A

Project Report

On

**Alzheimer's Disease Detection Using MRI Scan**

*Submitted for partial fulfillment of the requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

By

**G. Abhinav (19N81A0598)**

**J. Naveen kumar(19N81A0596)**

**K. Nikhil(19N81A05A4)**

Under the guidance of

**Mr. T. Bala Krishna (Ph.D)**

*Assistant Professor*

Department of CSE



**Department of Computer Science and Engineering**

**Sphoorthy engineering college**

**(Affiliated to JNTUH & Recognized by AICTE)**

**Nadergul, Saroor Nagar Mandal, R.R. Dist., Telangana– 501510**

**2023**

# 

# **DECLARATION**

We the undersigned, declare that the mini project title “**Alzheimer's Disease Detection Using MRI Scan**” carried out at “Sphoorthy engineering college” is original and is being submitted to the Department of Computer science engineering, Sphoorthy Engineering College, and Hyderabad towards partial fulfillment for the award of Bachelor of Technology.

We declare that the result embodied in the Major Project work has not been submitted to any other University or Institute for the award of any Degree or Diploma.

**G. Abhinav (19N81A0598)**

**J. Naveen Kumar (19N81A0596)**

**K. Nikhil (19N81A05A4)**

## **C:\Users\admin\Downloads\logo-new.jpeg**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**CERTIFICATE**

This is to certify that this Project Report entitled **“Alzheimer's Disease Detection Using MRI Scan”** is a bonafide work carried out by **G. Abhiav (19N81A0598), J.Naveen Kumar(19N81A0596) , K. Nikhil (19N81A05A4),** in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology** in **Computer Science And Engineering** from **Sphoorthy Engineering College,** affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad, during the Academic Year 2022-23 under our guidance and supervision.The results embodied in this report have not been submitted to any other university or institute for the award of any degree.

**INTERNAL GUIDE HEAD OF THE DEPARTMENT**

**Mr. Tejavath Bala Krishna (Ph.D)**  **Mr. P. Ram Mohan (Ph.D)**

Professor Head

Department of CSE Department of CSE

SPHN SPHN

**EXTERNAL EXAMINER PRINCIPAL**

**Dr. V. Venkata Krishna (Ph.D)**

# **ACKNOWLEDGEMENT**

We express our deep sense of gratitude to our Project Coordinators **Mr. Venu Gopal & Mrs. Srilatha Reddy** for their constant and continuous support throughout the project. We sincerely thank our Project Guide **Mr. T. Bala Krishna (Ph.D)**, Asst. Professor, Department of Computer Science & Cybersecurity, Sphoorthy Engineering College, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad for his inspiring guidance, consistent encouragement, constructive criticism and helpful suggestions during the entire course of my research work.

We express our sincere thanks to **Mr. P. Ram Mohan Rao**, Associate Professor & Head of the Department, Department of Computer Science & Engineering, Sphoorthy Engineering College, Nadergul (V), Balapur (M), Rangareddy (D) for his encouragement which helped me to complete my Project work.

We deem it a great privilege to express our profound gratitude and sincere thanks to **Mr. S. Chalama Reddy** Chairman, **Mr. S. Jagan Mohan Reddy** Secretary, **Dr. V. Venkata Krishna** Principal, **Prof. M.V.S. Ram Prasad**, Director, Sphoorthy Engineering College, Nadergul (V), Balapur (M), Rangareddy (D), for their moral support and help in the completion of my research work.

We express our heartfelt thanks to Professors, Associate Professors, Assistant Professor and other professional non-teaching staff of Department of Computer Science and Engineering, Sphoorthy Engineering College, Nadergul (V), Balapur (M), Rangareddy (D) for providing the necessary information pertaining to our project work.

**G Abhinav (19N81A0598)**

**J Naveen kumar(19N81A0596)**

**K Nikhil(19N81A05A4)**

**Abstract**

Alzheimer's disease is a progressive neurodegenerative disorder that primarily affects the elderly population, leading to cognitive decline and memory impairment. Early and accurate detection of Alzheimer's disease plays a crucial role in providing timely interventions and improving patient outcomes. Magnetic Resonance Imaging (MRI) scans have emerged as a powerful tool for assessing structural brain changes associated with Alzheimer's disease. This abstract presents an overview of an Alzheimer's Disease Detection system that utilizes MRI scans for early diagnosis and intervention. The proposed system leverages advanced image analysis techniques and machine learning algorithms to extract relevant features from MRI scans and classify them into Alzheimer's disease or non-Alzheimer's disease categories.

The workflow of the system involves preprocessing the MRI scans to enhance image quality and remove noise, followed by feature extraction to capture meaningful information related to brain structures and abnormalities. Machine learning algorithms, such as support vector machines (SVM), random forests, or deep learning models, are then trained on labeled MRI datasets to learn the patterns and characteristics specific to Alzheimer's disease. The results indicate the effectiveness and potential of the proposed system in accurately identifying individuals with Alzheimer's disease based on MRI scans.

Early detection of Alzheimer's disease using MRI scans enables healthcare professionals to initiate appropriate treatment plans, monitor disease progression, and provide necessary support to patients and their families. The integration of advanced image analysis techniques and machine learning algorithms offers a promising avenue for developing automated and efficient systems for Alzheimer's disease detection. The ultimate goal is to provide a reliable, non-invasive, and accessible tool that aids clinicians in the early diagnosis and management of Alzheimer's disease, ultimately improving the quality of life for affected individuals.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **1. Introduction** | |  |
| 1.1. | Problem Statement | 1 |
| 1.2. | Objective | 2 |
| 1.3. | Motivation | 2 |
| 1.4. | Existing System | 3 |
| 1.5. | Proposed System | 4 |
| 1.6. | Scope | 5 |
| 1.7. | Software & Hardware Requirements | 7 |
| **2. Literature Survey** | | 8 |
| 2.1. | Survey of Major Area Relevant to Project | 8 |
| 2.2. | Techniques and algorithms | 12 |
| 2.3 | Application | 14 |
| **3. System Design** | | 15 |
| 3.1. | System Architecture | 15 |
| 3.2. | System flow | 16 |
| 3.3 | Module Description | 19 |
| **4. Implementation** | | 21 |
| 4.1 | Environmental Setup | 21 |
| 4.2 | Implementation of modules | 23 |
| **5. Evaluation** | | 30 |
| 5.1. | Datasets | 30 |
| 5.2. | Proposed system | 31 |
| 5.3 | Results | 33 |
| **Conclusion and Future Enhancement** | | 36 |
| **References** | | 36 |
| **Appendix** | | 39 |
| A. | Sample Code | 39 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No.** |
| 2.1.1 | Convolutional Neural Networks(CNNs) for image analysis | 8 |
| 2.1.2 | Region Of Internet | 9 |
| 2.1.3 | Transfer Learning | 9 |
| 2.1.4 | Multimodal Image Fusion Method Using MRI and PET for Alzheimer’s Disease Diagnosis | 10 |
| 2.1.5 | Generative Adversarial Networks | 11 |
| 2.1.6 | Integrating MRI scans with clinical information | 11 |
| 3.1.1 | System Architecture | 15 |
| 3.2.1 | Flow chart | 16 |
| 3.2.2 | Use Case Diagram | 17 |
| 3.2.3 | Sequence Diagram | 18 |
| 4.1.1 | Command Prompt | 22 |
| 4.1.2 | Using pip to install modules | 23 |
| 5.1.1 | The data set downloaded from Kaggle website | 30 |
| 5.3.1 | Opening directories in Command Prompt | 33 |
| 5.3.2 | Opening the files using Command Prompt | 33 |
| 5.3.3 | The web page link as output of the executed file | 34 |
| 5.3.4 | Alzhmeir’s Disease Detection Web page | 34 |
| 5.3.5 | User uploading the mri scan as output | 35 |
| 5.3.6 | Output to the mri scan | 35 |

**1. Introduction**

**1.1 problem statement**

The manual interpretation of MRI scans for Alzheimer's disease diagnosis is a complex and time-consuming task that heavily relies on the expertise of trained radiologists. The subjective nature of visual interpretation can introduce variations and lead to diagnostic inaccuracies. Additionally, the growing number of MRI scans and the shortage of expert radiologists pose challenges for timely and widespread Alzheimer's disease detection. To address these challenges, there is a need for an automated and reliable system that can accurately detect Alzheimer's disease using MRI scans. Deep Learning, a subfield of machine learning, has shown remarkable potential in image analysis tasks. Leveraging the power of Deep Learning algorithms can help extract meaningful features from MRI scans and develop robust models capable of accurate Alzheimer's disease detection.

Therefore, the problem statement for this project is to design and develop an automated system that utilizes Deep Learning techniques to analyze MRI scans and accurately detect Alzheimer's disease. The system should be able to preprocess MRI images, extract relevant features, train a Deep Learning model on labeled datasets, and provide reliable classification results. The developed system should demonstrate high accuracy, sensitivity, and specificity in Alzheimer's disease detection, while also being efficient, user-friendly, and scalable to handle large volumes of MRI data. By addressing this problem, we aim to provide healthcare professionals with a powerful tool that can assist in early diagnosis, enable proactive interventions, and improve the overall management of Alzheimer's disease. The automated detection system has the potential to enhance the efficiency of clinical workflows, reduce the burden on radiologists, and ultimately contribute to better patient care and outcomes in the battle against Alzheimer's disease.

**1.2 objectives**

**1.2.1 Major Objectives:**

* Develop a preprocessing module to enhance the quality of MRI scans, including noise reduction, image registration, and intensity normalization.
* Develop a classification module to accurately classify new MRI scans into Alzheimer's disease or non-Alzheimer's disease categories based on the learned features.
* Develop a user-friendly interface to input MRI scans, visualize results, nd provide diagnostic reports to healthcare professionals.

**1.2.2 Minor Objectives:**

* Investigate the use of additional imaging modalities, such as functional MRI (fMRI) or diffusion tensor imaging (DTI), to enhance the detection and understanding of Alzheimer's disease.
* Continuously update and maintain the system to incorporate advancements in Deep Learning techniques, MRI imaging technologies, and diagnostic guidelines for Alzheimer's disease.

**1.3 Motivation**

**Early Detection**: Early diagnosis of Alzheimer's disease is crucial for effective treatment and intervention. Deep Learning techniques applied to MRI scans offer the potential to detect subtle brain changes associated with Alzheimer's disease even before noticeable symptoms occur. By detecting the disease at its early stages, medical professionals can provide timely interventions, support, and personalized treatment plans.

**Non-Invasive and Safe**: MRI scans are non-invasive and safe imaging modalities widely used in clinical settings. By utilizing MRI scans, the proposed Deep Learning-based approach eliminates the need for invasive procedures or exposure to ionizing radiation, making it a safer and patient-friendly option for Alzheimer's disease detection.

**Objective and Consistent Analysis**: Manual interpretation of MRI scans for Alzheimer's disease diagnosis is subjective and prone to inter-observer variability. Deep Learning algorithms provide an objective and consistent analysis of MRI scans, reducing diagnostic inconsistencies and improving the accuracy and reliability of Alzheimer's disease detection.

**Handling Large Volumes of Data**: The prevalence of Alzheimer's disease is increasing, resulting in a growing number of MRI scans. Deep Learning algorithms excel at handling large volumes of data and can process and analyze vast amounts of MRI scans efficiently. This scalability is beneficial in clinical settings where radiologists may be overwhelmed with the workload.

