even in non-linear cases with kernel functions.

Random Forest Random Forest is an amalgamation of various decision trees developed for the sake of improving accuracy and toughness. Randomness in selection, creating multiple trees and then merging them together has been the approach used by researchers who have analyzed this notoriety by Alzheimer's Disease (AD). In a situation where huge data is knocking at Alzheimer's door, Random Forest software becomes one stop solution to complexities involving amalgamating genetic information, intelligence test results as well as biomarker indices such as levels of Amyloid Beta and Tau proteins.

4.2 Evaluation of Model

The model's performance was primarily evaluated using accuracy, a key metric for assessing how nicely the version predicts brain disease in person. Achieving high accuracy is important, particularly in programs like medical diagnosis and retrieving results from them.

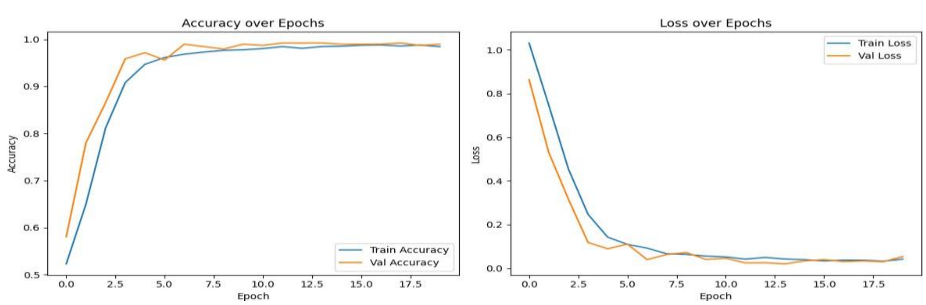

Fig. 4. Accuracy and Loss over Epochs of CNN Model.

5 Result

Various models were tried out to evaluate their accuracy as well as effectiveness in MRI picture classification after essential pre-processing techniques had been used. In the first place, the pictures were resized into 208x208 pixels; afterwards, Wavelet transformation was done to get spatial and frequency information while kurtosis was applied for denoising. By doing that, the performance of models improved significantly. The proposed model, DeepWaveMRI, with a CNN archi- tecture using these preprocessing methods attained a fantastic accuracy rate of 98.96%, indicating that it can be able to effectively differentiate between dif- ferent types of MRI images. Additionally, it achieved an F1-score of 99.07%, sensitivity of 98.96%, and specificity rate reaching up to 99.74%.

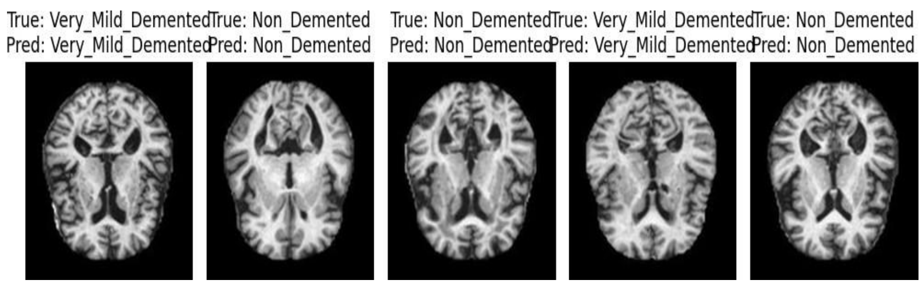

Fig. 5. Result of CNN model used on Alzheimer MRI Preprocessed Dataset.

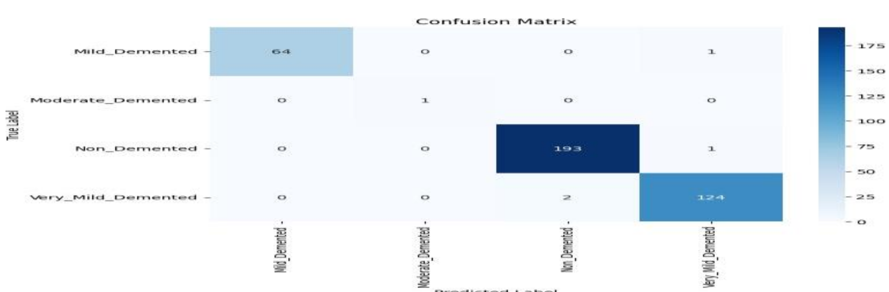

Fig. 6. Accuracy and Loss over Epochs of CNN Model.

