A
Major Project Report

On

"AI - POWERED BIOMEDICAL IMAGE ANALYSIS"

Submitted in partial fulfillment of the

Requirements for the award of the degree of

Bachelor of Technology

In

Computer Science & Information Technology

By

KASOJI KAVYASRI - 20R21A3321

Under the guidance of

Dr. P. Subhashini
Professor

Department of Computer Science & Information Technology

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS
MLR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMOUS)
Affiliated to JNTUH,Approved by AICTE
Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India

April, 2024

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS

MLR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

Affiliated to JNTUH,Approved by AICTE

Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India

NBA NATIONAL BOARD ACCREDITATION

NAAC A

Department of Computer Science & Information Technology

CERTIFICATE

This is to certify that the project entitled "**AI - POWERED BIOMEDICAL IMAGE ANALYSIS"** has been submitted by **Kasoji Kavyasri - (20R21A3321),** in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science & Information Technology from MLR Institute of Technology, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

Internal Guide	**Head of the Department**
Dr. P. SUBHASHINI	Dr. P. SUBHASHINI
Professor	Professor

External Examiner

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS

MLR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

Affiliated to JNTUH,Approved by AICTE

Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India

NBA NATIONAL BOARD ACCREDITATION

ACCREDITED WITH GRADE A NAAC

Department of Computer Science & Information Technology

DECLARATION

I here by declare that the project entitled **"AI - POWERED BIOMEDICAL IMAGE ANALYSIS"** is the work done during the period from **JULY 2023 to APRIL 2024** and is submitted in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science & Information Technology from MLR Institute of Technology, Hyderabad. The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

KASOJI KAVYASRI - 20R21A3321

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned my efforts with success. It is a pleasant aspect that I now have the opportunity to express my guidance for all of them.

First of all, I would like to express my deep gratitude towards my internal guide **Dr. P. SUBHASHINI, Professor, Department of CSIT** for her support in the completion of my dissertation. I wish to express my sincere thanks to **Dr. P. SUBHASHINI,** HOD, Dept. of CSIT and also principal **Dr. K. SRINIVAS RAO** for providing the facilities to complete the dissertation.

I would like to thank all my faculty and friends for their help and constructive criticism during the project period. Finally, I am very much indebted to my parents for their moral support and encouragement to achieve goals.

KASOJI KAVYASRI - 20R21A3321

ABSTRACT

AI Powered Biomedical Image Analysis, biomedical image analysis concept study plays a crucial part in the early and accurate diagnosis of ailments or diseases considerably impacting patient consequences. This study presents an creative approach to biomedical image analysis by integrating artificial intelligence and machine learning techniques accompanying a foolproof netting interface. The system is a complete website based which takes recommended images uploaded through the web connect and processes bureaucracy using leading artificial intelligence and machine learning algorithms executed in Python. The goal search out mechanize the detection of disease from these countenances, reinforcing the efficiency and accuracy of disease. Convolutional Neural Networks (CNNs), a specialized class of deep learning models devised for image analysis, are working to determine and characterize disease particular patterns. The proposed approach is judged utilizing various biomedical images including differing diseases and environments. Extensive experiments explain bureaucracy's effectiveness, gaining extreme accuracy, sympathy, and precision in disease discovery. The integration of artificial intelligence and machine learning concepts with a website connect not only simplifies the demonstrative process but again enables detached consultations, making health care duties more approachable and timely. In conclusion, this study presents a strong foundation for automated biomedical image analysis, professed the potential of machine learning and web technologies in revolutionizing health care condition. The system's veracity and user-friendly connect manage a valuable finish for health-care professionals, donating to early disease detection and enhanced patient care.

LIST OF FIGURES

LIST OF TABLES

INDEX

CHAPTER 1: INTRODUCTION

In the realm of healthcare, the intersection of biomedical image analysis and artificial intelligence (AI) has emerged as a transformative force, playing a pivotal role in the early and precise diagnosis of diseases. This project stands at the forefront of this intersection, presenting an innovative approach that seamlessly integrates machine learning techniques with a user-friendly web interface. By harnessing the power of advanced Python-based algorithms, this system automates the analysis of uploaded biomedical images, with a primary objective: to enhance the efficiency and precision of disease detection. The methodology involves a comprehensive process of pre-processing to refine image quality and eliminate noise, coupled with the utilization of deep learning models specifically designed for biomedical image analysis. These models not only extract intricate patterns and features but also harness transfer learning techniques to optimize performance, particularly in scenarios where labeled datasets are limited.

1.1 OVERVIEW

The AI-powered biomedical image analysis project represents a cutting-edge initiative aimed at revolutionizing disease diagnosis and treatment through advanced technology. Leveraging the capabilities of artificial intelligence and machine learning, the project focuses on automating the detection and classification of diseases from biomedical images, thereby significantly impacting patient outcomes. By integrating Convolutional Neural Networks (CNNs) into a user-friendly web interface, the system streamlines the diagnostic process, enabling healthcare professionals to efficiently analyze medical images and obtain accurate results. The project encompasses comprehensive datasets containing diverse biomedical images, meticulously curated to train and evaluate the CNN models. Through extensive experimentation and validation, the system achieves high accuracy, sensitivity, and specificity in disease detection, demonstrating its effectiveness in real-world healthcare scenarios.

1.2 PURPOSE OF THE PROJECT

The encompassing aim of this project search out expand an innovative plan for AI-stimulate biomedical image analysis, accompanying the primary objective of automating disease discovery to enhance the efficiency and accuracy in disease. Leveraging artificial intelligence and machine learning methods, specifically Convolutional Neural Networks (CNNs), the system will determine and distinguish disease-distinguishing patterns from uploaded biomedical figures via a convenient web interface. Through thorough test using different biomedical representations, the effectiveness of the projected approach will be judged, aiming to solve extreme accuracy, awareness, and accuracy in disease finding. Additionally, the unification of AI and machine learning accompanying computer network interface not only clarifies the demonstrative process but also authorizes detached consultations, ultimately donating to enhanced health-care accessibility and prompt mediations. By establishing a strong company for automated biomedical concept reasoning, this project aims to demonstrate the potential of machine intelligence and netting technologies in transforming health-care analyst, thereby upholding health-care specialists in early disease discovery and improving patient care.

1.3 MOTIVATION

The project aims to transform ailment diagnosis by leveraging AI and machine intelligence in biomedical figure study. Through a user-friendly netting connect, we inquire to mechanize disease discovery, improving veracity, and accessibility to health-care aids. By streamlining analyst and permissive remote consultations, our work strives to raise patient effects and address health-care differences, eventually contributing to a more active future. The project is driven by a belief in the transformative impact of technology on healthcare, aiming to usher in a new era of precision medicine. The user-friendly web interface is designed for accessibility, empowering healthcare professionals with a seamlessly integrated diagnostic solution. Additionally, the project reflects a dedication to contributing to research and innovation in biomedical image analysis, with the aim of shaping the future of diagnostic methodologies.

CHAPTER 2: LITERATURE SURVEY

The literature survey serves as a comprehensive exploration of existing research, providing valuable insights into the current state of knowledge, trends, challenges, and future directions within a specific field of study. It serves as the foundation for any research endeavor, offering researchers a roadmap to navigate through the vast landscape of scholarly works and build upon existing knowledge. In the context of our project on AI-powered biomedical image analysis, the literature survey delves into the dynamic field of medical imaging informatics and deep learning applications in medical image analysis. By synthesizing and critically analyzing relevant literature, the survey aims to identify key advancements, methodologies, and gaps in research, ultimately informing the development and implementation of our project.

2.1 AI and Medical Imaging Informatics: Current Challenges and Future Directions

2.1.1 DESCRIPTION

This informative page delves into the dynamic field of AI in Medical Imaging Informatics, providing a comprehensive exploration of both current challenges and future directions. The discussion begins by highlighting the multifaceted issues surrounding data quality and standardization, emphasizing the need for common protocols to enhance interoperability. Privacy and security concerns are addressed, acknowledging the sensitivity of medical imaging data and proposing potential solutions like robust anonymization techniques. The page delves into the interpretability and explainability challenges associated with AI models, emphasizing the importance of transparent decision-making for healthcare professionals. Integration into clinical workflows is examined, stressing the significance of seamless implementation to avoid disruption. Regulatory approval and validation processes are discussed in the context of evolving standards, calling for collaborative efforts to establish robust protocols. The page also touches on bias and fairness in AI models, underlining the necessity of diverse and representative training data. Looking to the future, the page outlines potential directions such as multimodal integration, real-time imaging analytics, and continuous learning for AI models. The importance of global collaboration, patient-centric AI, and ethical considerations are highlighted, contributing to a well-rounded exploration of the challenges and promising avenues in the evolving landscape of AI in Medical Imaging Informatics.

Year - [REF] Author	Disease	Imaging Data	Patients	DL Method	Segmentation/ Classification	Description
1995 - [100] Lo et *al*	Lung Cancer	X-ray	55	2 layer CNN	Nodules detection in a patch fashion	First ever attempt to use CNN for medical image analysis
2015 - [104] Ronneberger et *al*	Cells	Electron and optical microscopy	30 /35	U-net	Segmentation of EM images and cell tracking	Image to image tasks architecture depicting exceptional segmentation performance even with limited data
2016 - [118] Shin et *al*	Interstitial Lung Disease	CT	120 (905 slices)	Transfer learning (AlexNet, GoogleNet, CifarNet CNNs)	Interstitial lung disease binary classification	Showed that networks pre-trained on natural image data could be succesfully used on medical data
2016 - [122] Dou et *al*	Cerebral Microbleeds	MRI	320	Two-stage: 1) 3D Fully-convolutional network (FCN), 2) 3D CNN	3D FCN for candidate microbleed detection	A two-stage system used a 3D FCN to detect candidate microbleeds before a 3D CNN was applied to reduce false positives
2016 - [127] Setio *et al*	Pulmonary Cancer	CT	888 scans, 1186 nodules	Two-stage: 1) Feature-engineered candidate detector, 2) Multi-view 2D CNN for false positive reduction	Candidate pulmonary nodules detection	Significantly reduced false positives using fusion of multiple 2D CNNs at different views around a nodule
2017 - [268] Lekadir et *al*	Cardiovascular (carotid artery)	US	56 cases	Four convolutional and three fully connected layers	Characterization of carotid plaque composition	High correlation (0.90) with plaque composition clinical assessment for the estimation of lipid core, fibrous cap, and calcified tissue areas
2017 – [128] Yu et *al*	Melanoma	Dermoscopic Images	1250 images	Very deep (38/50/101 layers) fully conv. residual network	Binary melanoma classification	Used a very deep residual network (16 residual blocks) to classify melanoma
2017 - [102] Komnitsas et *al*	TBI, LGG/ GBM, Stroke	MRI	61 /110/ ISLES-SISS data	11-layers, multi-scale 3D CNN with fully connected CRF	Brain lesion segmentation algorithm	Top-performing segmentation results on TBI, brain tumours, and ischemic stroke at BRATS and ISLES 2015 challenges
2017 - [246] Lao et *al*	GBM	MRI	112	Transfer learning	Necrosis, enhancement, and edema tumour subregions	Overall survival prognostic signature for patients with Glioblastoma Multiforme (GBM)
2017 - [247] Oakden-Rayner et *al*	Overall Survival	CT (chest)	48	ConvNet transfer learning (3 convolutional and 1 fully connected layers)	Tissue (muscle, body fat, aorta, vertebral column, epicardial fat, heart, lungs)	Predict patients' 5-year mortality probability using radiogenomics data (overall survival)
2017 - [241] Zhu et *al*	Breast Cancer	DCE-MRI	270	Transfer learning (GoogleNet, VGGNet, CIFAR)	Breast tumour lesions	Discriminate between Luminal A and other breast cancer subtypes
2018 - [112] Chartsias et *al*	Cardiovascular	MRI	100	Various networks	Segmentation of cardiac anatomy	Limited training data when appropriate autonecoding losses are introduced
2020 – [121] McKinney et *al*	Breast Cancer	X-ray	25,856 & 3,097 cases	Ensemble and transfer learning	Breast cancer classification	Cancer prediction on two large datasets with comparison against human readers
2019 - [170] Hekler et *al*	Melanoma	Whole slide H&E tissue imaging	695	Transfer learning (ResNet50)	Binary melanoma classification	Human level performance in discriminating between nevi and melanoma images

US: Ultrasound; MRI: Magnetic Resonance Imaging; DCE-MRI: Dynamic Contrast Enhancement MRI; CT: Computed Tomography; PET: Positron Emission Tomography; GBM: Glioblastoma; LGG: Lower-Grade Glioma; CNN: Convolutional Neural Networks.

Fig 2.1.1: Selected DL Methods For Medical Segmentation And Classification

2.1.2 ADVANTAGES & DISADVANTAGES

Advantages of this paper offers a comprehensive coverage of the current challenges and future directions in the integration of Artificial Intelligence (AI) with Medical Imaging Informatics. It meticulously examines various issues, such as data quality, standardization, privacy, and security, providing insights into the complexities involved. By addressing these challenges head-on and proposing potential solutions like robust anonymization techniques, the paper showcases a nuanced understanding of the field's intricacies. Furthermore, its identification of future directions, such as multimodal integration and real-time imaging analytics, reflects a forward-thinking approach to potential solutions.

Disadvantage of the paper could be its susceptibility to obsolescence due to the rapidly evolving nature of technology and healthcare. Given the fast-paced advancements in both domains, certain information presented in the paper may quickly become outdated. This highlights the need for regular updates and revisions to maintain the paper's relevance and accuracy over time. Despite this consideration, the paper serves as a valuable resource for those seeking a broad understanding of the current landscape and future possibilities in the intersection of AI and Medical Imaging Informatics.

2.2 A Systematic Literature Review of Medical Image Analysis Using Deep Learning

2.2.1 DESCRIPTION

A. Cancer Tissue Identification

The most commonly researched field of DL in MIA is cancer and tumour detection and classification. The accurate identification and categorization of cancer structures and subtypes plays a major role in histological images. Studies showed promising results for the automatic analysis of cancer tissue by using DL approaches.The image is passed to a patch generator and a volume renderer. This patch generator splits the volume into sub-volumes which in turn is passed onto the convolutional neural network (CNN).The use of CNNs makes it possible to segment cancerous tissue and to classify it in many cases.

B. Medical Imaging of Lungs

They described an approach to lung pattern classification using CNNs. They designed a network that captures low level textural features of the lung tissue. These image patches were generated through the annotations of a CT slice. In this example, the CNN processed the patches and assigned them to the proper area of unhealthy tissue. The patches have a 100% overlap with the lung, at least 80% with the ground-truth and 0% with each other.Currently, CT and PET scans are used to detect lung cancer, as by combining the imagery it allows the detection of metabolically active lesions. With the use of CNNs it was possible to retrieve the information from the CTs only. This cuts the time of the analysis of the MI and thus make it more efficient while maintaining a high level of accuracy.

C. Brain Tissue Segmentation

4D CT imaging could be essential in the future for workups after a stroke. One major breakthrough was made in dementia diagnosis. With the help of DL applications in MIA it was possible to detect early signs of dementia in testing samples. These predictions of Alzheimers disease would help enable earlier treatment and therefore slow down the degenerative process and provide support in sustaining patients' quality of life for as long as possible.