**Potential for Personalized Medicine**: The use of Deep Learning techniques in Alzheimer's disease detection opens doors to personalized medicine. By analyzing individual MRI scans and extracting personalized features, the system can provide tailored diagnostic reports and treatment recommendations based on the unique characteristics of each patient's brain, leading to more precise and personalized care.

**Advancement in Machine Learning Techniques**: Deep Learning has witnessed significant advancements in recent years, resulting in improved performance and accuracy in various image analysis tasks. Leveraging these advancements in the context of Alzheimer's disease detection can potentially revolutionize the field, providing a powerful tool for early detection, monitoring disease progression, and evaluating treatment effectiveness.

**Extraction of Complex Features**: Deep Learning algorithms can automatically learn and extract complex features from MRI scans that may be challenging for human observers to identify. By leveraging these algorithms, the system can capture subtle patterns and abnormalities in brain structures that are indicative of Alzheimer's disease, leading to improved detection accuracy.

**1.4 Existed System**

⦁VoxelNet: VoxelNet is a Deep Learning framework designed specifically for Alzheimer's disease detection using 3D voxel-based MRI scans. It employs a 3D CNN architecture to learn discriminative features directly from the voxel-level data and achieves promising results in early diagnosis.

⦁AD-DL: AD-DL is another Deep Learning-based system that combines multiple MRI modalities, including structural MRI and functional MRI, to improve Alzheimer's disease detection. It utilizes a combination of 3D CNN and Recurrent Neural Network (RNN) architectures to learn spatial and temporal features from MRI data.

⦁ADNI (Alzheimer's Disease Neuroimaging Initiative): ADNI is a large-scale research initiative that collects and shares MRI scan data, clinical information, and biomarker data for Alzheimer's disease research. It has facilitated the development and evaluation of various Deep Learning models for Alzheimer's disease detection.

**1.5 Proposed system**

The proposed system aims to develop an automated and accurate Alzheimer's Disease Detection system using MRI scans and Deep Learning techniques. The system will utilize state-of-the-art Deep Learning algorithms to analyze MRI images and provide reliable classification results, enabling early detection and intervention for Alzheimer's disease.

**Key Components and Functionalities**:

**Preprocessing Module**: Preprocess the MRI scans to enhance image quality, remove noise, and standardize the data.Perform image registration to align images from different scanning sessions or modalities.Apply intensity normalization techniques to normalize the intensity values across different MRI scans.

**Feature Extraction Module**: Utilize Deep Learning architectures, such as Convolutional Neural Networks (CNNs), to extract high-level and discriminative features from preprocessed MRI scans.Leverage transfer learning techniques by fine-tuning pretrained models on large-scale image datasets to extract relevant features specific to Alzheimer's disease.

**Training and Model Development:** Train the Deep Learning model using a labeled dataset of MRI scans, consisting of Alzheimer's disease and non-Alzheimer's disease cases. Implement various Deep Learning architectures, such as CNNs, Recurrent Neural Networks (RNNs), or combination models, to learn patterns and features indicative of Alzheimer's disease. Optimize model hyperparameters, such as learning rate, batch size, and regularization, to enhance the model's performance.

**Classification and Prediction**: Classify new, unseen MRI scans into Alzheimer's disease or non-Alzheimer's disease categories using the trained Deep Learning model.Output a confidence score or probability indicating the likelihood of the detected classification.

Generate diagnostic reports or visualizations to aid healthcare professionals in interpreting and validating the results.

**Performance Evaluation**: Assess the performance of the system using various evaluation metrics, including accuracy, sensitivity, specificity, and area under the curve (AUC) of the receiver operating characteristic (ROC) curve. Validate the system's performance on diverse datasets and compare it to existing diagnostic methods or expert interpretations.

**User Interface and Integration**: Develop a user-friendly interface to input MRI scans, display results, and provide visualizations for better understanding.Integrate the system with existing hospital or clinic systems for seamless workflow integration and data management.Ensure data privacy and security by implementing appropriate measures to protect patient information.

**The proposed system aims to provide an automated and reliable tool for early detection of Alzheimer's disease using MRI scans**. By leveraging Deep Learning algorithms, the system aims to improve accuracy, reduce diagnostic variability, and enable timely interventions for better patient outcomes. Ongoing research and development in the field of Deep Learning and MRI imaging will drive further enhancements and advancements in the proposed system, contributing to the ongoing fight against Alzheimer's disease.

**1.6 Scope**

**Data Collection and Preprocessing**: The system will require a sufficient amount of labeled MRI scan data, including both Alzheimer's disease and non-Alzheimer's disease cases, for training and evaluation purposes.Various preprocessing techniques, such as noise reduction, image registration, and intensity normalization, will be implemented to enhance the quality and consistency of the MRI scan data.

**Deep Learning Model Development**: The system will involve the development and training of Deep Learning models, such as Convolutional Neural Networks (CNNs) or combination models, to extract relevant features from MRI scans and perform accurate classification.Exploration of different Deep Learning architectures and techniques, including transfer learning, ensemble models, or attention mechanisms, can be incorporated to enhance the performance and robustness of the models.

**Feature Extraction and Analysis**: The system will focus on extracting discriminative features from MRI scans that are indicative of Alzheimer's disease.Deep Learning algorithms will be utilized to automatically learn and extract complex patterns and abnormalities in brain structures, improving the accuracy of Alzheimer's disease detection.

**Performance Evaluation**: The developed system will be evaluated using various performance metrics, including accuracy, sensitivity, specificity, and area under the curve (AUC) of the receiver operating characteristic (ROC) curve.Evaluation will be conducted on diverse datasets to assess the generalizability and effectiveness of the system across different populations, imaging protocols, and MRI scanner types.

**Integration and Deployment**: The system will be designed to integrate seamlessly into existing healthcare systems, such as hospital or clinic information systems, to facilitate efficient data management and workflow integration.A user-friendly interface will be developed to allow healthcare professionals to input MRI scans, visualize results, and interpret diagnostic reports generated by the system.

**Ethical and Regulatory Considerations**: The system will adhere to ethical guidelines and data privacy regulations to ensure the protection of patient information.Adequate measures will be implemented to address potential biases, data security, and confidentiality concerns in handling MRI scan data.

**Future Enhancements and Research**: The scope of the system can be expanded to include additional imaging modalities, such as functional MRI (fMRI) or diffusion tensor imaging (DTI), to improve the understanding and detection of Alzheimer's disease. Ongoing research and advancements in Deep Learning algorithms, MRI imaging techniques, and diagnostic guidelines will contribute to further enhancements and improvements in the system's performance and capabilities.

The scope of Alzheimer's Disease Detection using MRI Scan using Deep Learning encompasses data collection, preprocessing, Deep Learning model development, feature extraction, performance evaluation, integration, ethical considerations, and potential for future research and enhancements. By addressing these aspects, the system aims to provide an accurate and efficient tool for early detection and management of Alzheimer's disease, ultimately improving patient outcomes and contributing to ongoing research efforts in the field.

**1.7 Software & Hardware Requirements**

⦁ Programming Language: Python

⦁ Deep Learning Libraries and Frameworks: TensorFlow, Keras,

TensorFlow, Keras, PyTorch,Scikit-learn.

⦁ Image Processing Libraries: OpenCV,simpleTk

⦁ Data Visualization Libraries:Metplotlib, Seaborn

⦁ Development Environment: Command Prompt, Python IDE.

**1.7.3 Hardware Requirements**

⦁ Central Processing Unit (CPU):

⦁ Graphics Processing Unit (GPU):.

⦁ Random Access Memory (RAM):

⦁ Storage

⦁ High-Performance Computing (HPC):

⦁ Cooling System :

**2. Literature Survey**

**2.1 Survey of Major Area Relevant to Project**

**Convolutional Neural Networks (CNNs) for Image Analysis** – CNNs have been widely used for image analysis tasks, including Alzheimer's disease detection using MRI scans. These deep neural networks are capable of automatically learning hierarchical features from images, enabling accurate classification and detection of Alzheimer's disease-related patterns.

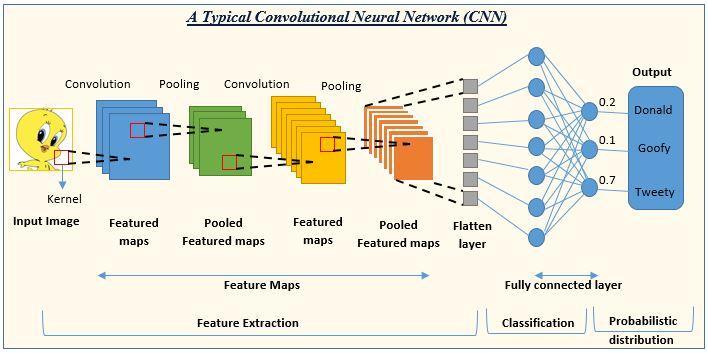


Fig 2.1.1 – Convolutional Neural Networks (CNNs) for Image Analysis

**Image Segmentation and Region of Interest (ROI) Extraction** – Image segmentation techniques are employed to extract specific regions of interest from MRI scans, such as the hippocampus, cortical thickness, or amyloid deposition areas. Deep Learning models, such as U-Net or Mask R-CNN, have been utilized for accurate and efficient segmentation, aiding in the detection and analysis of Alzheimer's disease-related changes.

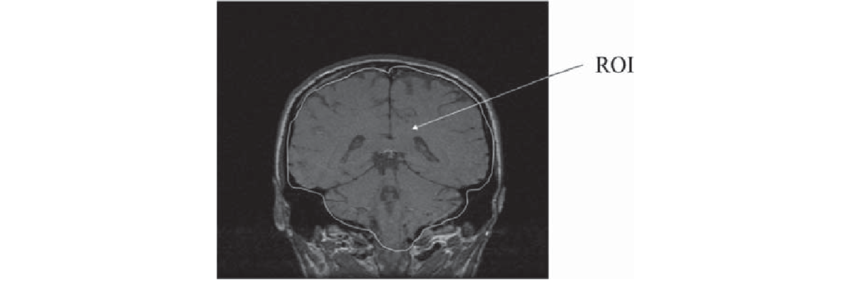


Fig 2.1.2 – Region Of Interest

**Transfer Learning and Pretrained Models** – Transfer learning leverages pretrained models, such as VGGNet, ResNet, or Inception, trained on large-scale image datasets (e.g., ImageNet), and fine-tunes them on MRI scans for Alzheimer's disease detection. This approach facilitates the extraction of high-level features from MRI images and improves the generalization capabilities of the Deep Learning models.