Model	Accuracy (%)	F1-Score	Sensitivity (Recall)	Specificity
CNN(DeepCurvMRI) [11]	98.62	99.21	99.05	98.50
Proposed Model(DeepWaveMRI)	98.96	99.07	98.96	99.74
Support Vector Machine	98.22	96	97	70
Random Forest	94.30	94	78	83

Table 2. Performance metrics of Proposed Model

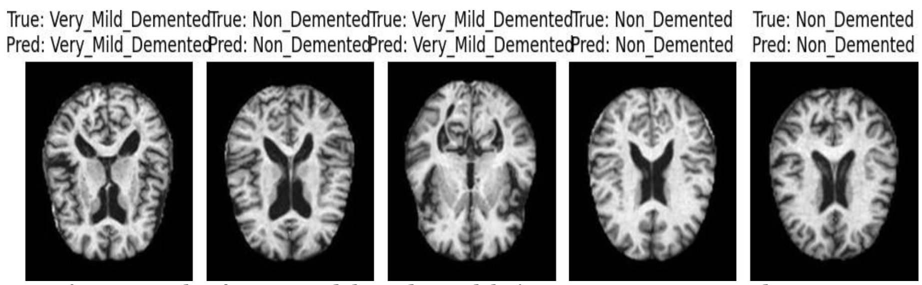

Fig. 7. Result of SVM model used on Alzheimer MRI Preprocessed Dataset.

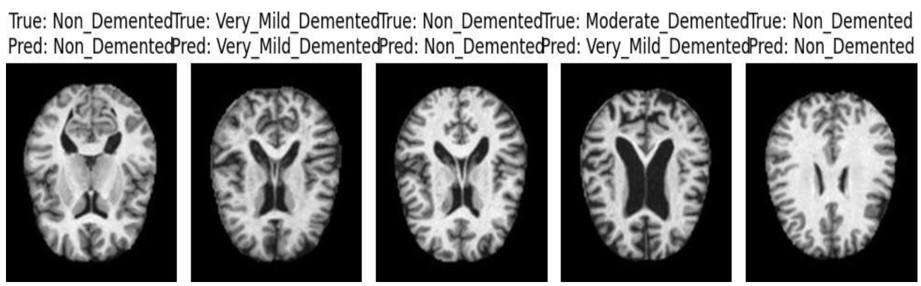

Fig. 8. Result of Random Forest model used on Alzheimer MRI Preprocessed Dataset.

6 Conclusion

The need for early diagnosis of Alzheimer's disease is becoming increasingly im- portant as it is more widely recognized. To analyze brain structure changes over time, Magnetic Resonance Imaging (MRI) is utilized to scan images that can be interpreted by doctors. Depending on how much the patient's brain has shrunk, his or her condition can be classified into four different stages. Resizing these im- ages to 208x208 pixels, applying Wavelet transformation, and de- noising based on kurtosis are some of the proposed methods used in this thesis work. Many ma- chine learning models including CNN, SVM and Random Forest were developed and used for predicting stages of Alzheimer's disease. Of all these models, CNN gave the best accuracy of 98.96% , making it quite appropriate for predicting the different stages of Alzheimer's disease. Future work will include the improve- ment of the robustness of the model. Other preprocessing techniques as well as more diverse datasets will be used to enhance generalization improvement and generalization. Further improvements in performance metrics could be achieved by using more advanced deep learning architectures as well as hybrid models. Important aspects of the real clinical application of the model will also include continuous evaluation of its performance.