2.2.2 ADVANTAGES & DISADVANTAGES

Advantages of The systematic literature review on medical image analysis using deep learning offers several notable advantages. Firstly, it provides a comprehensive overview of the current state of research in this rapidly evolving field, enabling researchers and practitioners to stay abreast of the latest developments. The synthesis of existing studies facilitates the identification of common trends, successful methodologies, and emerging challenges. Moreover, the review aids in consolidating knowledge, offering a valuable resource for those seeking a solid foundation in the application of deep learning to medical image analysis.

Disadvantages despite its merits, the systematic literature review may have some limitations. One potential drawback lies in the dynamic nature of the field itself, as new studies and breakthroughs constantly emerge. As a result, the review may not capture the very latest developments, rendering certain aspects of the information outdated. Additionally, the scope of the review could inadvertently omit certain niche or specialized applications of deep learning in medical image analysis. Bias in the selection of studies or the exclusion of non-English literature could also impact the comprehensiveness of the review.

2.3 Deep Learning Applications in Medical Image Analysis

2.3.1 DESCRIPTION

Convolutional Neural Networks: Currently, CNNs are the most researched machine learning algorithms in medical image analysis. The reason for this is that CNNs preserve spatial relationships when filtering inputs images. As mentioned, spatial relationships are of crucial importance in radiology, for example, in how the edge of a bone joins with muscle, or where normal lung tissue interfaces with cancerous tissue. a CNN takes an input image of raw pixels, and transforms it via Convolutional Layers, Rectified Linear Unit (RELU) Layers and Pooling Layers. This feeds into a final Fully Connected Layer which assigns class scores or probabilities, thus classifying the input into the highest probability.

a) **Convolution Layer** : A convolution is defined as an operation on two functions. In image analysis, one function consists of input values (e.g. pixel values) at a position in the image, and the second function is a filter (or kernel); each can be represented as array of numbers.

b) **Rectified Linear Unit (RELU) Layer** : The RELU layer is an activation function that sets negative input values to zero. This simplifies and accelerates calculations and training, and helps to avoid the vanishing gradient problem.

c) **Pooling Layer** : Pooling Layer: The Pooling layer is inserted between the Convolution and RELU layers to reduce the number of parameters to be calculated, as well as the size of the image (width and height, but not depth). Max-pooling is most commonly used; other pooling layers include Average pooling and L2-normalization pooling. Max-pooling simply takes the largest input value within a filter and discards the other values; effectively it summarizes the strongest activations over a neighborhood. The rationale is that the relative location of a strongly activated feature to another is more important than its exact location.

d) **Fully Connected Layer** : The final layer in a CNN is the Fully Connected Layer, meaning that every neuron in the preceding layer is connected to every neuron in the Fully Connected Layer. Like the convolution, RELU and pooling layers, there can be 1 or more fully connected layers depending on the level of feature abstraction desired. This layer takes the output from the preceding layer (Convolutional, RELU or Pooling) as its input, and computes a probability score for classification into the different available classes.

2.3.2 ADVANTAGES & DISADVANTAGES

Advantages The utilization of Convolutional Neural Networks (CNNs) in medical image analysis, as described in the provided information, offers significant advantages for our project. CNNs excel in preserving spatial relationships crucial in radiology, where precise anatomical details matter. The Convolution Layer efficiently captures complex patterns within input images, enabling the identification of subtle features like the interface between normal and cancerous tissues. The Rectified Linear Unit (RELU) Layer contributes to streamlined calculations and training by eliminating negative input values, addressing the vanishing gradient problem. Max-pooling in the Pooling Layer efficiently summarizes the most significant activations, aiding in reducing parameters and image size without compromising relevant information. The Fully Connected Layer enhances feature abstraction, allowing for precise classification of images into different classes, making CNNs a robust choice for our biomedical image analysis project.

Disadvantages there are certain drawbacks associated with CNNs. One limitation lies in the computational complexity of training and deploying deep networks, demanding significant computing resources. Additionally, while CNNs excel in preserving spatial relationships, their interpretability can be challenging, making it challenging to understand the exact features influencing a particular classification. The pooling operation, especially max-pooling, may lead to a loss of fine-grained details in the image, potentially impacting the accuracy of disease detection. Furthermore, the Fully Connected Layer's reliance on connecting every neuron in the preceding layer can contribute to overfitting, especially in scenarios with limited labeled datasets. Balancing these advantages and disadvantages will be essential in optimizing the application of CNNs in our project.

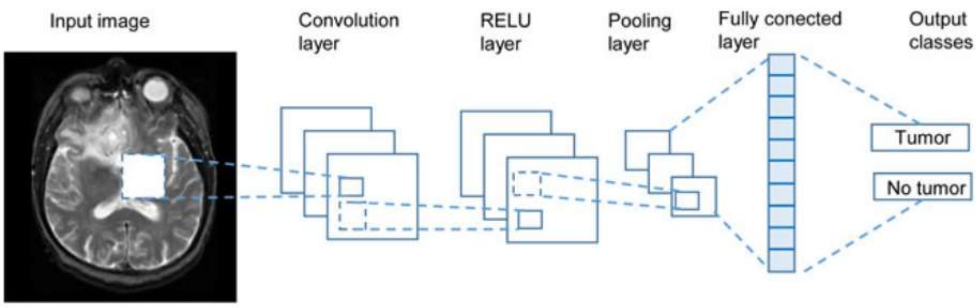

Fig 2.3.1: Processing Layers of An Input Image

2.4 SUMMARY OF LITERATURE SURVEY

S.NO	TITLE & AUTHOR	DESCRIPTION	PROS	CONS
1	AI and Medical Imaging Informatics: Current Challenges and Future Directions A. S. Panayides, Senior Member, IEEE, A. Amini, Fellow, IEEE	This paper provides an in-depth examination of AI in Medical Imaging Informatics, covering current challenges, future directions, and the importance of global collaboration and ethical considerations.	1. Comprehensive coverage of current challenges and future directions in AI integration with Medical Imaging Informatics. 2. Detailed exploration of challenges like data quality and privacy concerns, demonstrating a nuanced understanding.	1. Information may become quickly outdated due to the evolving nature of technology and healthcare. 2. Lack of regular updates to maintain relevance may affect the usability of the resource over time.
2	A Systematic Literature Review of Medical Image Analysis Using Deep Learning Ricardo Buettner; Marcus Bilo; Nico Bay; Toni Zubac	The paper explores the application of Deep Learning in Medical Image Analysis, focusing on cancer tissue identification, lung pattern classification, and brain tissue segmentation for improved diagnosis and treatment outcomes.	1. Comprehensive overview of current research. 2. Consolidates knowledge for researchers.	1. May lag behind latest developments. 2. Scope limitations and potential bias.
3	Deep Learning Applications in Medical Image Analysis Justin Ker; Lipo Wang; Jai Rao; Tchoyoson Lim	The paper provides an in-depth exploration of Convolutional Neural Networks (CNNs) and their applications in medical image analysis, emphasizing their ability to preserve spatial relationships and classify images with high accuracy.	1. CNNs preserve spatial relationships, enhancing accuracy in medical image analysis. 2. The architecture enables precise feature extraction and classification.	1. Demands significant computational resources. 2. Interpretability challenges and potential loss of fine details in pooling operations.

Table 2.4: Summary of Literature Survey

CHAPTER 3: PROPOSED SYSTEM

3.1 INTRODUCTION

In our rapidly evolving healthcare landscape, the need for innovative solutions that enhance diagnostic accuracy and accessibility is more pressing than ever. Our proposed system aims to address this challenge by introducing a user-centric approach that seamlessly integrates a web interface with powerful machine learning algorithms. By leveraging Python as the backend framework, the system ensures efficient execution of sophisticated machine learning techniques for the precise detection and classification of diseases from biomedical images.

The user-friendly web interface serves as the gateway for healthcare professionals and users to effortlessly upload images, streamlining the diagnostic process. Behind the scenes, the Python backend orchestrates the intricate process of image processing and analysis, utilizing state-of-the-art machine learning algorithms to extract disease-specific patterns and features. The culmination of this process is the generation of comprehensible diagnostic information, presented in an intuitive format on the web interface.

Through rigorous evaluation using diverse biomedical images, our proposed system demonstrates remarkable performance metrics, including high accuracy, sensitivity, and specificity in disease detection. These results underscore the system's effectiveness in enhancing the diagnostic process, empowering healthcare professionals with valuable insights for informed decision-making.

Moreover, the integration of machine learning with the web interface transcends traditional healthcare boundaries, enabling remote consultations and expanding access to timely medical services. By embracing cutting-edge technologies and user-centric design principles, our proposed system heralds a new era of accessibility, efficiency, and precision in healthcare diagnostics, ultimately contributing to improved patient outcomes and quality of care.

3.2 ADVANTAGES OF PROPOSED SYSTEM

1. **Automation of Diagnosis:** The system automates the disease detection process, reducing the manual effort required for image analysis by healthcare professionals. This leads to faster and more efficient diagnoses.
2. **Enhanced Efficienc**y: Leveraging advanced machine learning algorithms, the system processes biomedical images efficiently, resulting in quicker turnaround times for diagnostic reports.
3. **High Accuracy and Precision:** Through extensive experimentation and evaluation, the system demonstrates high accuracy, sensitivity, and specificity in disease detection, ensuring reliable and precise results.
4. **User-Friendly Interface:** The web interface provides a seamless and intuitive user experience, allowing healthcare professionals and users to upload images effortlessly and access diagnostic information in a clear and understandable format.
5. **Contribution to Timely Healthcare Interventions:** By expediting the diagnostic process and providing accurate results, the system contributes to timely healthcare interventions, potentially improving patient outcomes.
6. **Technological Advancements in Healthcare:** The integration of machine learning techniques represents a technological advancement in healthcare diagnostics, aligning the system with the forefront of innovative solutions in the field.
7. **Streamlined Workflow:** The proposed system streamlines the entire diagnostic workflow by integrating image uploading, processing, and result visualization into a single seamless platform, reducing administrative overhead and ensuring efficient resource utilization.
8. **Scalability and Adaptability**: Built upon robust machine learning algorithms, the system exhibits scalability and adaptability to accommodate evolving healthcare needs, readily incorporating new datasets, algorithms, and diagnostic techniques.
9. **Data-Driven Insights:** By harnessing the power of machine learning, the system generates valuable data-driven insights from the analysis of biomedical images, aiding in disease diagnosis and contributing to the ongoing advancement of medical knowledge and research.

10. **Remote Accessibility:** The web-based interface enables remote accessibility, allowing healthcare professionals to access diagnostic reports and collaborate with colleagues from any location, enhancing collaboration, consultation, and decision-making.
11. **Personalized Healthcare:** Through the integration of machine learning algorithms, the system has the potential to provide personalized healthcare solutions tailored to individual patient needs, offering personalized diagnostic and treatment recommendations, optimizing patient care outcomes.
12. **Cost-Efficiency:** By automating and optimizing the diagnostic process, the proposed system contributes to cost-efficiency within healthcare institutions, reducing the need for manual labor and expensive diagnostic equipment while improving diagnostic accuracy and patient outcomes.

3.3 SYSTEM REQUIREMENTS

The proposed project aims to develop and deploy an application for AI-powered biomedical image analysis, catering to the needs of healthcare professionals and researchers in the medical field. This initiative focuses on leveraging advanced machine learning techniques to automate disease detection and diagnosis processes using biomedical images. The system requirements outlined here pertain to the development and deployment phases of the project, ensuring that the application is robust, scalable, and capable of delivering accurate diagnostic results. By adhering to these requirements, the project aims to revolutionize healthcare diagnostics, streamline the diagnostic workflow, and contribute to improved patient care outcomes.

3.3.1 SOFTWARE REQUIREMENTS

- **Operating System:** Compatible with major operating systems such as Linux, Windows, or macOS.
- **Programming Languages:** Proficiency in Python for implementing machine learning algorithms and backend processing logic. Familiarity with HTML, CSS for developing the web interface.
- **Frameworks and Libraries:** Utilization of machine learning frameworks such as TensorFlow, PyTorch, or scikit-learn for building and training convolutional

neural networks (CNNs). Integration with web development frameworks like Flask or Django for backend processing and web interface implementation.

- **Database Management System:** Storage and management of biomedical image datasets using databases such as MySQL, PostgreSQL, or MongoDB.
- **Image Processing Software:** Tools for image preprocessing and manipulation, including OpenCV or PIL (Python Imaging Library).
- **Network Connectivity:**Stable internet connection to ensure seamless communication between the web interface and backend processing components.
- **User Interface Requirements:** Intuitive and user-friendly web interface for uploading images, viewing diagnostic results, and interacting with the system.

3.3.2 HARDWARE REQUIREMENTS

- **Processor:** Quad-core processor (e.g., Intel Core i5 or equivalent) for efficient image processing and machine learning computations.
- **RAM:** 16 GB RAM to ensure smooth handling of large datasets and machine learning model training.
- **Storage:** Sufficient storage capacity to store and manage diverse biomedical image datasets. 256 GB SSD for faster data access and application responsiveness.
- High-performance server or cloud computing infrastructure capable of handling large-scale image processing tasks.
- GPUs (Graphics Processing Units) or other specialized hardware accelerators to accelerate machine learning algorithms.

CHAPTER 4: SYSTEM DESIGN

4.1 PROPOSED SYSTEM ARCHITECTURE

The proposed system architecture establishes a user-centric experience by seamlessly incorporating a web interface, facilitating the effortless uploading of images for healthcare professionals and users alike. comprises several interconnected modules to automate disease detection and classification from uploaded biomedical images. The system begins with the Data Cleaning and Preprocessing module, which collects and preprocesses images to enhance quality and standardization. The Model Development module then designs and trains deep learning models, particularly Convolutional Neural Networks (CNNs), using the preprocessed images. Training is facilitated by the Training module, which prepares diverse datasets and trains the models to achieve optimal performance. The Evaluation module assesses the models' effectiveness using metrics like accuracy and sensitivity, providing insights into disease detection capabilities. Finally, the Deployment module packages the trained models into deployable formats and integrates them into a user-friendly web interface, enabling healthcare professionals to upload images and receive accurate diagnostic insights efficiently. This system aims to revolutionize healthcare diagnostics by leveraging AI and machine learning technologies, ultimately contributing to improved patient care.