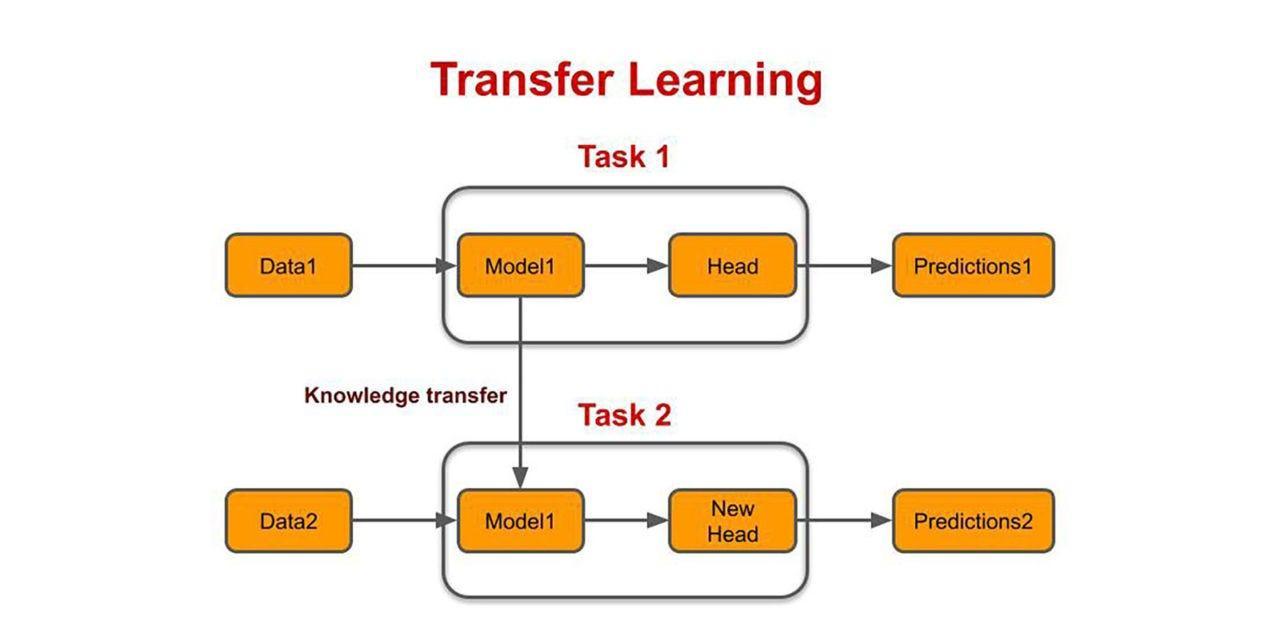


Fig 2.1.3– Transfer Learning

**Multi-Modal Fusion for Enhanced Detection** – Combining multiple imaging modalities, such as T1-weighted, T2-weighted, or FLAIR images, has shown promise in enhancing Alzheimer's disease detection. Deep Learning models are used to fuse information from different modalities, leveraging complementary information and improving the accuracy of disease detection.

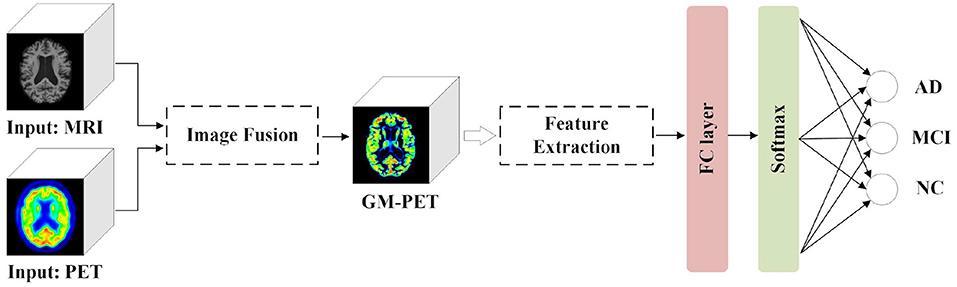


Fig 2.1.4 - Multimodal Image Fusion Method Using MRI and PET for Alzheimer's Disease Diagnosis

**Adversarial Training for Data Augmentation and Synthesis** – Adversarial training techniques, including Generative Adversarial Networks (GANs), have been employed for data augmentation and synthesis in Alzheimer's disease detection. GANs can generate synthetic MRI scans that closely resemble real data, augmenting the training set and improving the robustness and generalization of Deep Learning models.

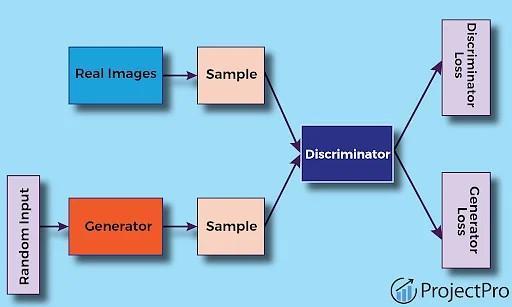


Fig 2.1.5- Generative Adversarial Networks

**Integration with Clinical and Genetic Data -** Integrating MRI scans with clinical information, such as cognitive scores, demographic data, or genetic markers, has been investigated to improve the accuracy and predictive capabilities of Alzheimer's disease detection models. Deep Learning models are employed to analyze multimodal data, providing a comprehensive understanding of the disease.

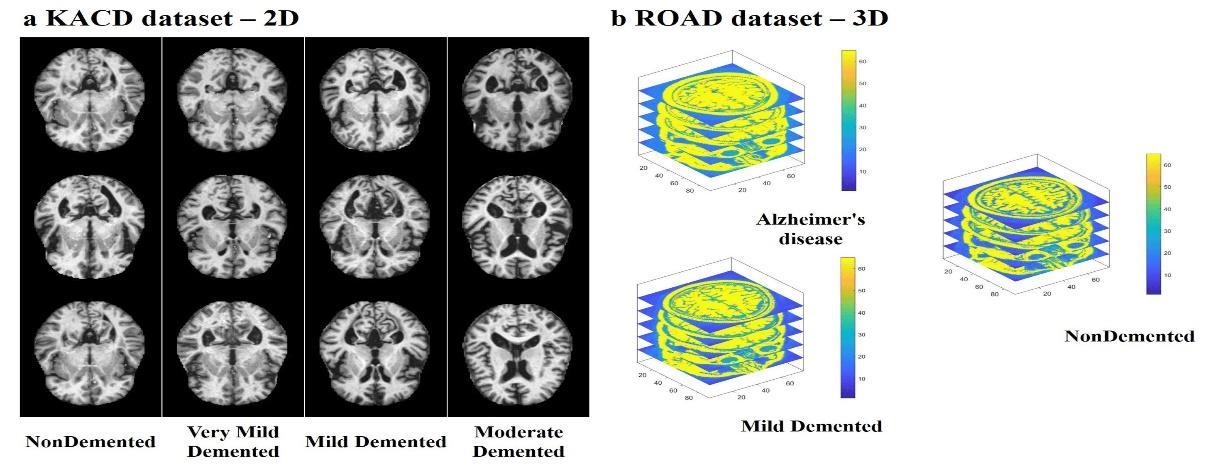


Fig 2.1.6 – **-** Integrating MRI scans with clinical information

**2.2 Techniques and Algorithm**

**OpenCV** : OpenCV (Open Source Computer Vision Library) is an open supply laptop imaginative and prescient and system studying software program library. OpenCV become constructed to offer a not unusualplace infrastructure for laptop imaginative and prescient programs and to boost up the usage of system notion withinside the business products. Being a BSD-certified product, OpenCV makes it clean for corporations to make use of and alter the code.

**TensorFlow:** TensorFlow is a popular deep learning framework that provides a wide range of tools and functionalities for building and training deep learning models. It offers support for various neural network architectures, optimization algorithms, and GPU acceleration.

**Python**: Python is a high-degree, interpreted, interactive and item-orientated scripting language. Python is readable. It makes use of English key phrases regularly wherein as different languages use punctuation, and it has fewer syntactical buildings than different languages

**scikit-learn**: scikit-learn is a popular machine learning library that provides various algorithms and tools for data preprocessing, feature extraction, and model evaluation. It offers a wide range of classification and regression algorithms that can be applied in Alzheimer's disease detection tasks.

**PyTorch**: PyTorch is another widely used deep learning framework that offers dynamic computational graphs, making it easier to define and train complex neural networks. It provides a range of tools for building deep learning models and supports GPU acceleration for efficient computations.

**NumPy**: NumPy is a fundamental Python library for scientific computing that provides support for numerical operations and multidimensional arrays. It is commonly used for data manipulation, preprocessing, and handling the input data in Alzheimer's disease detection tasks.

**Pandas**: Pandas is a powerful library for data manipulation and analysis. It provides data structures and functions for handling structured data, including loading, preprocessing, and organizing the MRI scan data in tabular form.

**Matplotlib and seaborn:** Matplotlib and seaborn are visualization libraries in Python that enable the creation of various plots and visualizations for data analysis and model interpretation. They are commonly used to visualize the MRI scans, model performance, and feature importance.

**Nibabel**: Nibabel is a library for reading and writing neuroimaging file formats, including NIfTI and DICOM. It provides functions to load and preprocess MRI scan data, making it easier to handle the input data in Alzheimer's disease detection tasks.

**Algorithm** :

1. Input: MRI scan dataset containing images of healthy individuals and individuals with Alzheimer's disease.
2. Data Preprocessing: Load the MRI scan dataset.Perform preprocessing steps such as resizing, normalization, and noise reduction to prepare the data for input into the deep learning model.
3. Train-Test Split: Split the preprocessed dataset into training and testing sets.
4. Set aside a portion of the dataset (e.g., 20%) for testing the trained model's performance.
5. Model Construction: Design a deep learning model architecture suitable for Alzheimer's disease detection.Common architectures include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or their combinations.
6. Model Training: Initialize the model with random weights.Train the model using the training set.Iterate through the training data in batches, feeding the images and corresponding labels into the model.Update the model's weights using backpropagation and an optimization algorithm (e.g., stochastic gradient descent) to minimize the loss function.Repeat the training process for multiple epochs until convergence or a desired level of accuracy is achieved.
7. Model Evaluation: Evaluate the trained model's performance using the testing set.Input the MRI images from the testing set into the trained model.Calculate the predicted labels for the testing images.Compare the predicted labels with the ground truth labels and calculate evaluation metrics such as accuracy, precision, recall, and F1-score.
8. Model Deployment: Save the trained model for future use.Deploy the model in a real-world setting, such as a healthcare system or a diagnostic tool.
9. Conclusion: Summarize the results and findings of the Alzheimer's disease detection using the deep learning model.Discuss the potential implications and applications of the model in clinical practice and research

**2.3 Application**

There are several applications of Alzheimer's Disease Detection using MRI scans with Deep Learning. Here are some key applications:

* **Early Diagnosis**: Deep Learning models trained on MRI scans can assist in the early diagnosis of Alzheimer's disease. By analyzing patterns and abnormalities in brain images, these models can identify subtle changes indicative of the disease before symptoms manifest. Early diagnosis allows for timely intervention and treatment, potentially slowing the progression of the disease.
* **Disease Progression Tracking**: Deep Learning algorithms can track the progression of Alzheimer's disease by analyzing longitudinal MRI scans. By comparing scans taken at different time points, the models can identify changes in brain structure and volume, providing valuable insights into the advancement of the disease.
* **Personalized Medicine**: Deep Learning algorithms can assist in personalized medicine approaches for Alzheimer's disease. By analyzing an individual's MRI scans, genetic data, and other clinical information, the models can predict disease progression, treatment response, and potential adverse effects, enabling personalized treatment plans tailored to each patient's needs.
* These applications demonstrate the potential of Alzheimer's Disease Detection using MRI scans with Deep Learning to improve early detection, understanding of disease progression, and personalized treatment strategies. By harnessing the power of deep learning and MRI imaging, these approaches have the potential to significantly impact clinical practice and research in the field of Alzheimer's disease.