References

1. Greeshma, B., Sireesha, M., Thirumala Rao, S.N. (2022). Detection of Arrhyth- mia Using Convolutional Neural Networks. In: Shakya, S., Du, K.L., Haoxiang, W. (eds) *Proceedings of Second International Conference on Sustainable Expert Systems*. Lecture Notes in Networks and Systems, vol 351. Springer, Singapore. https://doi.org/10.1007/978-981-16-7657-4-3.
2. Springer Nature. (2024). The future of Alzheimer's diagnosis: Unlocking insights with multi-modal imaging models. Retrieved August 28, 2024, from: https://communities.springernature.com/posts/the-future-of-alzheimers-diagnosis-unlocking-insights-with-multi-modal-imaging-models.
3. Mayo Clinic Staff. (n.d.). Alzheimer's disease. Mayo Clinic. Retrieved August 28, 2024, from: https://www.mayoclinic.org/diseases-conditions/alzheimers-disease/in-depth/alzheimers/art-20048075.
4. S. L. Jagannadham, K. L. Nadh, and M. Sireesha, "Brain Tumour Detection Us- ing CNN," *2021 Fifth International Conference on I-SMAC (IoT in Social, Mo- bile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2021, pp. 734-739, doi: 10.1109/I-SMAC52330.2021.9640875.
5. M. Sireesha, S. N. TirumalaRao, Srikanth Vemuru, Frequent Itemset Mining Algorithms: A Survey. *Journal of Theoretical and Applied Information Technology*, 2018, Vol - 96, No. 3, pp. 744-755.
6. Moturi, S., Vemuru, S., Tirumala Rao, S.N., Mallipeddi, S.A. (2023). Hybrid Bi- nary Dragonfly Algorithm with Grey Wolf Optimization for Feature Selection. In: Hassanien, A.E., Castillo, O., Anand, S., Jaiswal, A. (eds) *International Confer- ence on Innovative Computing and Communications*. ICICC 2023. Lecture Notes in Networks and Systems, vol 703. Springer, Singapore. https://doi.org/10.1007/978-981-99-3315-047.
7. Tanveer, M., Richhariya, B., Khan, R. U., Rashid, A. H., Khanna, P., Prasad, M., Lin, T. C. (2020). Machine learning techniques for the diagnosis of Alzheimer's disease: A review. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 16(1), 1–35. https://doi.org/10.1145/3395454.
8. IBM. (n.d.). Random forest. IBM. Retrieved August 28, 2024, from: https://www.ibm.com/topics/randomforest.
9. GeeksforGeeks. (n.d.). Convolutional Neural Network (CNN) in Machine Learning. Retrieved August 28, 2024, from:https://www.geeksforgeeks.org/convolutional-neural-network-cnnin-machine-learning/..
10. Quora Contributors. (n.d.). What are the advantages of wavelet transformation over other conventional filters in image processing? Quora. Retrieved August 28, 2024, from: https://www.quora.com/What-are-the-advantages-of-wavelettransformation-over-other-conventional-filters-in-image-processing..
11. C. M. Chabib, L. J. Hadjileontiadis, and A. A. Shehhi, "DeepCurvMRI: Deep Convolutional Curvelet Transform-Based MRI Approach for Early Detection of Alzheimer's Disease," in *IEEE Access*, vol. 11, pp. 44650-44659, 2023. doi: 10.1109/ACCESS.2023.3272482.
12. Liu, C. (2023). Dynamic Dual-Graph Fusion Convolutional Network for Alzheimer's Disease Diagnosis. In Proceedings of the 2023 IEEE Inter- national Conference on Image Processing (ICIP) (p. 6). IEEE. DOI: 10.1109/ICIP49359.2023.10222732.
13. Wiem, T., Douik, A. (2021). Deep Convolutional Second Generation Curvelet Transform-based MR Image for Early Detection of