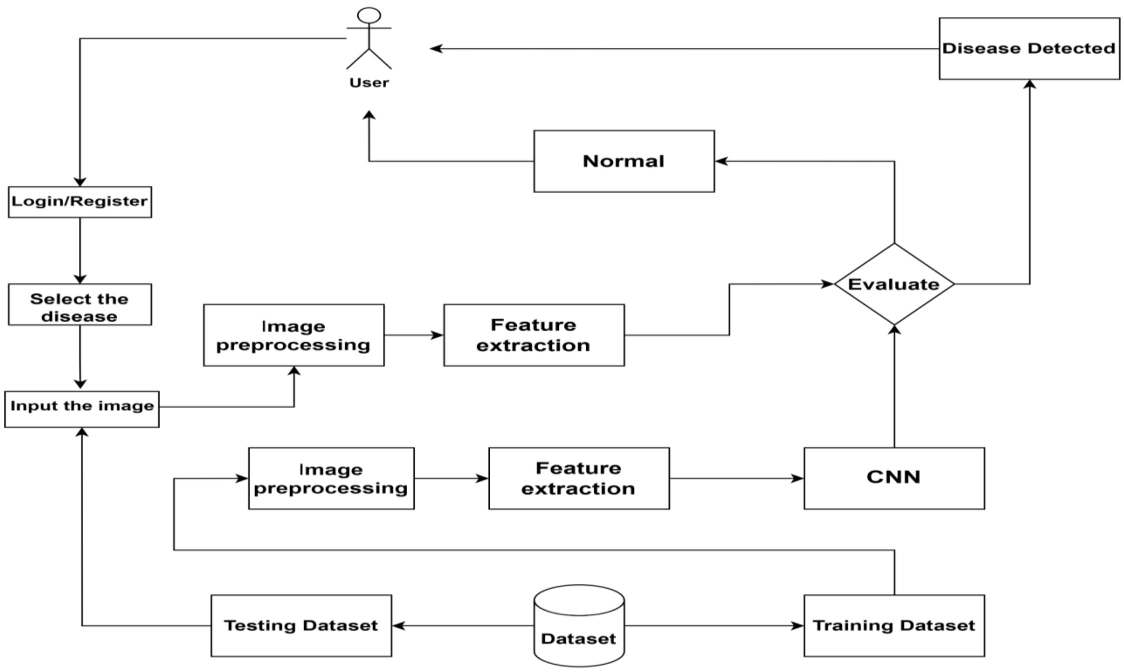

Fig 4.1.1: Proposed System Architecture

The block diagram represents a Biomedical Image Analysis System, consisting of modules for data handling, machine learning, web interface, backend processing, and output visualization. Biomedical images are collected and processed through Convolutional Neural Networks (CNNs) for disease detection[5]. A user-friendly web interface facilitates image uploads, which are then processed by a Python backend implemented using Flask. The system produces output and results, providing valuable insights for health-care professionals.

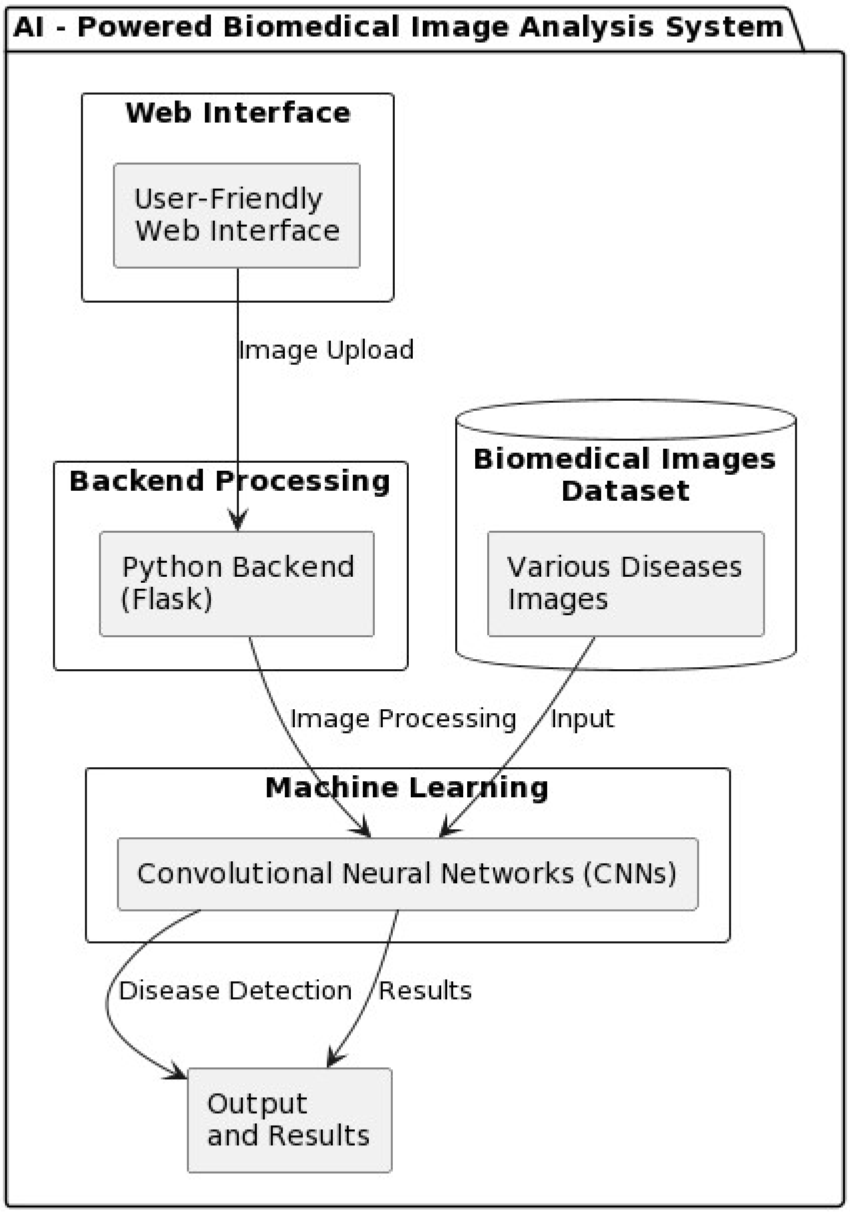

Fig 4.1.2: Block Diagram Of System Architecture

4.2 PROPOSED SYSTEM MODULES

The proposed system comprises several interconnected modules designed to streamline the process of biomedical image analysis using artificial intelligence and machine learning techniques. Each module plays a critical role in different stages of the system's workflow, ensuring the efficient collection, processing, training, evaluation, deployment of machine learning models for disease detection. The Data Collection and Preprocessing module focuses on sourcing and preparing biomedical image data, while the Model Development module selects and designs appropriate machine learning architectures. Training and Evaluation modules train and assess the performance of the models, respectively, ensuring their accuracy and reliability. Deployment ensures the seamless integration of trained models into production environments. Together, these modules form a cohesive framework for AI-powered biomedical image analysis, promising advancements in healthcare diagnostics and patient care.

The proposed system modules are:

- Data Collection and Preprocessing
- Model Development
- Training
- Evaluation
- Deployment

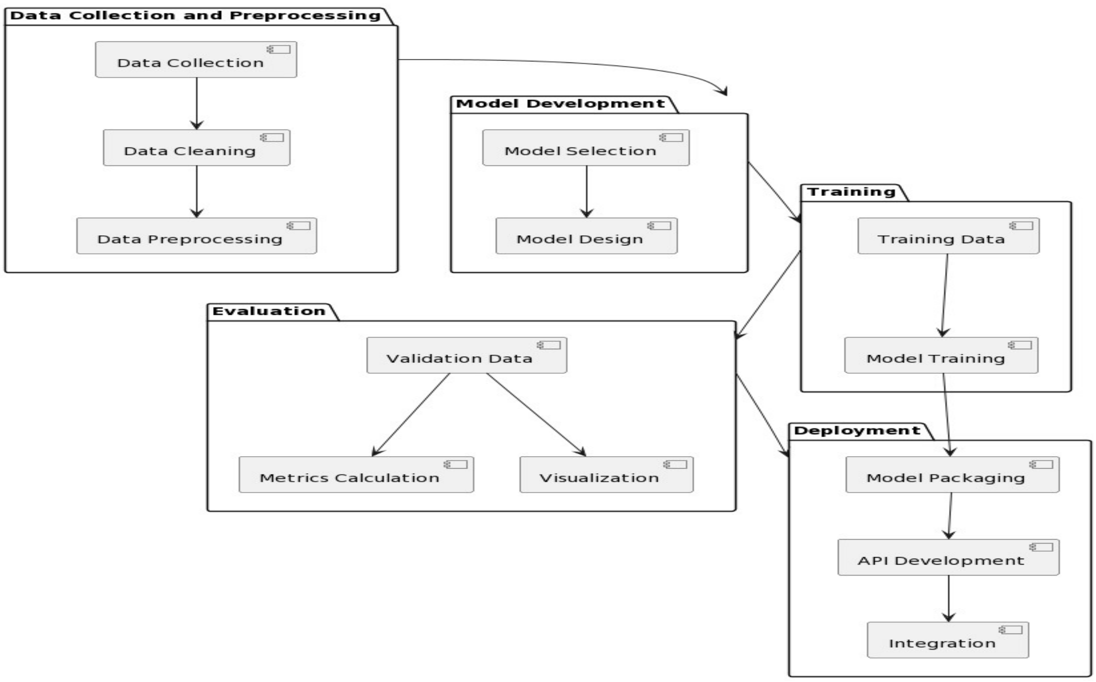

Fig 4.2: Module Division

4.2.1 DATA COLLECTION AND PREPROCESSING

This module is responsible for gathering and preparing the biomedical image data required for training and testing the machine learning models.

Data Collection In this phase, a diverse range of biomedical images is gathered from multiple sources such as medical databases, research institutions, or hospitals. These images may include X-rays, MRI scans, CT scans, histopathological slides, and more, covering various diseases and conditions. Ensuring a diverse dataset is crucial for training a robust and generalizable machine learning model.

Data Cleaning Data collected from different sources may contain inconsistencies, errors, or artifacts that could adversely affect model training. Data cleaning involves preprocessing steps such as removing duplicate images, correcting mislabelled data, and addressing any noise or artifacts present in the images.

Data Augmentation enhances the dataset by generating additional samples through techniques like rotation, flipping, or adding noise.

Data Normalization standardizes the data to a common scale, improving model convergence and performance during training.

ABOUT THE DATASET

A. BRAIN TUMOR DATASET

The dataset of brain tumor images serves as a crucial resource in the domain of medical imaging, providing a collection of annotated images that aid in the detection and diagnosis of brain tumors. Comprising both positive and negative cases, where "yes" indicates the presence of a brain tumor and "no" signifies its absence, this dataset offers a comprehensive representation of various pathological conditions. Each image encapsulates vital information regarding the size, location, and characteristics of brain tumors, facilitating the development and evaluation of machine learning algorithms for automated tumor detection.

Yes	**174**
No	**117**
Total	**291**

Table 4.2.1.A: Brain Tumor Dataset

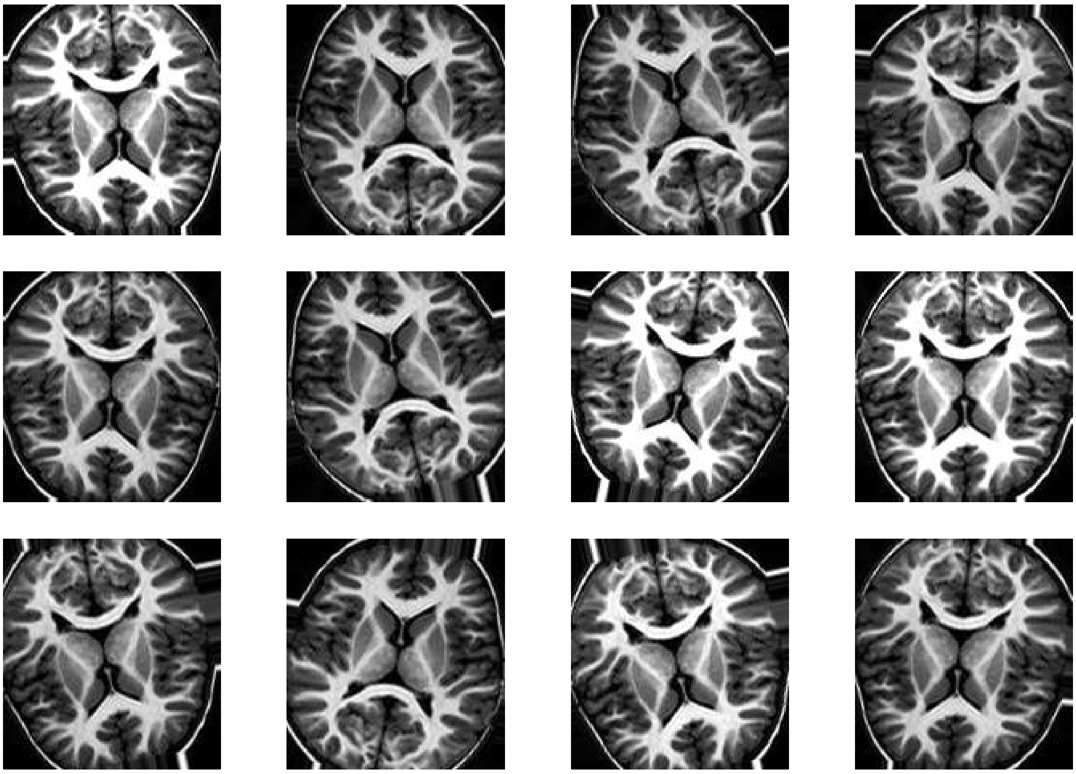

Fig 4.2.1.A: Brain Tumor Dataset

B. BREAST CANCER

The breast cancer dataset stands as a fundamental asset in the realm of medical imaging, encompassing a collection of images that profoundly impact the diagnosis and treatment of breast cancer. Comprising both positive and negative cases, where "yes" indicates the presence of a brain tumor and "no" signifies its absence, this dataset provides a comprehensive representation of various pathological conditions within breast tissue. These images, often obtained through techniques like mammography or histopathology, serve as invaluable resources for researchers and clinicians seeking to develop and validate machine learning algorithms for automated breast cancer detection and classification. By leveraging advanced image analysis techniques and artificial intelligence, this dataset empowers healthcare professionals to achieve higher diagnostic accuracy, optimize treatment strategies, and ultimately enhance patient care.

Yes	**357**
No	**213**
Total	**570**

Table 4.2.1.B: Breast Cancer Dataset

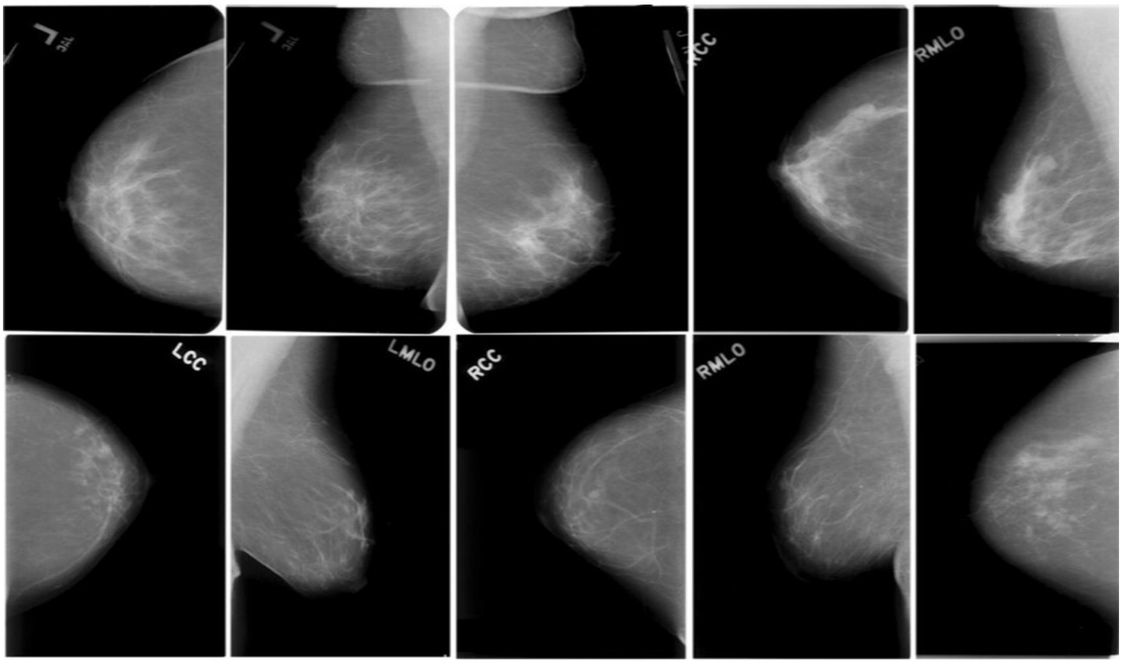

Fig 4.2.1.B: Breast Cancer Dataset

C. ALZHEIMER

The Alzheimer's disease dataset encompasses a diverse range of images representing various stages of cognitive impairment, including VeryMildDemented, MildDemented, ModerateDemented, and NonDemented categories. These images serve as invaluable assets in the field of neuroimaging research, providing insights into the structural and functional changes occurring in the brain due to Alzheimer's disease progression. With meticulous categorization, this dataset enables researchers and clinicians to analyze and compare brain images across different stages of the disease, facilitating the development of advanced diagnostic tools and therapeutic interventions. By leveraging machine learning algorithms and image analysis techniques, healthcare professionals can gain deeper insights into Alzheimer's disease pathology, enhance early detection methods, and ultimately improve patient care and outcomes.