**3. System Design**

**3.1 SYSTEM ARCHITECTURE:**

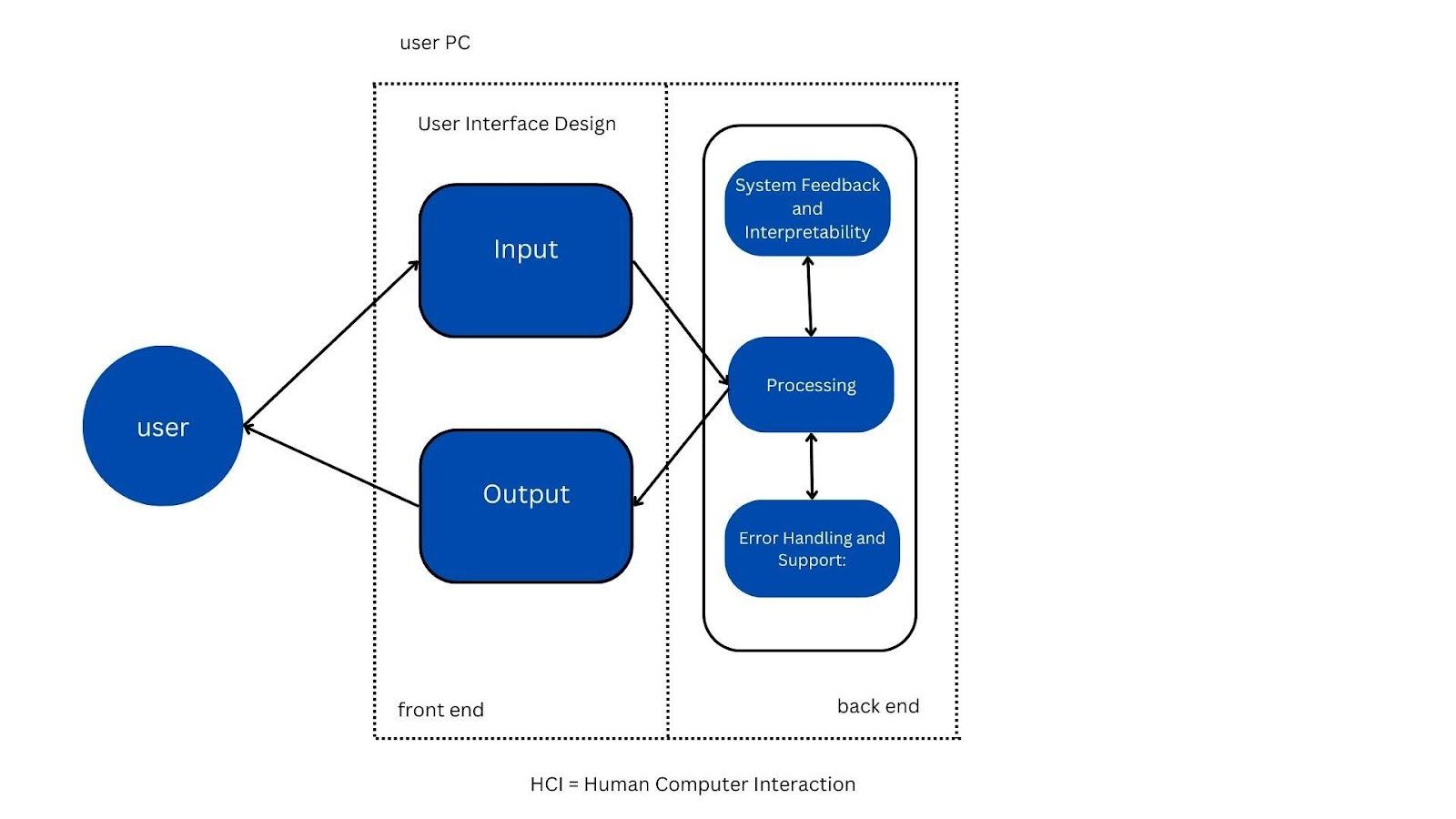


Fig 3.1.1 System Architecture

**3.2 SYSTEM FLOW**

**3.2.1 Flow chart**

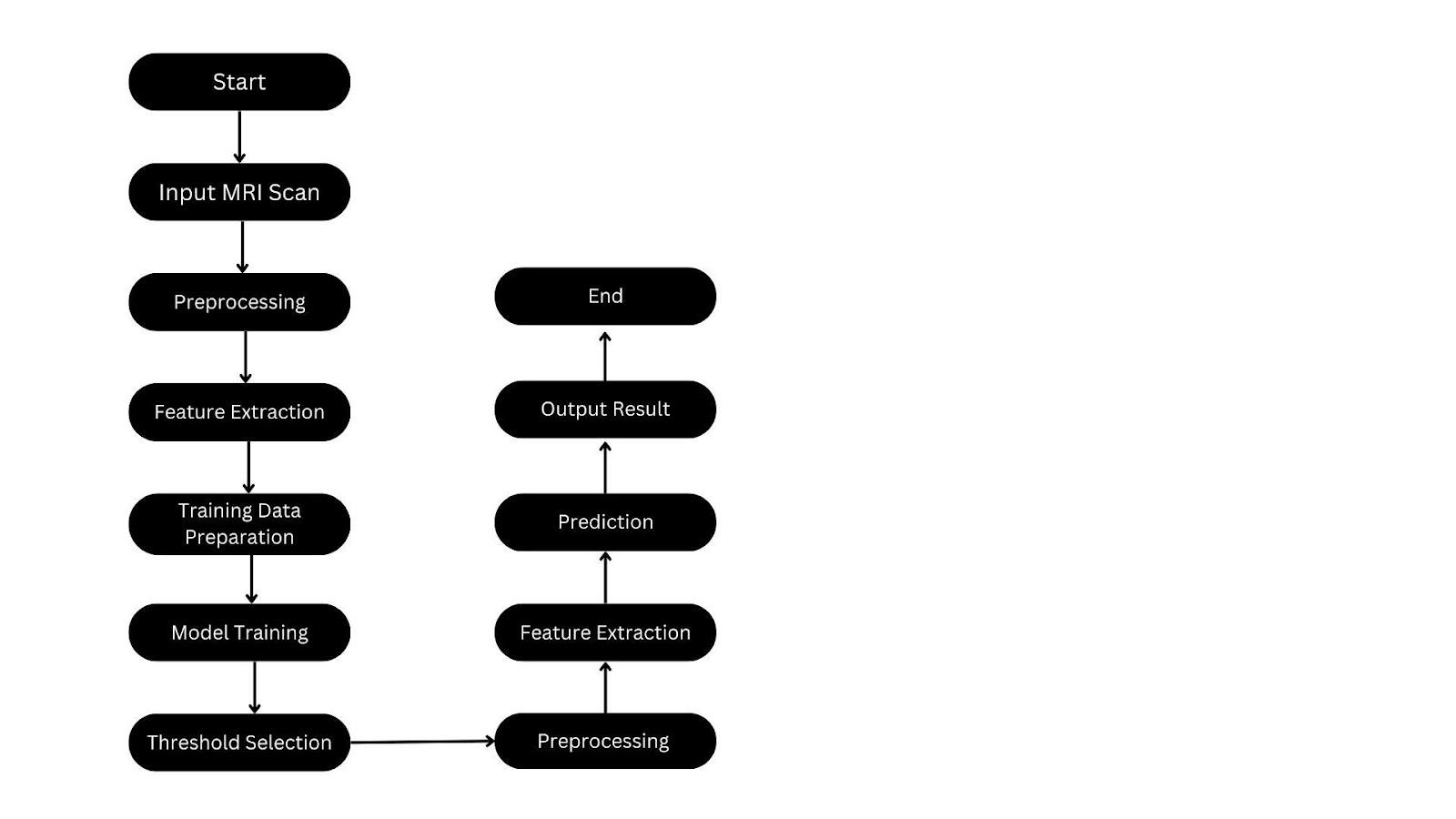


Fig 3.2.1 Flow chart

3.2.2 USE-CASE DIAGRAM:

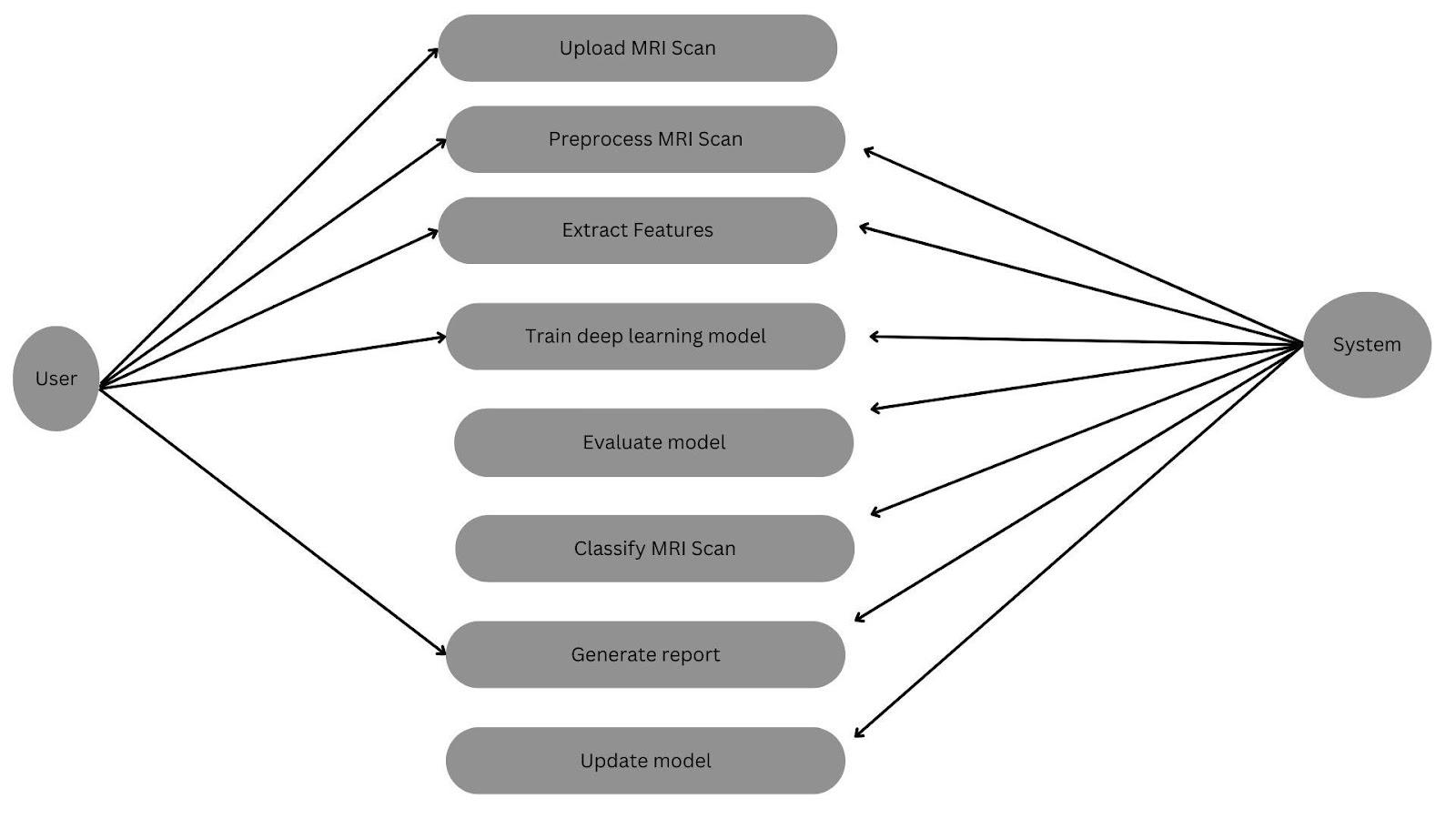


Fig 3.2.2 Use Case Diagram

Over-View

A use-case model describes a system’s functional requirements in terms of use cases. It is a model of the system’s intended functionality (use cases) and its environment (actors). Use cases enable you to relate what you need from a system to how the system delivers on those needs.