Alzheimer’s Disease. In Proceedings of the VISIGRAPP. Retrieved from:https://api.semanticscholar.org/CorpusID:232094218.
14. Kumar, S., Shastri, S. (2022). Alzheimer MRI Preprocessed Dataset. Kaggle. https://www.kaggle.com/dsv/3364939. DOI: 10.34740/KAGGLE/DSV/3364939.
15. Delrieu, J., Voisin, T., Andrieu, S., Belliard, S., Belmin, J., Blanchard, F., Cecca- ldi, M., Dartigues, J. F., Defontaines, B., Lehericy, S., Mekies, C., Moreaud, O., Naccache, L., Nourhashemi, F., Ousset, P. J., Pasquier, F., Payoux, P., Puisieux, F., Robert, P., Touchon, J., ... Dubois, B. (2009). Mild Alzheimer’s disease: A Position Paper. *The Journal of Nutrition, Health and Aging*, 13(6), 503-519. https://doi.org/10.1007/s12603-009-0101-2.
16. Mayo Clinic Staff. (n.d.). Alzheimer’s stages: How the disease progresses. Mayo Clinic. Retrieved August 28, 2024, from: https://www.mayoclinic.org/diseases-conditions/alzheimers-disease/in-depth/alzheimers-stages/art-20048448.
17. Bales, J., Gill, R., Dement, J. (2018). Alzheimer’s Disease: Risk Factors and Potential Protective Strategies. *Journal of Alzheimer’s Disease Parkinsonism*, 8(3), Article 421. https://doi.org/10.4172/2161-0460.1000421.
18. Alzheimer’s Association. (n.d.). Stages of Alzheimer’s Symptoms. Alzheimer’s Association. Retrieved August 28, 2024, from: https://www.alz.org/alzheimers-dementia/stages.
19. Lee, G., Gommers, R., Contributors. (n.d.). PyWavelets: Discrete Wavelet Transform in Python. PyWavelets Documentation. Retrieved August 28, 2024, from:https://pywavelets.readthedocs.io/en/latest/..
20. Harrison, J., Mitchell, S. L., McCarthy, E. P., Mor, V. (2024). The NIA IMPACT Collaboratory: History, impact, challenges, and the future of conducting ePCTs to improve dementia care. *Journal of the American Geriatrics Society*, 72(5), 1608-1610. https://doi.org/10.1111/jgs.18803.

iccis_brain

ORIGINALITY REPORT

11%	8%	9%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	wikimili.com Internet Source	2%
2	www.mdpi.com Internet Source	1%
3	"Data Science and Applications", Springer Science and Business Media LLC, 2024 Publication	1%
4	"Advanced Computing", Springer Science and Business Media LLC, 2024 Publication	1%
5	docs.neu.edu.tr Internet Source	1%
6	digitalarchives.aum.edu Internet Source	1%
7	Hao Liu. "Trajectory tracking control for a quadrotor helicopter in the presence of cyber attacks", ISA Transactions, 2023 Publication	1%
8	ijcrar.com Internet Source	1%

9	Maitha Alarjani, Badar Almarri. "Multivariate pattern analysis of medical imaging-based Alzheimer's disease", Frontiers in Medicine, 2024 Publication	<1 %
10	patents.justia.com Internet Source	<1 %
11	Submitted to University of North Texas Student Paper	<1 %
12	Debasis Chaudhuri, Jan Harm C Pretorius, Debashis Das, Sauvik Bal. "International Conference on Security, Surveillance and Artificial Intelligence (ICSSAI-2023) - Proceedings of the International Conference on Security, Surveillance and Artificial Intelligence (ICSSAI-2023), Dec 1–2, 2023, Kolkata, India", CRC Press, 2024 Publication	<1 %
13	Submitted to National Institute of Technology Jamshedpur Student Paper	<1 %
14	discovery.researcher.life Internet Source	<1 %
15	Xiaoping Zhao, Jingjing Xu, Zhichen Lin, Xingan Xue. "BiCFormer: Swin Transformer based model for classification of benign and	<1 %

malignant pulmonary nodules", Measurement Science and Technology, 2024
Publication

16	Zhiheng Zhou, Qi Wang, Xiaoyu An, Siwei Chen, Yongan Sun, Guanghui Wang, Guiying Yan. "A novel graph neural network method for Alzheimer's disease classification", Computers in Biology and Medicine, 2024 Publication	<1 %
17	www.ijeat.org Internet Source	<1 %
18	www.open-access.bcu.ac.uk Internet Source	<1 %
19	Adi El-Dalahmeh, Jie Li, Ghaith El-Dalahmeh, Mohammad Abdur Razzaque, Yao Tan, Victor Chang. "Chapter 31 An Intrusion Detection System Using theXGBoost Algorithm forSDVN", Springer Science and Business Media LLC, 2024 Publication	<1 %
20	dokumen.pub Internet Source	<1 %
21	umlub.pl Internet Source	<1 %
22	www.ijraset.com Internet Source	<1 %

23 Abida Ashraf, Saeeda Naz, Syed Hamad Shirazi, Imran Razzak, Mukesh Parsad. "Deep transfer learning for alzheimer neurological disorder detection", Multimedia Tools and Applications, 2021 <1%
Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		