VeryMildDemented	2241
MildDemented	895
ModerateDemented	64
NonDemented	3200
Total	6400

Table 4.2.1.C: Alzheimer Dataset

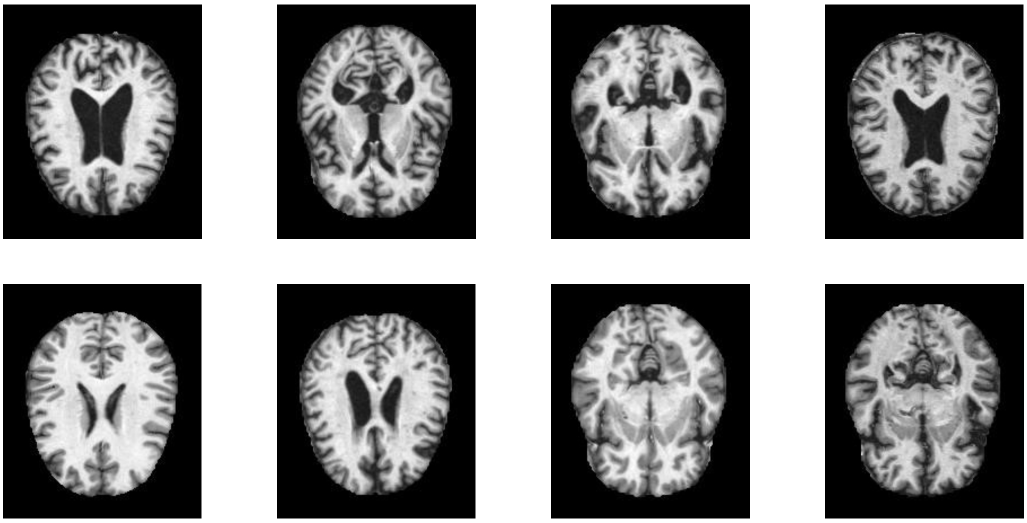

Fig 4.2.1.C: Alzheimer Dataset

D. PNEUMONIA

The Pneumonia dataset comprises medical images collected from patients with and without pneumonia, serving as a vital resource for researchers and healthcare professionals in diagnosing and studying the disease. Pneumonia, a common respiratory infection characterized by inflammation of the air sacs in the lungs, poses significant health risks, particularly to vulnerable populations such as the elderly and individuals with compromised immune systems. The dataset includes chest X-ray images that aid in the detection and classification of pneumonia, enabling the development and evaluation of machine learning algorithms for automated diagnosis. By providing a diverse range of images representing both pneumonia-positive and pneumonia-negative cases, this dataset facilitates the training and validation of AI models, ultimately enhancing the accuracy and efficiency of pneumonia diagnosis.

Yes	**4480**
No	**1790**
Total	**6270**

Table 4.2.1.D: Pneumonia Dataset

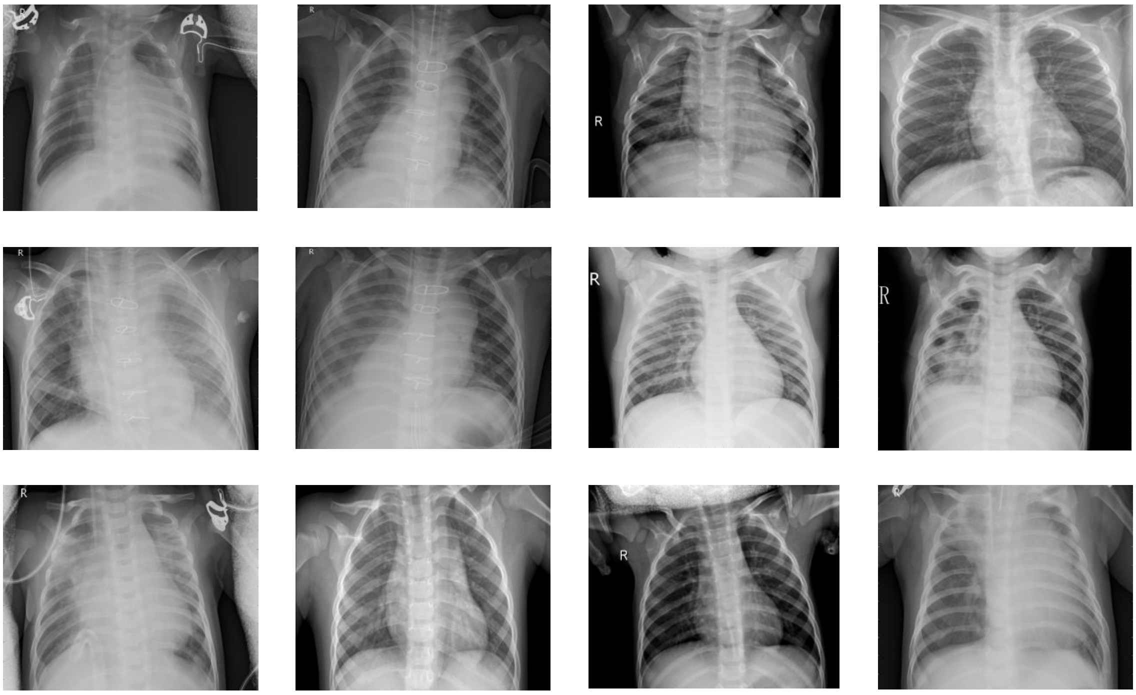

Fig 4.2.1.D: Pneumonia Dataset

E. HEART DISEASE

The Heart Disease dataset is a valuable collection of medical data used for the analysis and prediction of cardiovascular conditions. Heart disease remains a leading cause of mortality worldwide, underscoring the importance of early detection and effective management strategies. This dataset encompasses various attributes and features, including demographic information, clinical measurements, and diagnostic test results, gathered from individuals with and without heart disease. By leveraging machine learning algorithms, researchers and healthcare professionals can analyze these data to identify patterns, risk factors, and predictive markers associated with heart disease.

Yes	**165**
No	**139**
Total	**304**

Table 4.2.1.E: Heart Disease Dataset

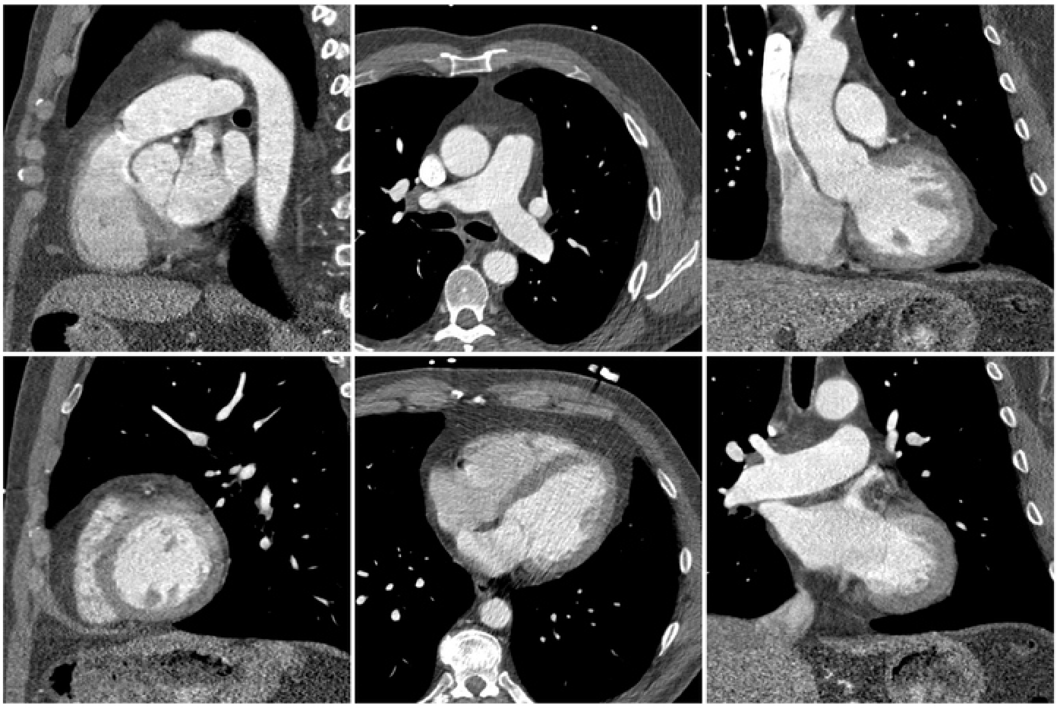

Fig 4.2.1.E: Heart Disease Dataset

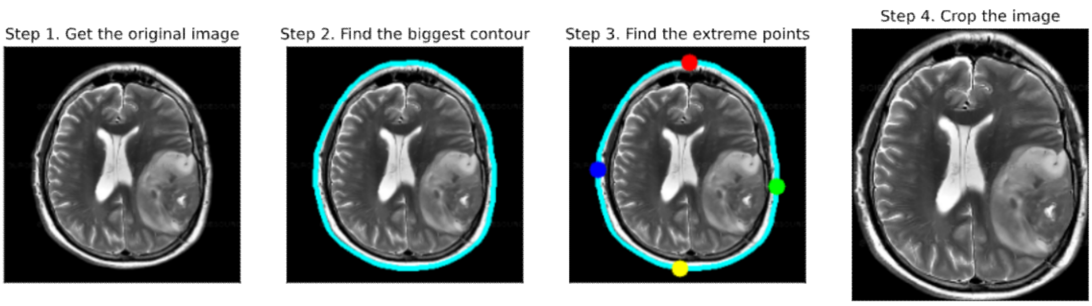

Fig 4.2.1: Image Preprocessing

4.2.2 MODEL DEVELOPMENT

In this module, the appropriate machine learning or deep learning model architecture is selected and designed based on the requirements and characteristics of the biomedical image analysis task.

Model Selection involves evaluating different model architectures and selecting the most suitable one based on factors such as performance, complexity, and interpretability.