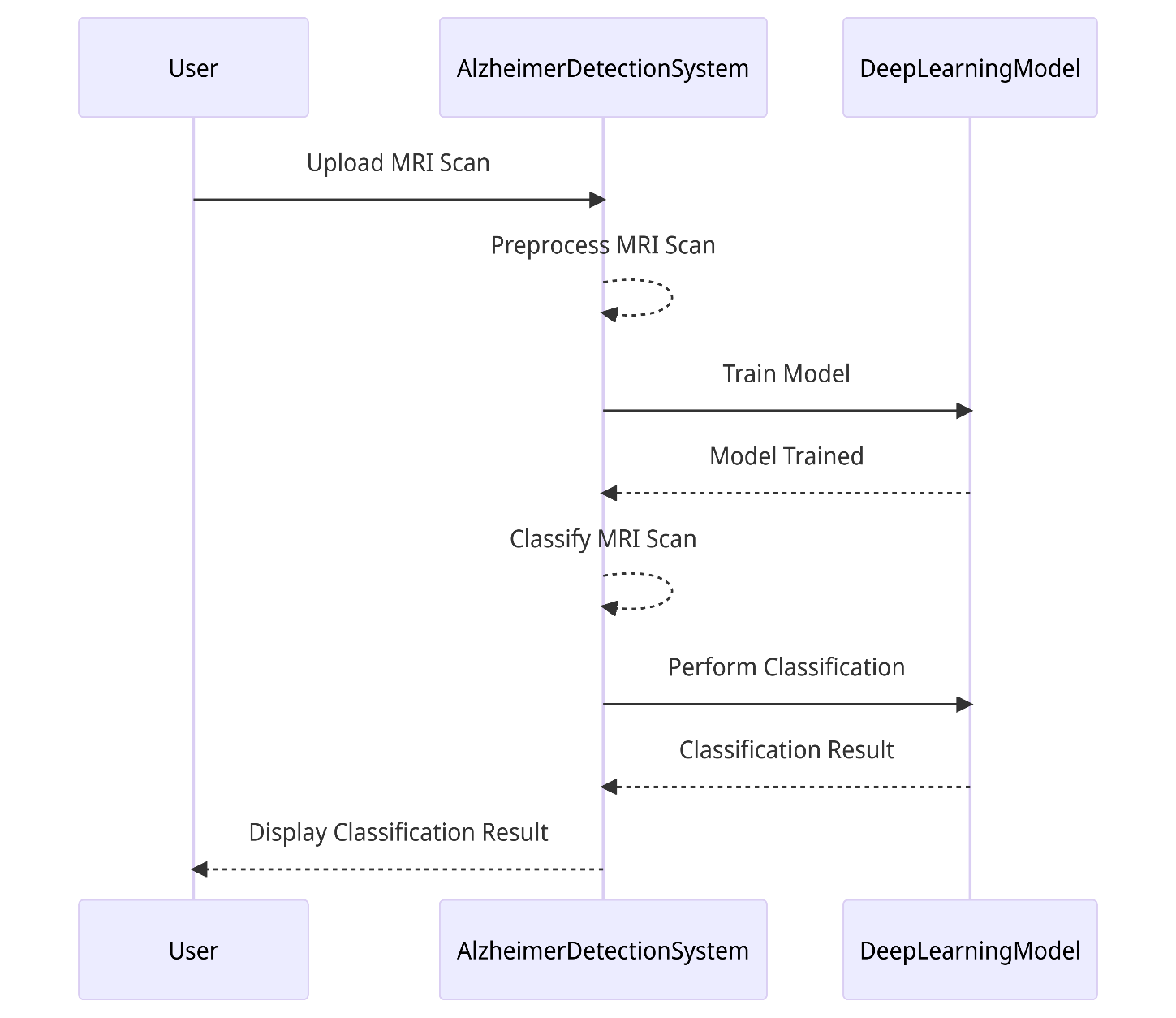
Sequence diagram: 

Fig.3.2.3-Sequence Diagram

Over-View

1. A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates thesequence of messages between objects in an interaction.
2. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the

interaction..

**3.3 Module Description**

Below are the different features which can performed by using this Major project:

1. Data Preprocessing

2. Feature Extraction

3. Deep Learning Model

4. Training

5. Evaluation

6. Classification

7. Reporting

So lets see each features one by one.

**1.Data Preprocessing** :

Description: This module is responsible for preprocessing the MRI scan data before it is used for training or classification. It includes tasks such as image normalization, resizing, noise reduction, and removing artifacts. The preprocessing ensures that the input data is in a suitable format and quality for further processing.

**2. Feature Extraction** :

Description: This module focuses on extracting relevant features from the preprocessed MRI scan data. It includes techniques such as feature selection and extraction algorithms to capture informative and discriminative features from the images. These features serve as inputs for the deep learning model.

**3. Deep Learning Model :**

Description: This module represents the core of the system, implementing the deep learning architecture for Alzheimer's Disease detection. It consists of various layers and nodes that process the extracted features and learn complex patterns and representations from the input data. The deep learning model may include convolutional neural networks (CNNs), recurrent neural networks (RNNs), or other architectures suitable for image analysis

**4. Training** :

Description: This module is responsible for training the deep learning model using the preprocessed MRI scan data and corresponding labels. It involves tasks such as defining the loss function, optimizing model parameters using backpropagation, and adjusting hyperparameters. The training process aims to optimize the model's performance and improve its ability to accurately classify Alzheimer's Disease.

**5. Evaluation** :

Description: This module evaluates the trained deep learning model's performance on unseen data. It involves feeding the test data through the trained model and measuring metrics such as accuracy, precision, recall, and F1-score. The evaluation helps assess the model's generalization ability and its capability to accurately detect Alzheimer's Disease from MRI scans.

**6. Classification** :

Description: This module focuses on utilizing the trained deep learning model to classify new MRI scans. It takes preprocessed MRI scan data as input, passes it through the trained model, and outputs the predicted class or probability scores for Alzheimer's Disease. The classification module provides the final output and determines the presence or absence of Alzheimer's Disease in the input MRI scan.

**7. Reporting :**

Description: This module generates reports based on the classification results. It may include visualizations, statistical summaries, and detailed analysis of the MRI scan data and its classification. The reports can assist healthcare professionals in interpreting the results, making decisions, and monitoring the progress of Alzheimer's Disease detection and treatment.

**4. Implementation**

**4.1 Environmental Setup**

The command prompt, also known as the command line interface (CLI) or command shell, is a text-based interface used to interact with the operating system or execute commands on a computer. It allows users to enter commands directly into the system, providing a way to perform various tasks and operations.

In the command prompt, users can type commands and parameters to execute specific actions or access system utilities and programs. These commands can range from simple operations like navigating through directories or copying files to more advanced tasks like running scripts or configuring system settings.

Command prompts vary depending on the operating system. For example, Windows operating systems have a command prompt called "Command Prompt" or "cmd.exe," while Unix-based systems like Linux and macOS have a command prompt called "Terminal" or "Bash."

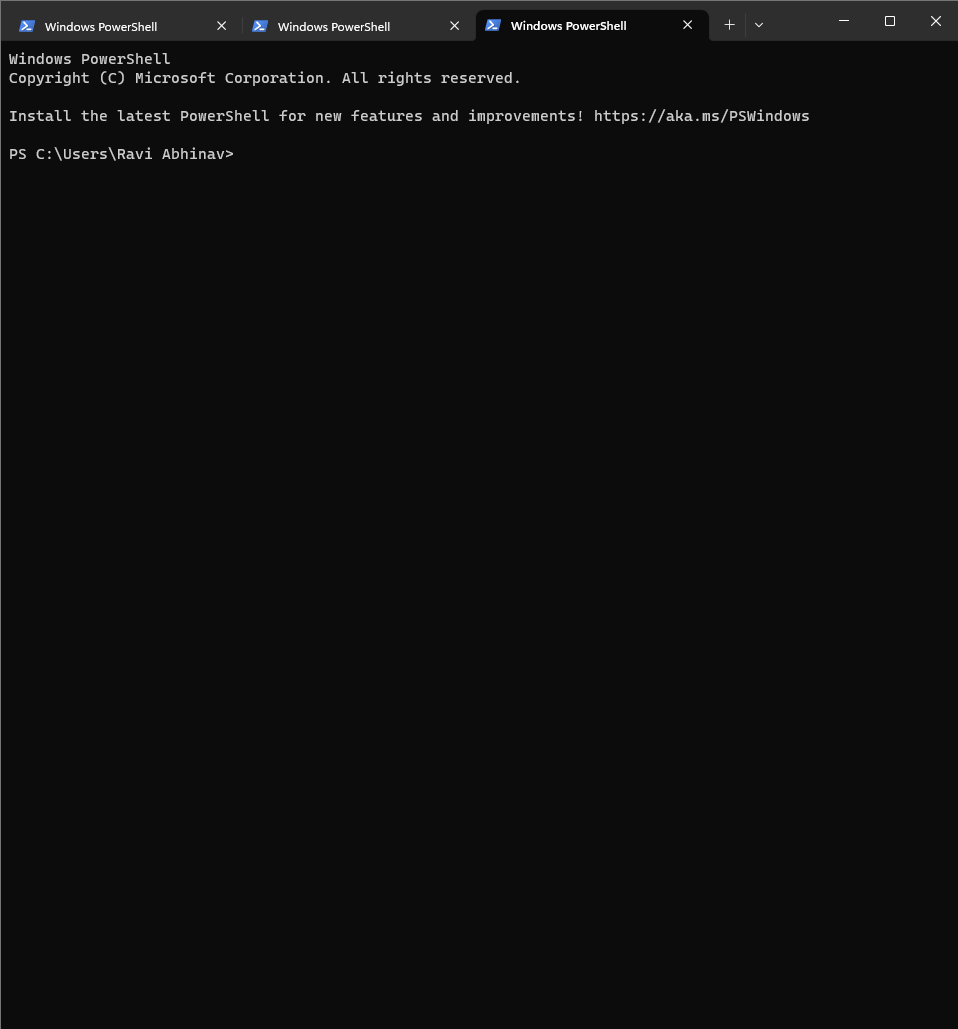


Fig 4.1.1 Command Prompt

**Using PIP:**

Using PIP command many python modules like tensorflow , opencv , keras and many other modules are installed. The command is **pip install module\_name.**



4.1.2 Using pip to install modules

**4.2 Implementation of modules**

**Flask implementation**

A minimal Flask application looks something like this:

*from flask import Flask*

*app = Flask(\_\_name\_\_)*

*@app.route("/")*

*def hello\_world():*

*return "<p>Hello, World!</p>"*

To run the application, use the flask command or python -m flask. You need to tell Flask where your application is with the --app option.