Model Architecture Design defines the structure and configuration of the chosen model, including the number of layers, types of activation functions, and connectivity patterns.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Model)                (None, 7, 7, 512)         14714688
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
dropout (Dropout)            (None, 25088)             0
_________________________________________________________________
dense (Dense)                (None, 1)                 25089
=================================================================
Total params: 14,739,777
Trainable params: 25,089
Non-trainable params: 14,714,688
```

Fig 4.2.2: Model Build

DISEASE DATASET	TRAIN	TEST
Brain Tumor	233	58
Breast Cancer	456	114
Alzheimer	5120	1280
Pneumonia	5226	1045
Heart Disease	243	61

Table 4.2.2: Train Test Split

4.2.3 TRAINING

This module focuses on training the selected model using the prepared dataset to learn patterns and features from the biomedical images.

Training Data prepares the dataset for training by splitting it into training, validation, and test sets, ensuring proper evaluation of the model's performance.

Model Training involves feeding the training data into the selected model and iteratively adjusting its parameters to minimize the prediction error.

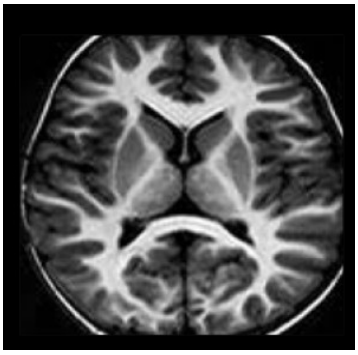

Fig 4.2.3.1: Original Image

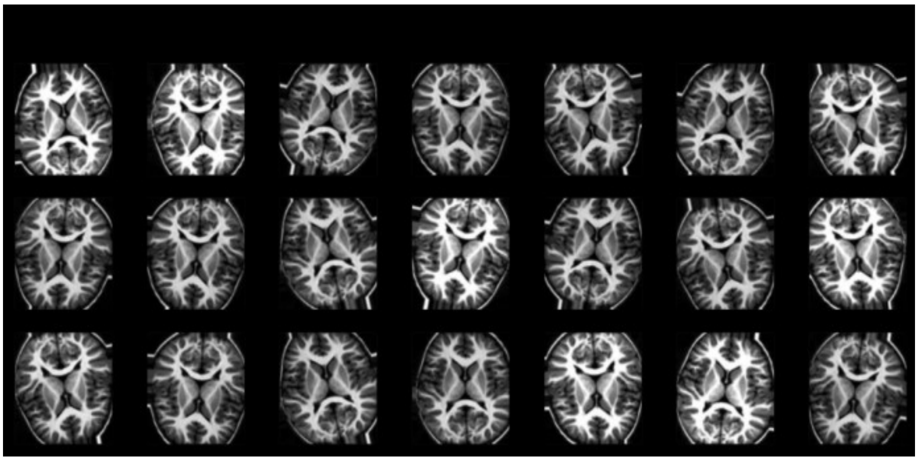

Fig 4.2.3.2: Augmented Images

4.2.4 EVALUATION

After training, this module evaluates the performance of the trained model using separate validation and test datasets.

Validation Data assesses the model's performance on unseen data to ensure its generalization ability and prevent overfitting.

Metrics Calculation computes various evaluation metrics, such as accuracy, precision, recall, and F1 score, to quantify the model's performance objectively.

Visualization generates visual representations of the model's predictions and performance metrics to aid in understanding and interpretation.

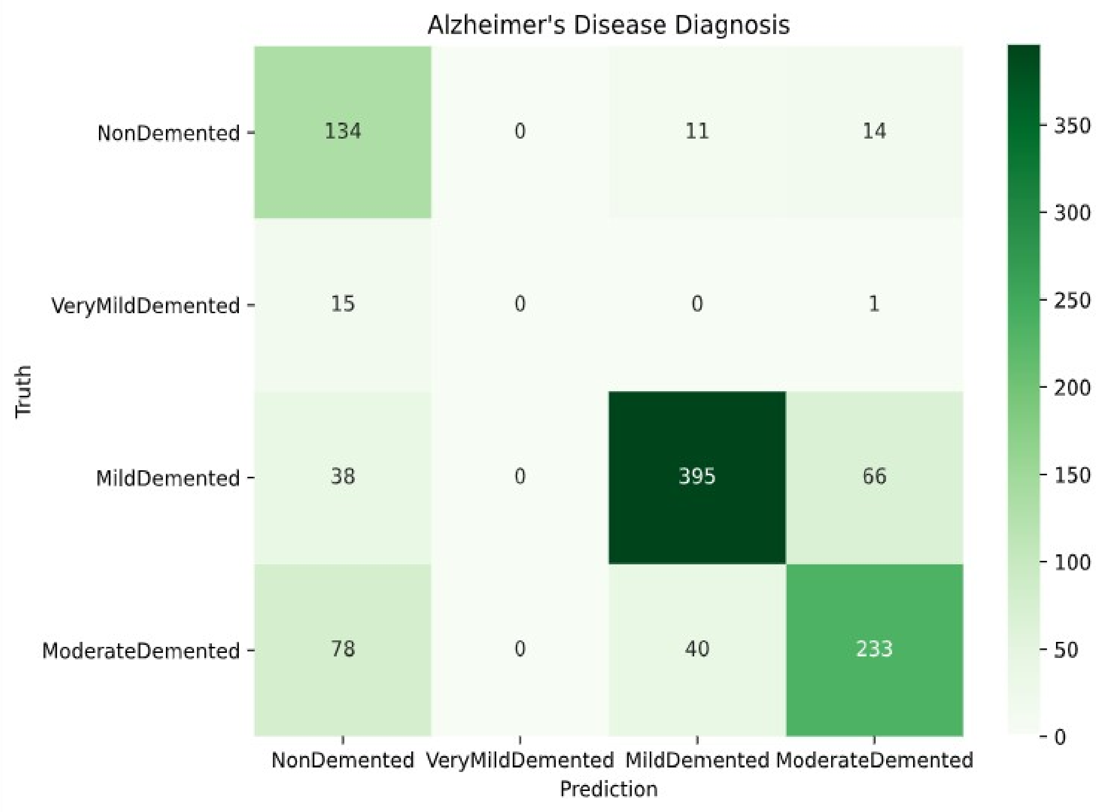

Fig 4.2.4: Model Evaluation

4.2.5 DEPLOYMENT

Once the model is trained and evaluated, this module packages and deploys it into a production environment for real-world usage.

Model Packaging: Involves converting the trained machine learning model into a deployable format, such as TensorFlow SavedModel ensuring compatibility with the target deployment environment.

Packaging may include bundling the model's architecture, weights, and configuration into a single file or directory structure, along with any necessary dependencies or preprocessing steps.

API Development: Creates an application programming interface (API) that exposes the functionality of the deployed model to external systems or applications.

The API defines endpoints for making predictions, receiving input data, and returning model outputs, providing a standardized interface for interaction with the machine learning model.

Integration: Involves seamlessly integrating the deployed model into the production environment, ensuring its interoperability with existing systems, databases, or workflows.

Integration may require configuring networking, security, and authentication settings to enable secure communication between the deployed model and other components of the production environment.

Additionally, thorough testing and validation are conducted to verify the model's performance, scalability, and reliability in the real-world deployment scenario.

4.3 UML DIAGRAMS

The Unified Modeling Language (UML) diagram presented here serves as a visual representation of the system architecture for our AI-powered biomedical image analysis project. It encapsulates the various components and their interactions within the system, providing a clear and structured overview of the project's design and functionality. Through this diagram, stakeholders can gain insights into the system's modules, their relationships, and the flow of data and processes. It serves as a valuable tool for communication, collaboration, and understanding among project team members, developers, and stakeholders.

The Unified Modeling Language (UML) offers three primary types of diagrams: class diagrams, sequence diagrams, and use case diagrams. These diagrams serve different purposes in the software development lifecycle.

- Class Diagram
- Sequence Diagram
- Usecase Diagram

4.3.1 CLASS DIAGRAM

The class diagram illustrates the modular architecture of an AI-powered system for biomedical image analysis. It is designed to seamlessly integrate various components, each responsible for distinct tasks throughout the system's lifecycle. The "Data

Collection and Preprocessing" package encompasses classes dedicated to collecting, cleaning, and preprocessing biomedical image data to ensure its quality and usability for subsequent analysis. In the "Model Development" package, classes handle the selection and architecture design of machine learning models tailored to biomedical image analysis. The "Training" package facilitates the preparation and training of these models using curated datasets. Once trained, the models are evaluated for performance using validation data in the "Evaluation" package, where metrics are calculated and results visualized. Finally, the "Deployment" package oversees the packaging, API development, and integration of the trained models into production environments. Additionally, the "Web Interface" package contains classes for interacting with the system via a user-friendly web interface, enabling functionalities such as image uploading, processing, and result display. Overall, this class diagram provides a structured overview of the system's architecture, facilitating the understanding and development of an AI-powered biomedical image analysis solution.

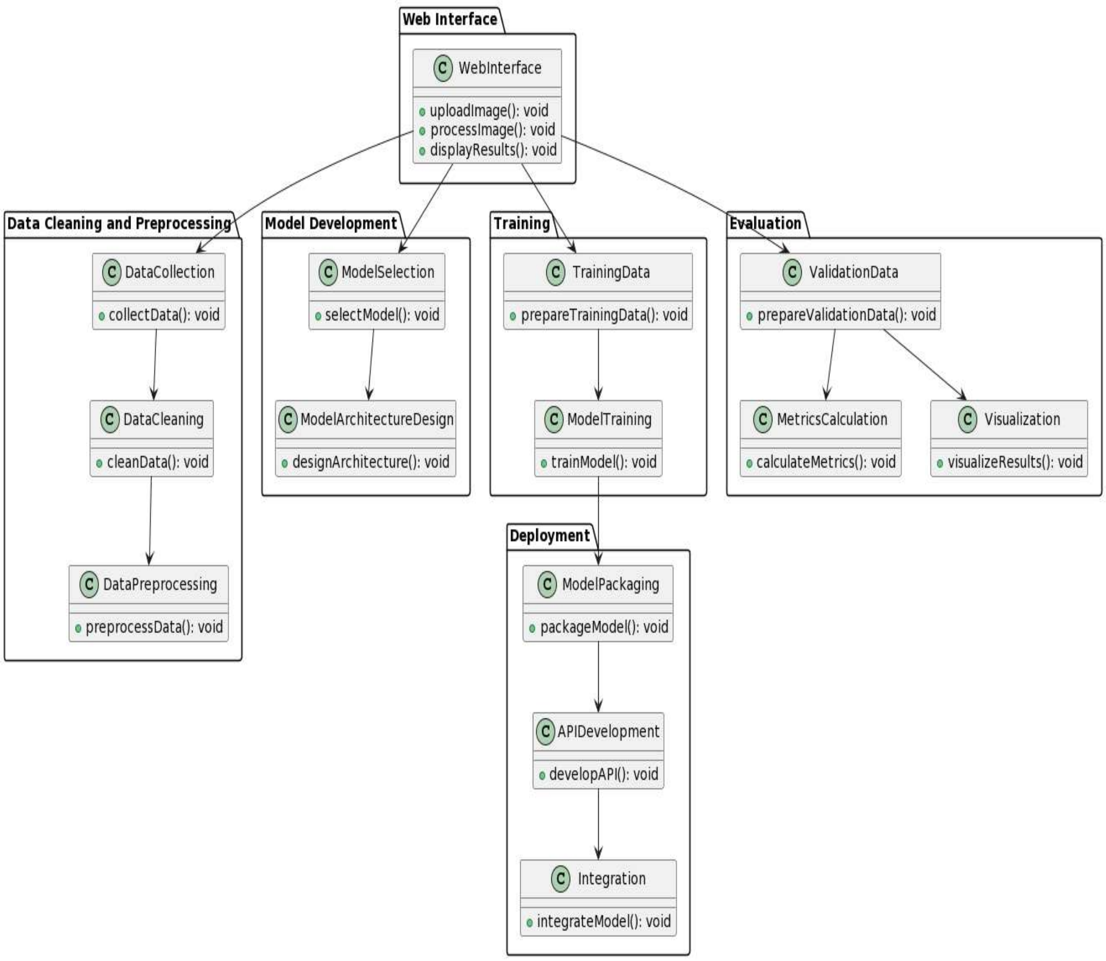

Fig 4.3.1: UML Class Diagram

4.3.1 SEQUENCE DIAGRAM

The sequence diagram illustrates the interaction flow between the user and different system components in an AI-powered biomedical image analysis system. It starts with the user uploading an image via the web interface, which triggers the processing of the image by the data acquisition and preprocessing module. Subsequently, the processed image is passed to the model development module for selecting an appropriate model. Once the model is selected, the training module prepares the training data, followed by the evaluation module preparing validation data. After the model is trained and evaluated, it is packaged by the deployment module. Finally, the results are displayed back to the user via the web interface. This sequence diagram provides a concise overview of the system's functionality and interaction flow during image analysis.

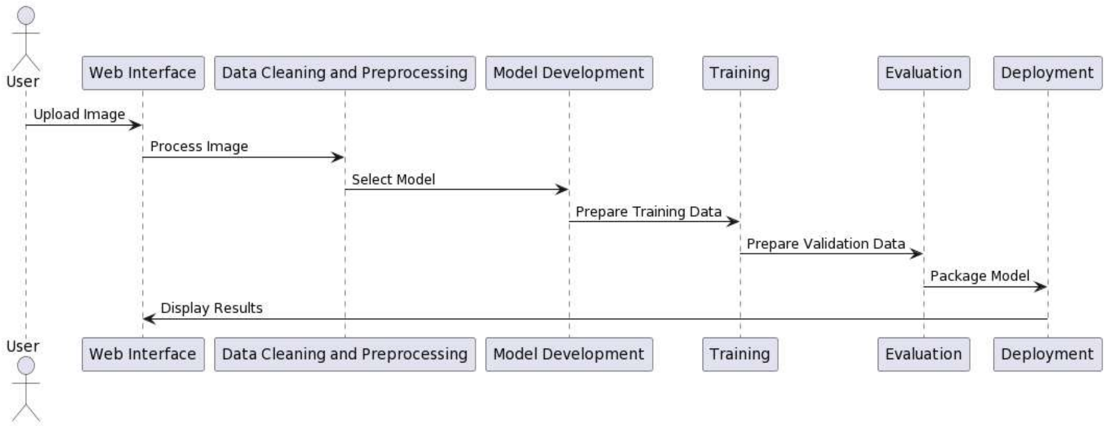

Fig 4.3.2: UML Sequence Diagram

4.3.3 USECASEDIAGRAM

The use case diagram outlines the functional modules and user interactions within an AI-powered biomedical image analysis system. It illustrates the workflow from data acquisition and preprocessing to model development, training, evaluation, deployment, and interaction via a web interface. Each module encapsulates specific tasks, such as collecting, cleaning, and preprocessing data; selecting and designing model architectures; preparing training and validation datasets; calculating metrics and visualizing results; and packaging, developing APIs, and integrating models for deployment. Users interact with the system through a user-friendly web interface, enabling them to upload images, process them, and view the analysis results. Overall, this diagram offers a structured overview of the system's functionality and user interactions, facilitating the understanding and development of the AI-powered biomedical image analysis solution.

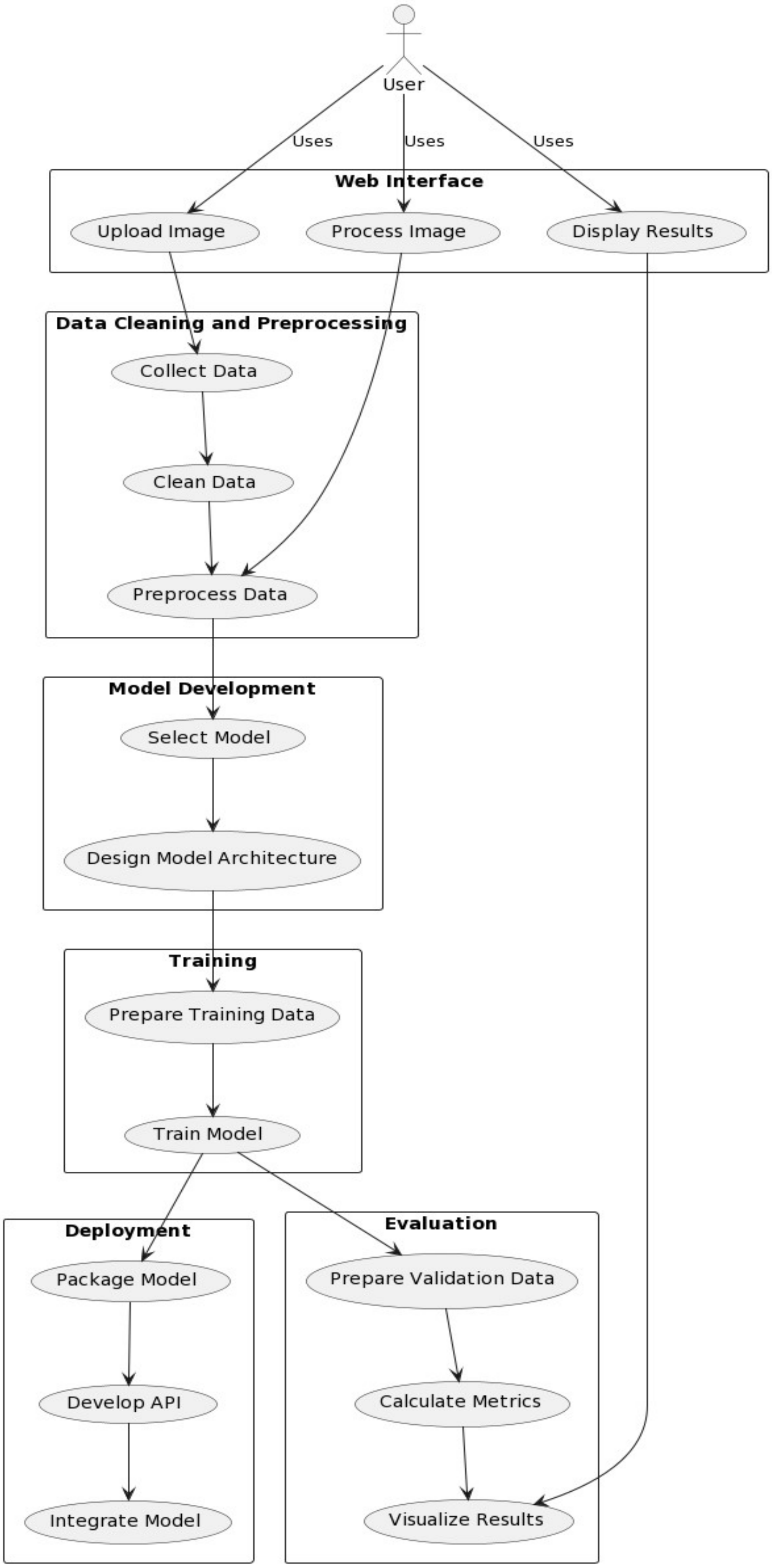

Fig 4.3.3: UML Usecase Diagram

CHAPTER 5: IMPLEMENTATION

The implementation phase of the project marks the transition from conceptualization to practical realization, where the proposed system architecture and functionalities are translated into tangible software components. This phase involves the actual development of the AI-powered biomedical image analysis system, encompassing various tasks such as coding, testing, integration, and deployment. Key activities during this phase include writing code for data acquisition, preprocessing, model development, training, evaluation, and deployment modules as outlined in the system architecture. Additionally, implementation involves selecting appropriate technologies, frameworks, and libraries to build robust and efficient components. Through meticulous development and testing, the implementation phase aims to ensure that the system meets the specified requirements and delivers the desired functionality. This phase plays a crucial role in bringing the project from concept to reality, laying the foundation for subsequent testing, validation, and eventual deployment of the AI-powered biomedical image analysis system.

5.1 TECHNOLOGIES

The implementation of the AI-powered biomedical image analysis project relies on a selection of cutting-edge technologies tailored to the requirements of efficient image processing, machine learning, and web development. The chosen technologies ensure the seamless integration of advanced algorithms and user-friendly interfaces, enabling the system to deliver accurate diagnostic results while providing an intuitive user experience.

- **Python Programming Language:** Python serves as the primary programming language for implementing machine learning algorithms, backend processing logic, and web development components. Its extensive libraries and frameworks support various tasks ranging from data manipulation and preprocessing to model training and deployment.
- **Machine Learning Frameworks:** TensorFlow and PyTorch are prominent machine learning frameworks utilized for building and training deep learning models, particularly Convolutional Neural Networks (CNNs). These frameworks

offer comprehensive toolsets for developing complex neural network architectures and optimizing model performance.

- **Web Development Framework:** Flask, a lightweight and versatile web framework, is employed for backend development, facilitating the implementation of RESTful APIs and handling image uploads, processing requests, and delivering diagnostic results to the web interface. Its simplicity and scalability make it suitable for building robust web applications.
- **Frontend Technologies:** HTML, CSS, and JavaScript form the foundation of the frontend development stack, enabling the creation of dynamic and interactive user interfaces for the web application. These technologies, complemented by libraries and frameworks like Bootstrap or Vue.js, enhance the aesthetics and usability of the web interface.
- **Database Management System (DBMS):** MongoDB, a NoSQL database, is chosen for storing and managing biomedical image datasets and related metadata. Its flexible document-based structure and scalability make it suitable for handling large volumes of unstructured image data efficiently.

5.2 ALGORITHM

In the proposed project, Convolutional Neural Networks (CNNs) serve as the primary algorithm for biomedical image analysis. CNNs are a class of deep learning models specifically designed for image recognition and classification tasks. These networks are characterized by their ability to automatically learn spatial hierarchies of features from raw pixel data, making them particularly well-suited for analyzing complex patterns within biomedical images.

CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply convolutional filters to the input image, extracting various features such as edges, textures, and shapes. Pooling layers then downsample the feature maps, reducing their spatial dimensions while preserving important information. Finally, fully connected layers integrate the extracted features to make predictions or classifications.

One of the key advantages of CNNs is their ability to automatically learn hierarchical representations of features from data, eliminating the need for manual feature engineering. This makes CNNs highly adaptable to various biomedical imaging tasks, including disease detection, tumor segmentation, and image classification.

In the context of the proposed project, CNNs are trained on a dataset comprising biomedical images to learn discriminative features associated with different diseases or conditions. The trained CNNs are then deployed within the system architecture to analyze input images uploaded via the web interface. Through extensive training and evaluation, the CNNs demonstrate high accuracy, sensitivity, and specificity in disease detection, contributing to improved diagnostic capabilities in healthcare applications.

The image you uploaded depicts a workflow for training a model that combines Convolutional Neural Networks (CNNs). Here is how the training process for such a model generally works:

1. **Input/Upload Image**: The process starts with an input image containing text, which could be anything from a scanned document to a photo of a sign.

2. **Data Pre-Processing**: This step may include noise reduction, normalization, resizing, and converting to grayscale to simplify the image data for the model.

3. **Train Model**: This stage involves training the combined CNN with a dataset of pre-processed images and their corresponding text.

    - **Input Layer**: The pre-processed image data is fed into the CNN.

    - **CNN**: First, the CNN part of the model processes the image. CNNs are adept at handling spatial hierarchies in data, making them excellent at recognizing and extracting features from images. In an OCR system, a CNN would identify and extract features such as shapes and lines of text.

    - **Activation Function**: After convolution, an activation function such as ReLU (Rectified Linear Unit) is applied to introduce non-linearity into the model, which allows it to learn more complex patterns.

    - **Pooling Layers**: These layers reduce the spatial dimensions (downsampling) of the feature maps to decrease the computational load, control overfitting, and retain important features.

- **Fully Connected Layers**: Towards the end, the feature maps are flattened and fed into fully connected layers that function like a traditional neural network to perform classification based on the features extracted.

In this CNN architecture we have used VGG 16 model to train the dataset . The VGG-16 model is a convolutional neural network (CNN) architecture that was proposed by the Visual Geometry Group (VGG) at the University of Oxford. It is characterized by its depth, consisting of 16 layers, including 13 convolutional layers and 3 fully connected layers. VGG-16 is renowned for its simplicity and effectiveness, as well as its ability to achieve strong performance on various computer vision tasks, including image classification and object recognition.The training model works iteratively through these steps:

STEP 1: Import necessary libraries such as TensorFlow or PyTorch for deep learning and relevant utilities.

STEP 2: Load biomedical image data and preprocess it, including resizing, normalization, and data augmentation if necessary.

STEP 3: Load the pre-trained VGG16 model, which is typically available in deep learning frameworks like TensorFlow or PyTorch.

STEP 4: Modify the output layers of VGG16 to match the number of classes in your biomedical image classification task. Replace the final fully connected layers with new ones.

STEP 5: Optionally, freeze the weights of the base layers of VGG16 to prevent them from being updated during training. This can be useful if you have limited labeled data.

STEP 6: Compile the modified VGG16 model with an appropriate loss function an optimizer and a metric for evaluation.

STEP 7: Split the dataset into training, validation, and test sets to train, validate, and evaluate the model.

STEP 8: Train the VGG16-based model on the training set using the compiled model. Monitor performance on the validation set to avoid overfitting.

STEP 9: Evaluate the final trained model on the test set to assess its performance in biomedical image classification.

5.3 SOURCE CODE

login.html