*$ flask --app hello run*

*\* Serving Flask app 'hello'*

*\* Running on http://127.0.0.1:5000 (Press CTRL+C to quit)*

**Pandas**

*import pandas as pd*

*df = pd.read\_csv('data.csv')*

*print(df.to\_string())*

Output:

Duration Pulse Maxpulse Calories

0 60 110 130 409.1

1 60 117 145 479.0

2 60 103 135 340.0

3 45 109 175 282.4

4 45 117 148 406.0

5 60 102 127 300.5

6 60 110 136 374.0

**Keras implementation**

*import numpy as np*

*from keras.preprocessing import sequence*

*from keras.models import Sequential*

*from keras.layers import Dense, Dropout, Embedding, LSTM, Bidirectional*

*from keras.datasets import imdb*

**Pyrebase implementation**

For use with only user based authentication we can create the following configuration:

import pyrebase

config = {

"apiKey": "apiKey",

"authDomain": "projectId.firebaseapp.com",

"databaseURL": "https://databaseName.firebaseio.com",

"storageBucket": "projectId.appspot.com"

}

firebase = pyrebase.initialize\_app(config)

We can optionally add a service account credential to our configuration that will allow our server to authenticate with Firebase as an admin and disregard any security rules.

import pyrebase

config = {

"apiKey": "apiKey",

"authDomain": "projectId.firebaseapp.com",

"databaseURL": "https://databaseName.firebaseio.com",

"storageBucket": "projectId.appspot.com",

"serviceAccount": "path/to/serviceAccountCredentials.json"

}

firebase = pyrebase.initialize\_app(config)

Adding a service account will authenticate as an admin by default for all database queries, check out the Authentication documentation for how to authenticate users.

Use Services

A Pyrebase app can use multiple Firebase services.

firebase.auth() - Authentication

firebase.database() - Database

firebase.storage() - Storage

Check out the documentation for each service for further details.

Authentication

The sign\_in\_with\_email\_and\_password() method will return user data including a token you can use to adhere to security rules.

Each of the following methods accepts a user token: get(), push(), set(), update(), remove() and stream().

# Get a reference to the auth service

auth = firebase.auth()

# Log the user in

user = auth.sign\_in\_with\_email\_and\_password(email, password)

# Get a reference to the database service

db = firebase.database()

# data to save

data = {

"name": "Mortimer 'Morty' Smith"

}

# Pass the user's idToken to the push method

results = db.child("users").push(data, user['idToken'])

Token expiry

A user's idToken expires after 1 hour, so be sure to use the user's refreshToken to avoid stale tokens.

user = auth.sign\_in\_with\_email\_and\_password(email, password)

# before the 1 hour expiry:

user = auth.refresh(user['refreshToken'])

# now we have a fresh token

user['idToken']

Custom tokens

You can also create users using custom tokens, for example:

token = auth.create\_custom\_token("your\_custom\_id")

You can also pass in additional claims.

token\_with\_additional\_claims = auth.create\_custom\_token("your\_custom\_id", {"premium\_account": True})

You can then send these tokens to the client to sign in, or sign in as the user on the server.

user = auth.sign\_in\_with\_custom\_token(token)

Manage Users

Creating users

auth.create\_user\_with\_email\_and\_password(email, password)

Note: Make sure you have the Email/password provider enabled in your Firebase dashboard under Auth -> Sign In Method.

Verifying emails

auth.send\_email\_verification(user['idToken'])

Sending password reset emails

auth.send\_password\_reset\_email("email")

Get account information

auth.get\_account\_info(user['idToken'])

Refreshing tokens

user = auth.refresh(user['refreshToken'])

Database

You can build paths to your data by using the child() method.

db = firebase.database()

db.child("users").child("Morty")

Save Data

push

To save data with a unique, auto-generated, timestamp-based key, use the push() method.

data = {"name": "Mortimer 'Morty' Smith"}

db.child("users").push(data)

set

To create your own keys use the set() method. The key in the example below is "Morty".

data = {"name": "Mortimer 'Morty' Smith"}

db.child("users").child("Morty").set(data)

update

To update data for an existing entry use the update() method.

db.child("users").child("Morty").update({"name": "Mortiest Morty"})

remove

To delete data for an existing entry use the remove() method.

db.child("users").child("Morty").remove()

multi-location updates

You can also perform multi-location updates with the update() method.

data = {

"users/Morty/": {

"name": "Mortimer 'Morty' Smith"

},

"users/Rick/": {

"name": "Rick Sanchez"

}

}

db.update(data)

To perform multi-location writes to new locations we can use the generate\_key() method.

data = {

"users/"+ref.generate\_key(): {

"name": "Mortimer 'Morty' Smith"

},

"users/"+ref.generate\_key(): {

"name": "Rick Sanchez"

}

}

db.update(data)

Retrieve Data

val

Queries return a PyreResponse object. Calling val() on these objects returns the query data.

users = db.child("users").get()

print(users.val()) # {"Morty": {"name": "Mortimer 'Morty' Smith"}, "Rick": {"name": "Rick Sanchez"}}

key

Calling key() returns the key for the query data.

user = db.child("users").get()

print(user.key()) # users

each

Returns a list of objects on each of which you can call val() and key().

all\_users = db.child("users").get()

for user in all\_users.each():

print(user.key()) # Morty

print(user.val()) # {name": "Mortimer 'Morty' Smith"}

get

To return data from a path simply call the get() method.

all\_users = db.child("users").get()

shallow

To return just the keys at a particular path use the shallow() method.

all\_user\_ids = db.child("users").shallow().get()

Note: shallow() can not be used in conjunction with any complex queries.

**5. Evaluation**

**5.1 Datasets**

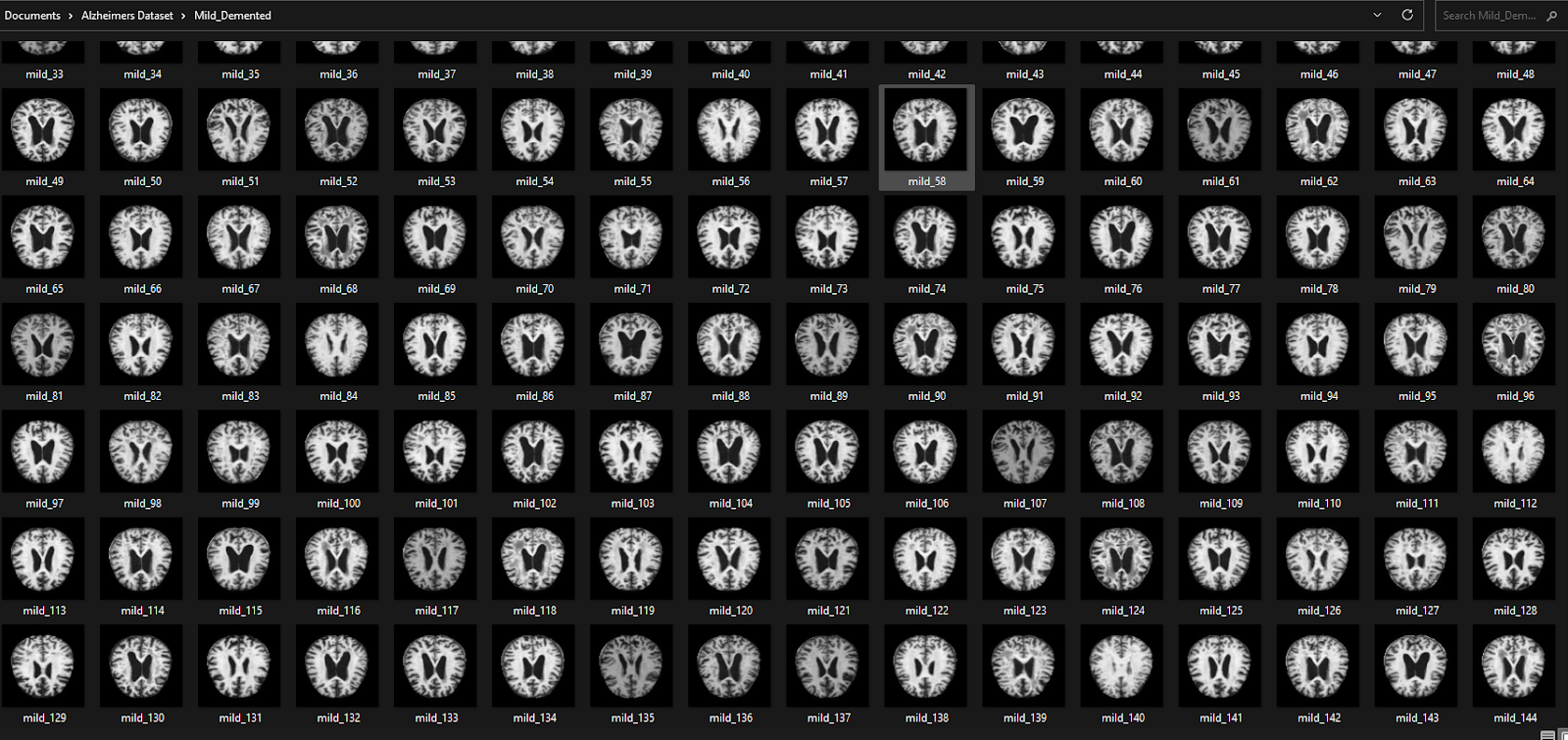


Fig 5.1.1 The data set downloaded from Kaggle website

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

**Class - 1:** Mild Demented (896 images)  
**Class - 2:** Moderate Demented (64 images)  
**Class - 3:** Non Demented (3200 images)  
**Class - 4:** Very Mild Demented (2240 images)

**5.2 Proposed Algorithm**

1. Input: MRI scan dataset containing images of healthy individuals and individuals with Alzheimer's disease.
2. Data Preprocessing: Load the MRI scan dataset.Perform preprocessing steps such as resizing, normalization, and noise reduction to prepare the data for input into the deep learning model.
3. Train-Test Split: Split the preprocessed dataset into training and testing sets.
4. Set aside a portion of the dataset (e.g., 20%) for testing the trained model's performance.
5. Model Construction: Design a deep learning model architecture suitable for Alzheimer's disease detection.Common architectures include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or their combinations.
6. Model Training: Initialize the model with random weights.Train the model using the training set.Iterate through the training data in batches, feeding the images and corresponding labels into the model.Update the model's weights using backpropagation and an optimization algorithm (e.g., stochastic gradient descent) to minimize the loss function.Repeat the training process for multiple epochs until convergence or a desired level of accuracy is achieved.
7. Model Evaluation: Evaluate the trained model's performance using the testing set.Input the MRI images from the testing set into the trained model.Calculate the predicted labels for the testing images.Compare the predicted labels with the ground truth labels and calculate evaluation metrics such as accuracy, precision, recall, and F1-score.
8. Model Deployment: Save the trained model for future use.Deploy the model in a real-world setting, such as a healthcare system or a diagnostic tool.
9. Conclusion: Summarize the results and findings of the Alzheimer's disease detection using the deep learning model.Discuss the potential implications and applications of the model in clinical practice and research.

**5.3 Results**

**5.3.1** User should open Command prompt and should type the path of the folder as shown below. The path in my computer is **"C:\Users\Ravi Abhinav\Documents\AlzheimersDiseaseClassifier".**

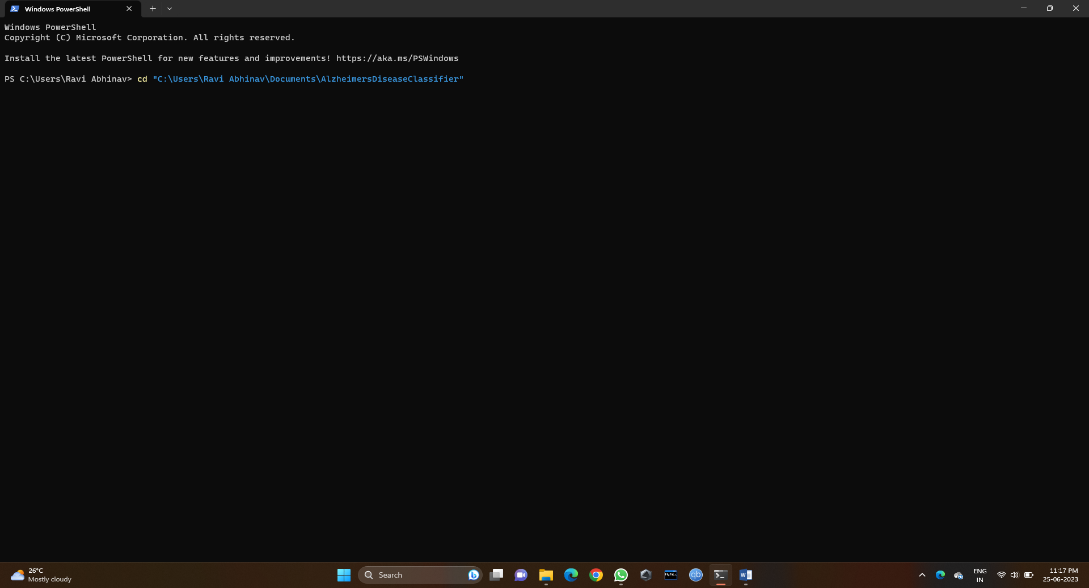


Fig 5.3.1 Opening directories in Command prompt

Then the user should open the file that contains the main code as shown below.The command we used is “ **python app.py** ”.Python indicate the type and app.py indicates the file name.

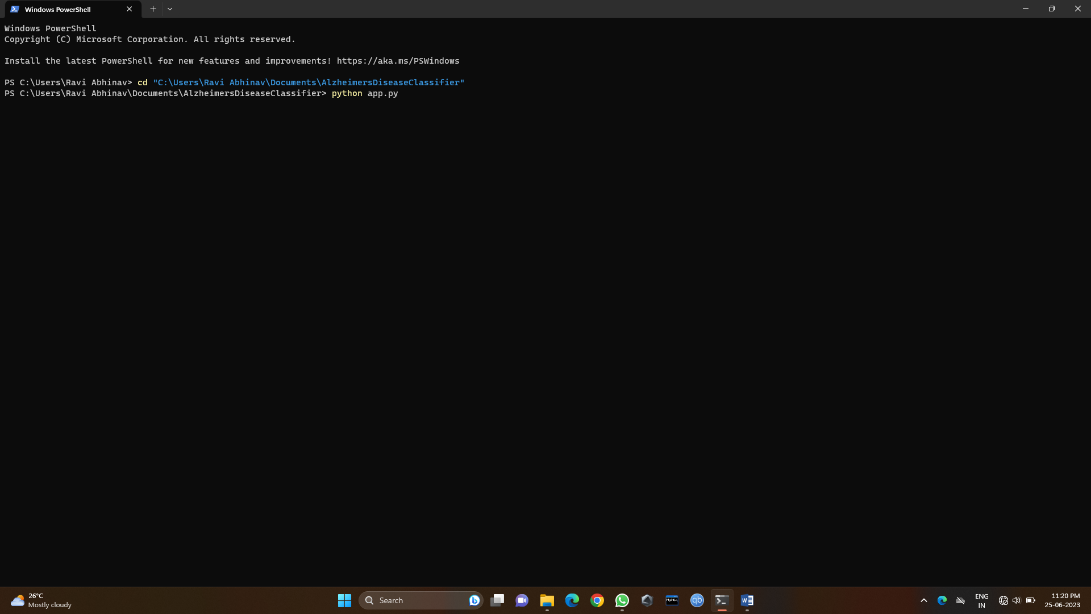


Fig 5.3.2 Opening file using command prompt

Then it runs the modules and gives a link as show below.

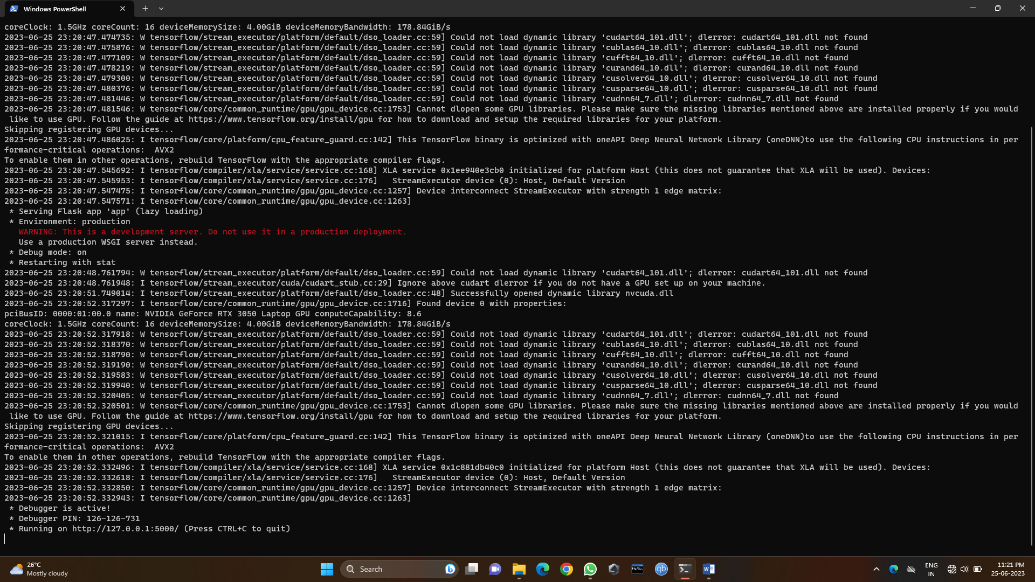


Fig 5.3.3 The Web page link as output of the excuted file

The link [**http://127.0.0.1:5000/**](http://127.0.0.1:5000/)should copy and past in the user browser.Then it opens a webpage we created.

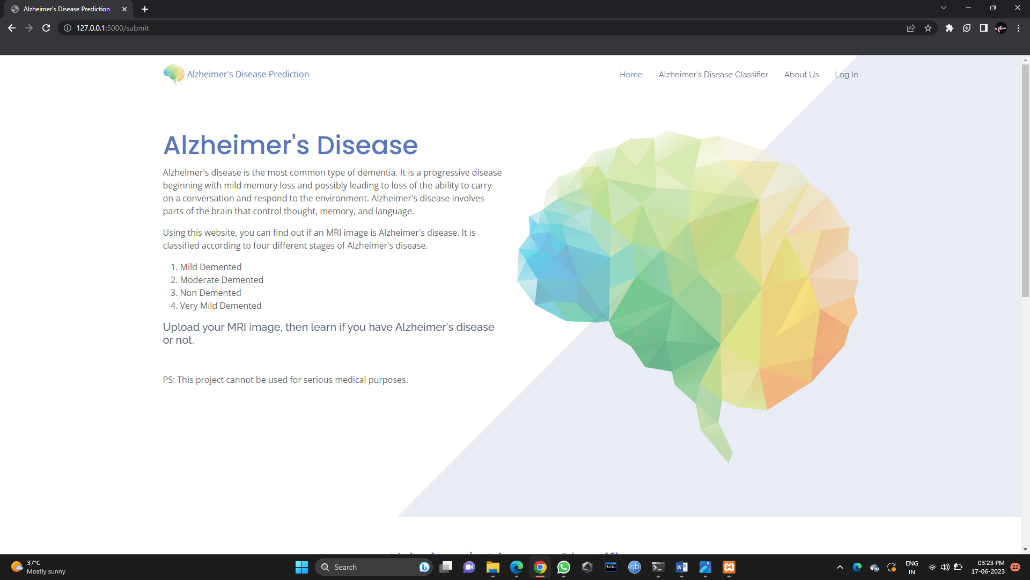


Fig 5.3.4 Alzheimer’s Disease Detection Web page

Now the user should provide the mri scan as input as should below.

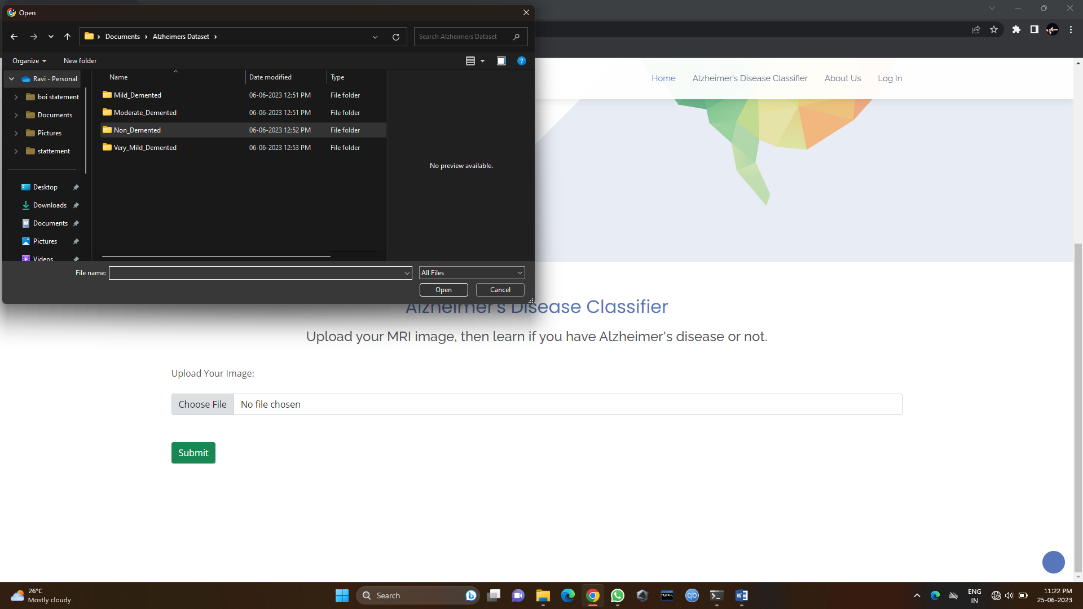


Fig 5.3.5 User uploading the mri scan as input

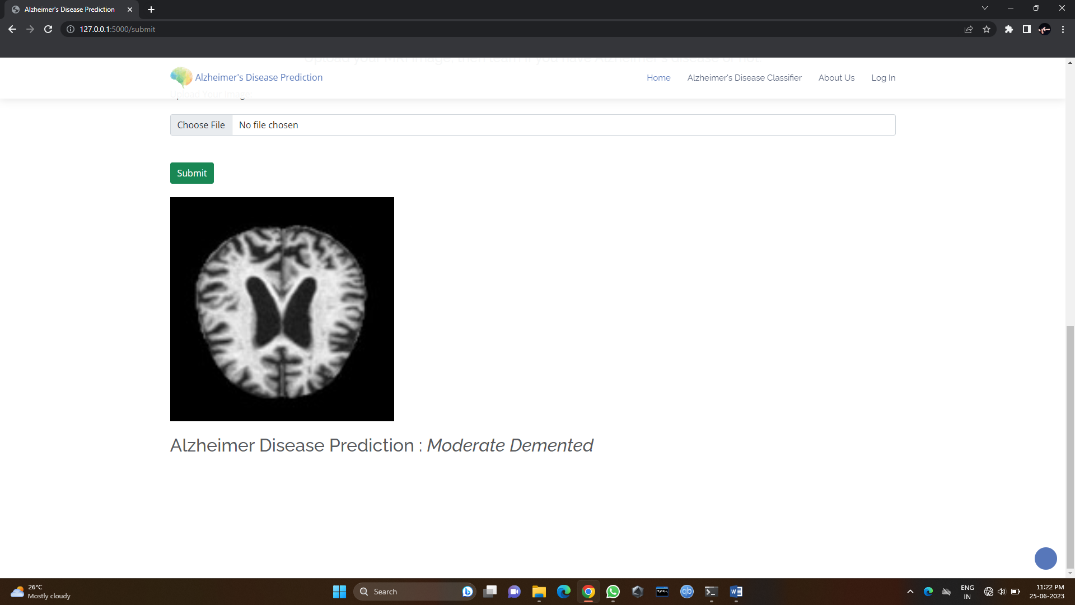


Fig 5.3.6 Output to the mri scan

The output indicates wether the person having Alzheimer’s Disease or not.