```
<!DOCTYPE html>
<html lang="en">
<style>
    .headstyle {
        color: rgb(255, 255, 255);
        font-variant: petite-caps;
        background-color: rgb(0, 0, 0, 0.8);
        margin-bottom: 0px;
    }
    .divstyle {
        border-radius: 10px 10px 10px 10px;
        margin-left: 1px;
        margin-right: 1px;
    }
    .login-container {
        background: linear-gradient(to right,#383838, #f4f0f2);
        padding: 20px;
        border-radius: 10px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
        margin: 20px;
        text-align: center;
    }
    .login-container input[type="submit"] {
        background-color: #4caf50;
        color: white;
        cursor: pointer;
    }
    .login-container input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
crossorigin="anonymous">
    <title>HealthCure</title>
</head>
<body style="background-color: rgb(0, 0, 0, 0.2);">
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <img style="height:70px; width:5%; border-radius:50%;" src="/static/Health Hub
(1).png" alt="">
        <h4 style="color: white;"><b>AI-POWERED BIOMEDICAL IMAGE ANALYSIS</b></h4>        <div
class="container-fluid">
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false"
                aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/index">Index</a>
                    </li>
```

```
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/login">Login</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/register">New
Register</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
    <h1 class='text-center py-3' style="font-variant: petite-caps;margin-
bottom:0px"><b><i>HEALTH HUB</i></b></h1>
    <div class="container login-container" style="color: black;">
        <h1>User Login</h1>
        <form action="/login" method="POST">
            <label for="username">Username:</label>
            <input type="text" name="username" required><br><br>
            <label for="password">Password:</label>
            <input type="password" name="password" required><br><br>
            <input type="submit" value="Login">
        </form>
        <p>Don't have an account? <a href="register.html" style="color:
gray;">Register</a></p>
    </div>
    <footer style="background-color: #1e0505; color: #fff; text-align: center; position: fixed;
bottom: 0; width: 100%; padding: 1px 0;">
        <p>&copy; 2024 AI Powered Biomedical Image Analysis</p>
    </footer>
    <script>
        // Change background color dynamically
        const projectDescription = document.querySelector('.login-container');
        const colors = ['#808080', '#BEBEBE', ' #585858', ' #A9A9A9', ' #BEBEBE', '
#909090', ];
        function changeBackgroundColor() {
            const randomColor = colors[Math.floor(Math.random() * colors.length)];
            projectDescription.style.background = `linear-gradient(to right,
${randomColor},    #F5F5F5)`;
        }
        setInterval(changeBackgroundColor, 3000);
    </script>
</body>
</html>
```

register.html

```
<!DOCTYPE html>
<html lang="en">
<style>
    .headstyle {
        color: rgb(255, 255, 255);
        font-variant: petite-caps;
        background-color: rgb(0, 0, 0, 0.8);
        margin-bottom: 0px
    }
    .divstyle {
        border-radius: 10px 10px 10px 10px;
        margin-left: 1px;
        margin-right: 1px
    }
    .register-container {
```

```
        background: linear-gradient(to right,#383838, #f4f0f2);
        padding: 20px;
        border-radius: 10px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
        margin-top: 20px;
        color: #ffffff;
    }
    .register-container input[type="submit"] {
        background-color: #4caf50;
        color: white;
        cursor: pointer;
    }
    .register-container input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
crossorigin="anonymous">
    <title>HealthCure</title>
</head>
<body style="background-color: rgb(0, 0, 0, 0.2);">
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <img style="height:70px; width:5%; border-radius:50%;" src="/static/Health Hub
(1).png" alt="">
        <h4 style="color: white;"><b>AI-POWERED BIOMEDICAL IMAGE ANALYSIS</b></h4>
         <div class="container-fluid">
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false"
                aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/index">Index</a></li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/login">Login</a></li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/register">New
Register</a></li>
                </ul>
            </div>
        </div>
    </nav>
    <h1 class='text-center py-3' style="font-variant: petite-caps;margin-
bottom:0px"><b><i>HEALTH HUB</i></b></h1>
    <div class="container register-container" style="text-align: center;">
        <h1>User Registration</h1>
        <form action="/register" method="POST">
            <label for="username">Username:</label>
            <input type="text" name="username" required><br><br>
            <label for="password">Password:</label>
            <input type="password" name="password" required><br><br>
            <label for="email">Email ID:</label>
```

```
            <input type="email" id="email" name="email" placeholder="Enter email"
name="email"><br><br>
            <input type="submit" value="Register" >
        </form>
        <p>Already have an account? <a href="login.html" style="color: #ffffff;">Login</a></p>
    </div>
    <footer style="background-color: #1e0505; color: #fff; text-align: center; position: fixed;
bottom: 0; width: 100%; padding: 1px 0;">
        <p>&copy; 2024 AI Powered Biomedical Image Analysis</p>
    </footer>
</body>
</html>
```

homepage.html

```
<!DOCTYPE html>
<html lang="en">
<style>
    .headstyle {
        color: rgb(255, 255, 255);
        font-variant: petite-caps;
        background-color: rgb(0, 0, 0, 0.8);
        margin-bottom: 0px;
    }
    .navbar {
        position: fixed;
    }
    .divstyle {
        border-radius: 10px 10px 10px 10px;
        margin-left: 1px;
        margin-right: 1px;
    }
    .project-description {
        background: linear-gradient(to right,#383838, #f4f0f2);
        padding: 35px;
        padding-bottom: 20px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
        margin: 20px;
        color: #ffffff;
        text-align: justify;
        border-radius: 10px 10px 10px 10px;
        margin-left: 10px;
        margin-right: 10px
    }
    .gallery {
        width: 100%;
        border-collapse: collapse;
    }
    .gallery td {
        width: 33%;
        padding: 10px;
        text-align: center;
    }
    .gallery img {
        max-width: 100%;
        height: auto;
        display: block;
        margin: 0 auto;
    }
</style>
```

```
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
crossorigin="anonymous">
    <title>Health Hub</title>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
        <img style="height:70px; width:5%; border-radius:50%;" src="/static/Health Hub
(1).png" alt="">
        <h4 style="color: white;"><b>AI-POWERED BIOMEDICAL IMAGE ANALYSIS</b></h4>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false"
                aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/braintumor">Brain
Tumor</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/breastcancer">Breast
Cancer</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page"
href="/alzheimer">Alzheimer</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page"
href="/pneumonia">Pneumonia</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/heartdisease">Heart
Disease</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/logout1">LogOut</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
        <div class='divstyle' style='margin:40px 20px 60px 20px'>
            <div class="row py-3">
                <div class="col md-3">
                    <h3 class='text-center py-3 headstyle' style="font-size: 18px;"><b>Brain
Tumor Detection</b></h3>
                    <a href="./braintumor"><img src="../static/icons/braintumor.png"
                            class="img-fluid mx-auto d-block"></a>
                </div>
                <div class="col md-3">
                    <h3 class='text-center py-3 headstyle' style="font-size: 18px;"><b>Breast
Cancer Detection</b></h3>
```

```
                    <a href="./breastcancer"><img src="../static/icons/breastcancer.png"
                            class="img-fluid mx-auto d-block"></a>
                </div>
                <div class="col md-3">
                    <h3 class='text-center py-3 headstyle' style="font-size:
18px;"><b>Alzheimer Detection</b></h3>
                    <a href="./alzheimer"><img src="../static/icons/alzheimer.png"
                            class="img-fluid mx-auto d-block"></a>
                </div>
                <div class="col md-3">
                    <h3 class='text-center py-3 headstyle' style="font-size:
18px;"><b>Pneumonia Detection</b></h3>
                    <a href="./pneumonia"><img src="../static/icons/pneumonia.png"
                            class="img-fluid mx-auto d-block"></a>
                </div>
                <div class="col md-3">
                    <h3 class='text-center py-3 headstyle' style="font-size: 18px;"><b>Heart
Disease Detection</b></h3>
                    <a href="./heartdisease"><img src="../static/icons/heartdisease.png"
                            class="img-fluid mx-auto d-block"></a>
                </div>
                <div class="col md-3">
                </div>
            </div>
        </div>
</body>
<footer style="background-color: #1e0505; color: #fff; text-align: center; position: fixed;
bottom: 0; width: 100%; padding: 1px 0;">
    <p>&copy; 2024 AI Powered Biomedical Image Analysis</p>
</footer>
</html>
```

braintumor.html

```
<!doctype html>
<html lang="en">
<style type='text/css'>
    body {
        background-image: url(static/bt.png);
        background-position: center;
        background-size: cover;
        font-family: sans-serif;
        margin-top: 40px;
    }
    .regform {
        width: 800px;
        background-color: rgb(0, 0, 0, 0.8);
        margin: auto;
        color: #FFFFFF;
        padding: 10px 0px 10px 0px;
        text-align: center;
        border-radius: 15px 15px 0px 0px;
    }
    .main-form {
        width: 800px;
        margin: auto;
        background-color: rgb(0, 0, 0, 0.7);
        padding-left: 50px;
        padding-right: 50px;
        padding-bottom: 20px;
```

```
        color: #FFFFFF;
    }
    .form-group {
        margin-top: 5px;
        margin-bottom: 5px;
    }
</style>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
crossorigin="anonymous">
    <title>Brain Tumor Detection</title>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
        <img style="height:70px; width:5%; border-radius:50%;" src="/static/Health Hub
(1).png" alt="">
        <h4 style="color: white;"><b>AI-POWERED BIOMEDICAL IMAGE ANALYSIS</b></h4>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false"
                aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/home1">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " class="active" aria-current="page"
href="/braintumor">Brain Tumor</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/breastcancer">Breast
Cancer</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page"
href="/alzheimer">Alzheimer</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page"
href="/pneumonia">Pneumonia</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/heartdisease">Heart
Disease</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/logout1">LogOut</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
    <div class='regform mt-3'>
```

```
        <h1>Brain Tumor Detection</h1>
    </div>
    <form action='resultbt' class='main-form needs-validation' method="POST"
enctype="multipart/form-data">
        <div class="row">
            <div class="col">
                <div class='form-group'>
                    <label for="firstname">Firstname</label>
                    <input type="text" name="firstname" id="firstname" class="form-control"
required='True'>
                </div>
            </div>
            <div class="col">
                <div class='form-group'>
                    <label for="lastname">Lastname</label>
                    <input type="text" name="lastname" id="lastname" class="form-control"
required='True'>
                </div>
            </div>
        </div>
        <div class='form-group'>
            <label for="phone">Phone No.</label>
            <input type="number" name="phone" id="phone" class="form-control" required='True'>
            <small class='form-text ' style='color: #FFFFFF;'>* Include your area code</small>
        </div>
        <div class='form-group'>
            <label for="email">Email</label>
            <input type="email" name="email" id="email" class="form-control" required='True'>
        </div>
        <div class="row">
            <div class="col">
                <div class="form-group">
                    <label for="gender">Gender</label>
                    <select name="gender" id="gender" class="form-control" required='True'>
                        <option value="male">Male</option>
                        <option value="female">Female</option>
                    </select>
                </div>
            </div>
            <div class="col">
                <div class='form-group'>
                    <label for="age">Age</label>
                    <input type="number" name="age" id="age" class="form-control"
required='True'>
                </div>
            </div>
        </div>
        <div class='form-group'>
            <label for="inputGroupFile02">Upload your Brain MRI </label>
            <input type="file" class="form-control" id="file" name='file'>
        </div>
        <div class='text-center'>
            <button type='submit' class='btn btn-outline-success' style='width:700px;margin-
top: 30px;'> Submit
            </button>
        </div>
    </form>
</body>
</html>
```

alzheimer.html

```
<!doctype html>
<html lang="en">
<style type='text/css'>
    body {
        background-image: url(static/al.png);
            background-position: center;
            background-size: cover;
        font-family: sans-serif;
        margin-top: 40px;
    }
    .regform {
        width: 800px;
        background-color: rgb(0, 0, 0, 0.8);
        margin: auto;
        color: #FFFFFF;
        padding: 10px 0px 10px 0px;
        text-align: center;
        border-radius: 15px 15px 0px 0px;

    }
    .main-form {
        width: 800px;
        margin: auto;
        background-color: rgb(0, 0, 0, 0.7);
        padding-left: 50px;
        padding-right: 50px;
        padding-bottom: 20px;
        color: #FFFFFF;
    }
    .form-group {
        margin-top: 5px;
        margin-bottom: 5px;
    }
</style>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
crossorigin="anonymous">
    <title>Alzheimer Detection</title>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
        <img style="height:70px; width:5%; border-radius:50%;" src="/static/Health Hub
(1).png" alt="">
        <h4 style="color: white;"><b>AI-POWERED BIOMEDICAL IMAGE ANALYSIS</b></h4>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false"
                aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/home1">Home</a>
```

```
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/braintumor">Brain
Tumor</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/breastcancer">Breast
Cancer</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page"
href="/alzheimer">Alzheimer</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page"
href="/pneumonia">Pneumonia</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/heartdisease">Heart
Disease</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/logout1">LogOut</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
    <div class='regform mt-3'>
        <h1>Alzheimer Detection</h1>
    </div>
    <form action='resulta' class='main-form needs-validation' method="POST"
enctype="multipart/form-data">
        <div class="row">
            <div class="col">
                <div class='form-group'>
                    <label for="firstname">Firstname</label>
                    <input type="text" name="firstname" id="firstname" class="form-control"
required='True'>
                </div>
            </div>
            <div class="col">
                <div class='form-group'>
                    <label for="lastname">Lastname</label>
                    <input type="text" name="lastname" id="lastname" class="form-control"
required='True'>
                </div>
            </div>
        </div>
        <div class='form-group'>
            <label for="phone">Phone No.</label>
            <input type="number" name="phone" id="phone" class="form-control" required='True'>
            <small class='form-text ' style='color: #FFFFFF;'>* Include your area code</small>
        </div>
        <div class='form-group'>
            <label for="email">Email</label>
            <input type="email" name="email" id="email" class="form-control" required='True'>
        </div>
        <div class="row">
            <div class="col">
                <div class="form-group">
```

```
                    <label for="gender">Gender</label>
                    <select name="gender" id="gender" class="form-control" required='True'>
                        <option value="male">Male</option>
                        <option value="female">Female</option>
                    </select>
                </div>
            </div>
            <div class="col">
                <div class='form-group'>
                    <label for="age">Age</label>
                    <input type="number" name="age" id="age" class="form-control"
required='True'>
                </div>
            </div>
        </div>
        <div class='form-group'>
            <label for="inputGroupFile02">Upload your Brain MRI </label>
            <input type="file" class="form-control" id="file" name='file'>
        </div>
        <div class='text-center'>
            <button type='submit' class='btn btn-outline-success' style='width:700px;margin-
top: 30px;'> Submit
            </button>
        </div>
    </form>
</body>
</html>
```

pneumonia.html

```
<!doctype html>
<html lang="en">
<style type='text/css'>
    body {
        background-image: url(static/al.png);
            background-position: center;
            background-size: cover;
        font-family: sans-serif;
        margin-top: 40px;
    }
    .regform {
        width: 800px;
        background-color: rgb(0, 0, 0, 0.8);
        margin: auto;
        color: #FFFFFF;
        padding: 10px 0px 10px 0px;
        text-align: center;
        border-radius: 15px 15px 0px 0px;

    }
    .main-form {
        width: 800px;
        margin: auto;
        background-color: rgb(0, 0, 0, 0.7);
        padding-left: 50px;
        padding-right: 50px;
        padding-bottom: 20px;
        color: #FFFFFF;
    }
    .form-group {
```

```
        margin-top: 5px;
        margin-bottom: 5px;
    }
</style>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
crossorigin="anonymous">
    <title>Alzheimer Detection</title>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
        <img style="height:70px; width:5%; border-radius:50%;" src="/static/Health Hub
(1).png" alt="">
        <h4 style="color: white;"><b>AI-POWERED BIOMEDICAL IMAGE ANALYSIS</b></h4>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false"
                aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/home1">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/braintumor">Brain
Tumor</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/breastcancer">Breast
Cancer</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page"
href="/alzheimer">Alzheimer</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page"
href="/pneumonia">Pneumonia</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/heartdisease">Heart
Disease</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/logout1">LogOut</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
    <div class='regform mt-3'>
        <h1>Alzheimer Detection</h1>
    </div>
```

```
    <form action='resulta' class='main-form needs-validation' method="POST"
enctype="multipart/form-data">
        <div class="row">
            <div class="col">
                <div class='form-group'>
                    <label for="firstname">Firstname</label>
                    <input type="text" name="firstname" id="firstname" class="form-control"
required='True'>
                </div>
            </div>
            <div class="col">
                <div class='form-group'>
                    <label for="lastname">Lastname</label>
                    <input type="text" name="lastname" id="lastname" class="form-control"
required='True'>
                </div>
            </div>
        </div>
        <div class='form-group'>
            <label for="phone">Phone No.</label>
            <input type="number" name="phone" id="phone" class="form-control" required='True'>
            <small class='form-text ' style='color: #FFFFFF;'>* Include your area code</small>
        </div>
        <div class='form-group'>
            <label for="email">Email</label>
            <input type="email" name="email" id="email" class="form-control" required='True'>
        </div>
        <div class="row">
            <div class="col">
                <div class="form-group">
                    <label for="gender">Gender</label>
                    <select name="gender" id="gender" class="form-control" required='True'>
                        <option value="male">Male</option>
                        <option value="female">Female</option>
                    </select>
                </div>
            </div>
            <div class="col">
                <div class='form-group'>
                    <label for="age">Age</label>
                    <input type="number" name="age" id="age" class="form-control"
required='True'>
                </div>
            </div>
        </div>
        <div class='form-group'>
            <label for="inputGroupFile02">Upload your Brain MRI </label>
            <input type="file" class="form-control" id="file" name='file'>
        </div>
        <div class='text-center'>
            <button type='submit' class='btn btn-outline-success' style='width:700px;margin-
top: 30px;'> Submit
            </button>
        </div>
    </form>
</body>
</html>
```

resultAlzheimer.html

```
<!doctype html>
<html lang="en">
<style type='text/css'>
    body {
        background-image: url(static/result.png);
        background-position: center;
        background-size: cover;
        font-family: sans-serif;
        margin-top: 40px;
    }
    .regform {
        width: 800px;
        background-color: rgba(253, 252, 252, 0.8);
        margin: auto;
        color: #0f0f0f;
        padding: 10px 0px 10px 0px;
        text-align: center;
        border-radius: 15px 15px 0px 0px;
    }
    .main-form {
        width: 800px;
        margin: auto;
        background-color: rgb(0, 0, 0, 0.7);
        padding-left: 50px;
        padding-right: 50px;
        padding-bottom: 20px;
        color: #FFFFFF;
    }
    .form-group {
        margin-top: 5px;
        margin-bottom: 5px;
    }
    p{
        font-size:20px;
        font-family: sans-serif;
    }
</style>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
crossorigin="anonymous">
    <title>Alzheimer Detection</title>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
        <img style="height:70px; width:5%; border-radius:50%;" src="/static/Health Hub
(1).png" alt="">
        <h4 style="color: white;"><b>AI-POWERED BIOMEDICAL IMAGE ANALYSIS</b></h4>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false"
                aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
```

```
                <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/home1">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/braintumor">Brain
Tumor</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/breastcancer">Breast
Cancer</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page"
href="/alzheimer">Alzheimer</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page"
href="/pneumonia">Pneumonia</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/heartdisease">Heart
Disease</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link " aria-current="page" href="/logout1">LogOut</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
    <div class='regform mt-3'>
        <h1>Alzheimer Test Results </h1>
    </div>
    <div class='main-form'>
        <div >
            <div class="row">
                <div class="col">
                    <img src="static/uploads/{{filename}}"
                        style='width:250px;height:250px;margin-top: 30px;margin-bottom:30px;'>
                </div>
                <div class="col"  style='margin-top: 30px;margin-bottom:30px;'>
                    <p>First Name : {{fn}}</p>
                    <p>Last Name : {{ln}}</p>
                    <p>Age : {{age}}</p>
                    <p> Gender: {{gender}}</p>
                    {% if r==0 %}
                    <div>
                        <p> Result:<i> NonDemented </i></p>
                    </div>
                    {% elif r==1 %}
                    <div>
                        <p> Result:<i> VeryMildDemented </i></p>
                    </div>
                    {% elif r==2 %}
                    <div>
                        <p> Result:<i> MildDemented </i></p>
                    </div>
                    {% else %}
                    <div>
                        <p> Result:<i> ModerateDemented </i></p>
```

```
                    </div>
                    {% endif %}
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

db.sql

```
/*
SQLyog Community v13.1.7 (64 bit)
MySQL - 5.5.29 : Database - earth1
*********************************************************************
*/
/*!40101 SET NAMES utf8 */;
/*!40101 SET SQL_MODE=''*/;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
CREATE DATABASE /*!32312 IF NOT EXISTS*/`biomedical` /*!40100 DEFAULT CHARACTER SET latin1 */;
USE `biomedical`;

/*Table structure for table `users` */
DROP TABLE IF EXISTS `users`;

CREATE TABLE `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `email` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;

/*Data for the table `users` */

insert  into `users`(`id`,`username`,`password`,`email`) values
(1,'kavya','kavya123',NULL),
(2,'vishala','vishu',NULL),
(3,'sindhu','srambha',NULL),
(4,'raju','raju','raju@gmail.com');

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

app.py

```
import base64
import io
from flask import Flask, render_template, request, redirect, url_for, session, flash, jsonify
from flaskext.mysql import MySQL
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn import ensemble
from sklearn.model_selection import train_test_split
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from flask import Flask, render_template, request
import requests
import numpy as np
from flask import Flask, flash, request, redirect, url_for, render_template
import urllib.request
import os
from werkzeug.utils import secure_filename
import cv2
import pickle
import imutils
import sklearn
from tensorflow.keras.models import load_model
# from pushbullet import PushBullet
import joblib
import numpy as np
from tensorflow.keras.applications.vgg16 import preprocess_input

app = Flask(__name__)
app.secret_key = 'your_secret_key'  # Change to a strong, secret key
mysql = MySQL(app)
# MySQL configuration
app.config['MYSQL_DATABASE_USER'] = 'root'
app.config['MYSQL_DATABASE_PASSWORD'] = 'root'
app.config['MYSQL_DATABASE_DB'] = 'biomedical'
app.config['MYSQL_DATABASE_HOST'] = '127.0.0.1'  # Modify as needed
mysql.init_app(app)

# Loading Models
braintumor_model = load_model('models/braintumor.h5')
alzheimer_model = load_model('models/alzheimer_model.h5')
heart_model = pickle.load(open('models/heart_disease.pickle.dat', "rb"))
pneumonia_model = load_model('models/pneumonia_model.h5')
breastcancer_model = joblib.load('models/cancer_model.pkl')

# Configuring Flask
UPLOAD_FOLDER = 'static/uploads'
ALLOWED_EXTENSIONS = set(['png', 'jpg', 'jpeg'])
app = Flask(__name__)
app.config['SEND_FILE_MAX_AGE_DEFAULT'] = 0
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
app.secret_key = "secret key"

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1] in ALLOWED_EXTENSIONS

#BRAIN TUMOR FUNCTIONS #
def preprocess_imgs(set_name, img_size):
    """ Resize and apply VGG-15 preprocessing """
    set_new = []
    for img in set_name:
        img = cv2.resize(img,dsize=img_size,interpolation=cv2.INTER_CUBIC)
        set_new.append(preprocess_input(img))
    return np.array(set_new)
def crop_imgs(set_name, add_pixels_value=0):
    """ Finds the extreme points on the image and crops the rectangular out of them """
    set_new = []
    for img in set_name:
        gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
```

```
        gray = cv2.GaussianBlur(gray, (5, 5), 0)
        thresh = cv2.threshold(gray, 45, 255, cv2.THRESH_BINARY)[1]
        thresh = cv2.erode(thresh, None, iterations=2)
        thresh = cv2.dilate(thresh, None, iterations=2)
        cnts = cv2.findContours(
            thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        cnts = imutils.grab_contours(cnts)
        c = max(cnts, key=cv2.contourArea)
        extLeft = tuple(c[c[:, :, 0].argmin()][0])
        extRight = tuple(c[c[:, :, 0].argmax()][0])
        extTop = tuple(c[c[:, :, 1].argmin()][0])
        extBot = tuple(c[c[:, :, 1].argmax()][0])
        ADD_PIXELS = add_pixels_value
        new_img = img[extTop[1]-ADD_PIXELS:extBot[1]+ADD_PIXELS,
                      extLeft[0]-ADD_PIXELS:extRight[0]+ADD_PIXELS].copy()
        set_new.append(new_img)
    return np.array(set_new)