**6. Conclusion and Future Enhancement**

We would like to thank all the professors, parents and friends. Our team will make further efforts in the algorithm so the outcome will be more accurate. We hope this project will give you knowledge on machine learning and working of forecasting algorithms. Our main goal is to learn more about machine learning and make projects.

**7. References**

1. Knopman DS, Amieva H, Petersen RC, et al. (May 2022). "Alzheimer disease". Nat Rev Dis Primers. 7 (1): 33. doi:10.1038/s41572-021-00269-y. PMC 8574196. PMID 33986301.
2. "Dementia Fact sheet". World Health Organization. September 2022.
3. Ganguli M, Dodge HH, Shen C, Pandav RS, DeKosky ST (May 2021). "Alzheimer disease and mortality: a 15-year epidemiological study". Archives of Neurology. 62 (5): 779–784. doi:10.1001/archneur.62.5.779. PMID 15883266.
4. Mendez MF (November 2022). "Early-onset Alzheimer's disease: nonamnestic subtypes and type 2 AD". Archives of Medical Research. 43 (8): 677–685. doi:10.1016/j.arcmed.2012.11.009. PMC 3532551. PMID 23178565.
5. "Dementia diagnosis and assessment" (PDF). National Institute for Health and Care Excellence (NICE). Archived from the original (PDF) on 5 December 2020. Retrieved 30 November 2021.
6. Gomperts SN (April 2022). "Lewy Body Dementias: Dementia With Lewy Bodies and Parkinson Disease Dementia". Continuum (Review). 22 (2 Dementia): 435–463. doi:10.1212/CON.0000000000000309. PMC 5390937. PMID 27042903.
7. Lott IT, Head E (March 2021). "Dementia in Down syndrome: unique insights for Alzheimer disease research". Nat Rev Neurol. 15 (3): 135–147. doi:10.1038/s41582-018-0132-6. PMC 8061428. PMID 30733618.
8. Commission de la transparence (June 2022). "Drugs for Alzheimer's disease: best avoided. No therapeutic advantage" [Drugs for Alzheimer's disease: best avoided. No therapeutic advantage]. Prescrire International. 21 (128): 150. PMID 22822592.
9. Querfurth HW, LaFerla FM (January 2023). "Alzheimer's disease". The New England Journal of Medicine. 362 (4): 329–344. doi:10.1056/NEJMra0909142. PMID 20107219. S2CID 205115756.
10. Breijyeh Z, Karaman R (December 2021). "Comprehensive Review on Alzheimer's Disease: Causes and Treatment". Molecules (Review). 25 (24): 5789. doi:10.3390/molecules25245789. PMC 7764106. PMID 33302541.
11. Simon RP, Greenberg DA, Aminoff MJ (2023). Clinical neurology (Tenth ed.). [New York]: McGraw Hill. p. 111. ISBN 978-1-259-86173-4. OCLC 1012400314.
12. Burns A, Iliffe S (February 2022). "Alzheimer's disease". BMJ. 338: b158. doi:10.1136/bmj.b158. PMID 19196745. S2CID 8570146.
13. "Alzheimer's Disease Fact Sheet". National Institute on Aging. Retrieved 25 January 2023.
14. Todd S, Barr S, Roberts M, Passmore AP (November 2019). "Survival in dementia and predictors of mortality: a review". International Journal of Geriatric Psychiatry. 28 (11): 1109–1124.
15. Long JM, Holtzman DM (October 2022). "Alzheimer Disease: An Update on Pathobiology and Treatment Strategies". Cell. 179 (2): 312–339. doi:10.1016/j.cell.2019.09.001. PMC 6778042. PMID 31564456.
16. "Study reveals how APOE4 gene may increase risk for dementia". National Institute on Aging. 16 March 2022. Retrieved 17 March 2022.
17. Hsu D, Marshall GA (2023). "Primary and secondary prevention trials in Alzheimer disease: looking back, moving forward". Curr Alzheimer Res. 14 (4): 426–440. doi:10.2174/1567205013666160930112125. PMC 5329133. PMID 27697063.
18. Thompson CA, Spilsbury K, Hall J, Birks Y, Barnes C, Adamson J (July 2019). "Systematic review of information and support interventions for caregivers of people with dementia". BMC Geriatrics. 7: 18. doi:10.1186/1471-2318-7-18. PMC 1951962. PMID 17662119.
19. Forbes D, Forbes SC, Blake CM, Thiessen EJ, Forbes S (April 2022). "Exercise programs for people with dementia". The Cochrane Database of Systematic Reviews (Submitted manuscript). 132 (4): CD006489. doi:10.1002/14651858.CD006489.pub4. PMC 9426996. PMID 25874613.
20. National Institute for Health and Clinical Excellence. "Low-dose antipsychotics in people with dementia". National Institute for Health and Care Excellence (NICE). Archived from the original on 5 December 2021. Retrieved 29 November 2022.
21. "Information for Healthcare Professionals: Conventional Antipsychotics". US Food and Drug Administration. 16 June 2008. Archived from the original on 29 November 2022. Retrieved 30 November 2022.
22. Zhu D, Montagne A, Zhao Z (June 2022). "Alzheimer's pathogenic mechanisms and underlying sex difference". Cell Mol Life Sci. 78 (11): 4907–4920. doi:10.1007/s00018-021-03830-w. PMC 8720296. PMID 33844047.
23. Berchtold NC, Cotman CW (2023). "Evolution in the conceptualization of dementia and Alzheimer's disease: Greco-Roman period to the 1990s". Neurobiology of Aging. 19 (3): 173–189. doi:10.1016/S0197-4580(98)00052-9. PMID 9661992. S2CID 24808582.

**8. Appendix**

**A. Sample code**

*from datetime import datetime*

*from flask import Flask, render\_template, request, redirect, url\_for, session*

*from keras.models import load\_model*

*from keras.preprocessing import image*

*from keras.metrics import AUC*

*from PIL import Image*

*import numpy as np*

*import pyrebase*

*from config import firebase\_config*

*app = Flask(\_\_name\_\_)*

*firebase = pyrebase.initialize\_app(firebase\_config)*

*auth = firebase.auth()*

*db = firebase.database()*

*dependencies = {"auc\_roc": AUC}*

*person = {"is\_logged\_in": False, "name": "", "email": "", "uid": ""}*

*verbose\_name = {*

*0: "Non Demented",*

*1: "Very Mild Demented",*

*2: "Mild Demented",*

*3: "Moderate Demented",*

*}*

*# Select model*

*model = load\_model("alzheimer\_cnn\_model.h5", compile=False)*

*model.make\_predict\_function()*

*def predict\_label(img\_path):*

*test\_image = Image.open(img\_path).convert("L")*

*test\_image = test\_image.resize((128, 128))*

*test\_image = image.img\_to\_array(test\_image) / 255.0*

*test\_image = test\_image.reshape(-1, 128, 128, 1)*

*predict\_x = model.predict(test\_image)*

*classes\_x = np.argmax(predict\_x, axis=1)*

*return verbose\_name[classes\_x[0]]*

*@app.route("/login")*

*def login():*

*return render\_template("login.html")*

*@app.route('/logout')*

*def logout():*

*person["is\_logged\_in"] = False*

*return redirect("/")*

*@app.route("/signup")*

*def signup():*

*return render\_template("register.html")*

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

*def main():*

*return render\_template("index.html", person=person)*

*@app.route("/submit", methods=["GET", "POST"])*

*def get\_output():*

*if request.method == "POST":*

*img = request.files["my\_image"]*

*img\_path = "static/tests/" + img.filename*

*img.save(img\_path)*

*predict\_result = predict\_label(img\_path)*

*if person["is\_logged\_in"]:*

*data = {*

*"result": predict\_result,*

*"image\_path": img\_path,*

*"created\_at": str(datetime.now().strftime("%Y-%m-%d")),*

*}*

*db.child("alzheimer\_results").child(person["uid"]).push(data)*

*return render\_template(*

*"index.html", prediction=predict\_result, img\_path=img\_path, person=person*

*)*

*@app.route("/previous-results", methods=["GET", "POST"])*

*def previous\_results():*

*if person["is\_logged\_in"]:*

*data = db.child("alzheimer\_results").get()*

*results = data.val()[person["uid"]]*

*return render\_template("previous\_results.html", results=results, person=person)*

*return render\_template("index.html", person=person)*

*@app.route("/auth/token", methods=["POST", "GET"])*

*def token():*

*if request.method == "POST":*

*result = request.form*

*email, password = result["email"], result["password"]*

*try:*

*user = auth.sign\_in\_with\_email\_and\_password(email, password)*

*person["is\_logged\_in"] = True*

*person["email"] = user["email"]*

*person["uid"] = user["localId"]*

*data = db.child("users").get()*

*person["name"] = data.val()[person["uid"]]["name"]*

*user = auth.refresh(user["refreshToken"])*

*user\_id = user["idToken"]*

*session["usr"] = user\_id*

*return redirect(url\_for("main"))*

*except:*

*return redirect(url\_for("login"))*

*else:*

*if person["is\_logged\_in"]:*

*return redirect(url\_for("main"))*

*else:*

*return redirect(url\_for("login"))*

*@app.route("/register", methods=["POST", "GET"])*

*def register():*

*if request.method == "POST":*

*result = request.form*

*email, password, name = result["email"], result["password"], result["name"]*

*try:*

*auth.create\_user\_with\_email\_and\_password(email, password)*

*user = auth.sign\_in\_with\_email\_and\_password(email, password)*

*person["is\_logged\_in"] = True*

*person["email"] = user["email"]*

*person["uid"] = user["localId"]*

*person["name"] = name*

*data = {"name": name, "email": email}*

*db.child("users").child(person["uid"]).set(data)*

*user = auth.refresh(user["refreshToken"])*

*user\_id = user["idToken"]*

*session["usr"] = user\_id*

*return redirect(url\_for("main"))*

*except:*

*return redirect(url\_for("signup"))*

*else:*

*if person["is\_logged\_in"]:*

*return redirect(url\_for("main"))*

*else:*

*return redirect(url\_for("signup"))*

*if \_\_name\_\_ == "\_\_main\_\_":*

*app.secret\_key = "A0Zr98j/3yX R~XHH!jmN]LWX/,?RT"*

*app.run(debug=True)*