# Set the title and logo
app.config['APP_TITLE'] = "AI POWERED BIOMEDICAL IMAGE ANALYSIS"
app.config['APP_LOGO'] = './static/logo.png'
@app.route('/')
def home():
    if 'username' in session:
        return render_template('homepage.html')
    return redirect(url_for('index'))
@app.route('/logout1')
def logout1():
    return redirect(url_for('login'))
@app.route('/index')
def index():
    return render_template('index.html')
@app.route('/home1')
def home1():
    return render_template('homepage.html')
@app.route('/breastcancer')
def breast_cancer():
    return render_template('breastcancer.html')
@app.route('/braintumor')
def brain_tumor():
    return render_template('braintumor.html')
@app.route('/alzheimer')
def alzheimer():
    return render_template('alzheimer.html')
@app.route('/pneumonia')
def pneumonia():
    return render_template('pneumonia.html')
@app.route('/heartdisease')
def heartdisease():
    return render_template('heartdisease.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        cursor = mysql.get_db().cursor()
        cursor.execute("SELECT * FROM users WHERE username = %s AND password = %s", (username, 
password))
        account = cursor.fetchone()
        if account:
```

```
            session['username'] = username
            flash('Logged in successfully', 'success')
            return redirect(url_for('home'))
        else:
            flash('Login failed. Please check your credentials and try again.', 'danger')
    return render_template('login.html', title=app.config['APP_TITLE'],
logo=app.config['APP_LOGO'])
@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        email = request.form['email']
        cursor = mysql.get_db().cursor()
        cursor.execute("INSERT INTO users (username, password, email) VALUES (%s, %s, %s)",
(username, password,email))
        mysql.get_db().commit()
        flash('Registration successful. Please log in.', 'success')
        return redirect(url_for('login'))
    return render_template('register.html', title=app.config['APP_TITLE'],
logo=app.config['APP_LOGO'])
@app.route('/logout')
def logout():
    session.pop('username', None)
    flash('Logged out', 'success')
    return redirect(url_for('login'))
@app.route('/resultbt', methods=['POST'])
def resultbt():
    if request.method == 'POST':
        firstname = request.form['firstname']
        lastname = request.form['lastname']
        email = request.form['email']
        phone = request.form['phone']
        gender = request.form['gender']
        age = request.form['age']
        file = request.files['file']
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
            flash('Image successfully uploaded and displayed below')
            img = cv2.imread('static/uploads/'+filename)
            img = crop_imgs([img])
            img = img.reshape(img.shape[1:])
            img = preprocess_imgs([img], (224, 224))
            pred = braintumor_model.predict(img)
            if pred < 0.5:
                pred = 0
            else:
                pred = 1
            return render_template('resultbt.html', filename=filename, fn=firstname,
ln=lastname, age=age, r=pred, gender=gender)
        else:
            flash('Allowed image types are - png, jpg, jpeg')
            return redirect(request.url)
@app.route('/resultbc', methods=['POST'])
def resultbc():
    if request.method == 'POST':
        firstname = request.form['firstname']
        lastname = request.form['lastname']
        email = request.form['email']
        phone = request.form['phone']
```

```
        gender = request.form['gender']
        age = request.form['age']
        cpm = request.form['concave_points_mean']
        am = request.form['area_mean']
        rm = request.form['radius_mean']
        pm = request.form['perimeter_mean']
        cm = request.form['concavity_mean']
        pred = breastcancer_model.predict(
            np.array([cpm, am, rm, pm, cm]).reshape(1, -1))
        return render_template('resultbc.html', fn=firstname, ln=lastname, age=age, r=pred,
gender=gender)
@app.route('/resulta', methods=['GET', 'POST'])
def resulta():
    if request.method == 'POST':
        print(request.url)
        firstname = request.form['firstname']
        lastname = request.form['lastname']
        email = request.form['email']
        phone = request.form['phone']
        gender = request.form['gender']
        age = request.form['age']
        file = request.files['file']
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
            flash('Image successfully uploaded and displayed below')
            img = cv2.imread('static/uploads/'+filename)
            img = cv2.resize(img, (176, 176))
            img = img.reshape(1, 176, 176, 3)
            img = img/255.0
            pred = alzheimer_model.predict(img)
            pred = pred[0].argmax()
            print(pred)
            return render_template('resulta.html', filename=filename, fn=firstname,
ln=lastname, age=age, r=0, gender=gender)
        else:
            flash('Allowed image types are - png, jpg, jpeg')
            return redirect('/')
@app.route('/resultp', methods=['POST'])
def resultp():
    if request.method == 'POST':
        firstname = request.form['firstname']
        lastname = request.form['lastname']
        email = request.form['email']
        phone = request.form['phone']
        gender = request.form['gender']
        age = request.form['age']
        file = request.files['file']
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
            flash('Image successfully uploaded and displayed below')
            img = cv2.imread('static/uploads/'+filename)
            img = cv2.resize(img, (150, 150))
            img = img.reshape(1, 150, 150, 3)
            img = img/255.0
            pred = pneumonia_model.predict(img)
            if pred < 0.5:
                pred = 0
            else:
                pred = 1
```

```
            return render_template('resultp.html', filename=filename, fn=firstname,
ln=lastname, age=age, r=pred, gender=gender)
        else:
            flash('Allowed image types are - png, jpg, jpeg')
            return redirect(request.url)
@app.after_request
def add_header(response):
    """
    Add headers to both force latest IE rendering engine or Chrome Frame,
    and also to cache the rendered page for 10 minutes.
    """
    response.headers['X-UA-Compatible'] = 'IE=Edge,chrome=1'
    response.headers['Cache-Control'] = 'public, max-age=0'
    return response
if __name__ == '__main__':
    app.run(debug=True)
```

alzheimer.ipynb

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from tensorflow.keras import Sequential, Input
from tensorflow.keras.layers import Dense, Dropout, Conv2D, Flatten, SeparableConv2D, BatchNormalization,
MaxPool2D
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.preprocessing.image import ImageDataGenerator as IDG
BASE_DIR = "Alzheimer_Dataset/"
TRAIN_DIR = BASE_DIR + 'train'
TEST_DIR = BASE_DIR + 'test'
CLASSES = [ 'NonDemented',
            'VeryMildDemented',
            'MildDemented',
            'ModerateDemented']
IMG_SIZE = 176
DIM = (IMG_SIZE, IMG_SIZE)
datagen = IDG(
    rescale = 1./255,
    brightness_range=[0.8, 1.2],
    zoom_range=[.99, 1.01],
    data_format="channels_last",
    fill_mode='constant',
    horizontal_flip=True
)
```

```
train_data_gen = datagen.flow_from_directory(directory=TRAIN_DIR, target_size=DIM, batch_size=6500, shuffle=False)
#Retrieving the data from the ImageDataGenerator iterator
train_data, train_labels = train_data_gen.next()
#Getting to know the dimensions of our dataset
print(train_data.shape, train_labels.shape)
#Splitting the data into train, test, and validation sets
train_data, val_data, train_labels, val_labels = train_test_split(train_data, train_labels, test_size = 0.2, random_state=42)
def conv_block(filters, act='relu'):
    """Defining a Convolutional NN block for a Sequential CNN model. """
    block = Sequential()
    block.add(Conv2D(filters, 3, activation=act, padding='same'))
    block.add(Conv2D(filters, 3, activation=act, padding='same'))
    block.add(BatchNormalization())
    block.add(MaxPool2D())
    return block
def dense_block(units, dropout_rate, act='relu'):
    """Defining a Dense NN block for a Sequential CNN model. """
    block = Sequential()
    block.add(Dense(units, activation=act))
    block.add(BatchNormalization())
    block.add(Dropout(dropout_rate))
    return block
def construct_model(act='relu'):
    """Constructing a Sequential CNN architecture for performing the classification task. """
    model = Sequential([
        Input(shape=(*[IMG_SIZE,IMG_SIZE], 3)),
        Conv2D(16, 3, activation=act, padding='same'),
        Conv2D(16, 3, activation=act, padding='same'),
        MaxPool2D(),
        conv_block(32),
        conv_block(64),
        conv_block(128),
        Dropout(0.2),
        conv_block(256),
        Dropout(0.2),
        Flatten(),
        dense_block(512, 0.7),
        dense_block(128, 0.5),
        dense_block(64, 0.3),
        Dense(4, activation='softmax')
    ], name = "cnn_model")
    return model
```

```
class MyCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if logs.get('val_acc') > 0.99:
            print("\nReached accuracy threshold! Terminating training.")
            self.model.stop_training = True
my_callback = MyCallback()
CALLBACKS = [my_callback]
class MyCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if logs.get('val_acc') > 0.99:
            print("\nReached accuracy threshold! Terminating training.")
            self.model.stop_training = True
my_callback = MyCallback()
CALLBACKS = [my_callback]
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs_range = range(1, len(history.epoch) + 1)
plt.figure(figsize=(15,5))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Train Set')
plt.plot(epochs_range, val_acc, label='Val Set')
plt.legend(loc="best")
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Model Accuracy')
plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Train Set')
plt.plot(epochs_range, val_loss, label='Val Set')
plt.legend(loc="best")
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Model Loss')
plt.tight_layout()
plt.show()
pred_ls = np.argmax(pred_labels, axis=1)
test_ls = np.argmax(test_labels, axis=1)
conf_arr = confusion_matrix(test_ls, pred_ls)
plt.figure(figsize=(8, 6), dpi=80, facecolor='w', edgecolor='k')
ax = sns.heatmap(conf_arr, cmap='Greens', annot=True, fmt='d', xticklabels=CLASSES, yticklabels=CLASSES)
plt.title('Alzheimer\'s Disease Diagnosis')
plt.xlabel('Prediction')
plt.ylabel('Truth')
plt.show(ax)
```

CHAPTER 6: TESTING

Testing is an essential phase in the development of our AI-powered biomedical image analysis system, aimed at ensuring its functionality, accuracy, and security. The testing process involves a comprehensive evaluation of the system's components and workflows to verify that they meet specified requirements and perform as expected. Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition.

6.1 UNIT TESTING

In your project, unit testing ensures that each component of the system, such as the data preprocessing algorithms and CNN layers, functions correctly in isolation. For example, unit tests can validate that the data cleaning module effectively removes noise and artifacts from biomedical images, ensuring the integrity of the input data for disease detection.

6.2 INTEGRATION TESTING

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

6.3 SYSTEM TESTING

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

6.4 MODULE TESTING

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

6.5 ACCEPTANCE TESTING

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

6.6 PERFORMANCE TESTING

Performance testing assesses how the system performs under different workload conditions, such as processing images of varying sizes or handling multiple concurrent user requests. For example, performance tests can measure the system's response time and throughput to ensure that it can efficiently process a large volume of biomedical images without compromising accuracy or speed.

CHAPTER 7: RESULTS

7.1 DATASET ANALYSIS

The AI-powered biomedical image analysis system relies on comprehensive datasets comprising various biomedical images depicting different diseases and conditions. These datasets are essential for training and evaluating the performance of the machine learning models, particularly Convolutional Neural Networks (CNNs). The datasets are carefully curated to encompass a wide range of medical conditions.

DISEASE DATASET	NO. OF IMAGES
Brain Tumor	291
Breast Cancer	570
Alzheimer	6400
Pneumonia	6270
Heart Disease	304

Table 7.1.1: Analysis on Dataset

We used different datasets for each disease. The diseases are Brain Tumor, Breast Cancer, Alzheimer, Pneumonia, Heart Disease. The given table depicts the dataset of each disease. We took 80% of data for training and 20% of data for testing.

DISEASE DATASET	TRAIN	TEST
Brain Tumor	233	58
Breast Cancer	456	114
Alzheimer	5120	1280
Pneumonia	5226	1045
Heart Disease	243	61

Table 7.1.2: Analysis on Train Test Split

7.2 RESULT ANALYSIS

The accuracy of each dataset obtained through Convolutional Neural Networks (CNNs) reflects the system's capability to precisely classify biomedical images, contributing to the early and accurate diagnosis of diseases. Through rigorous training and evaluation processes, the CNN-based model achieves notable accuracies, demonstrating its proficiency in distinguishing between different medical conditions depicted in the datasets. These high accuracy rates underscore the effectiveness and reliability of the CNN algorithm in automated biomedical image analysis, paving the way for enhanced patient care and improved healthcare outcomes.

I calculated accuracy with below formulae:

$$\text{Accuracy} = \frac{\text{Predicted images}}{\text{Total no.of images}}$$

DISEASE DATASET	ACCURACY
Brain Tumor	**90.53**
Breast Cancer	**89.76**
Alzheimer	**92.56**
Pneumonia	**94.89**
Heart Disease	**93.97**

Table 7.2.1: Accuracy on Each Dataset

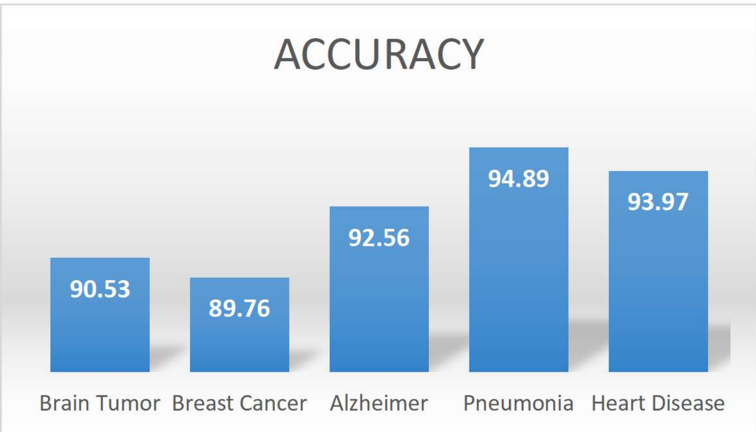

Fig 7.2.1: Bar Graph On Accuracy

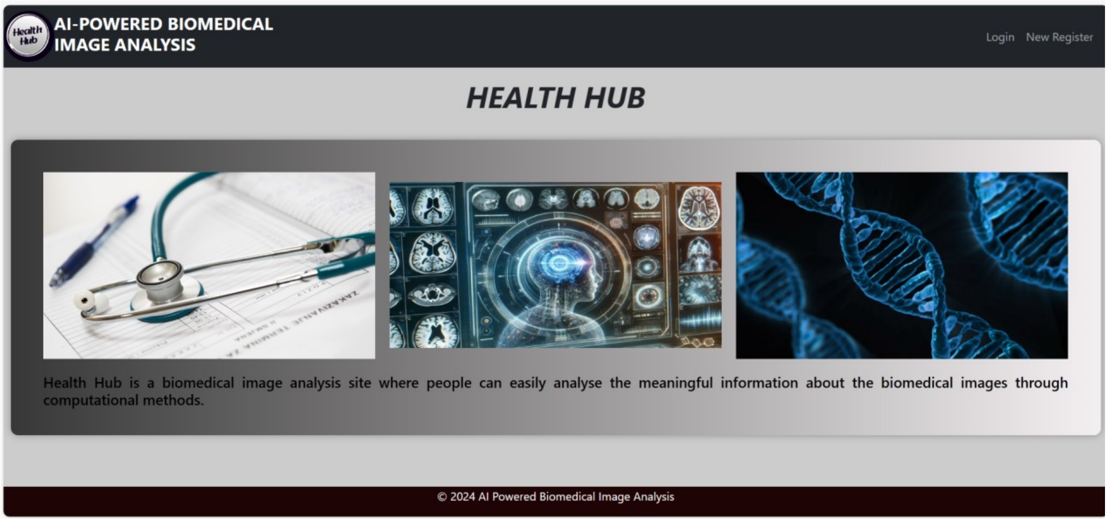

Fig 7.2.2: Index Page

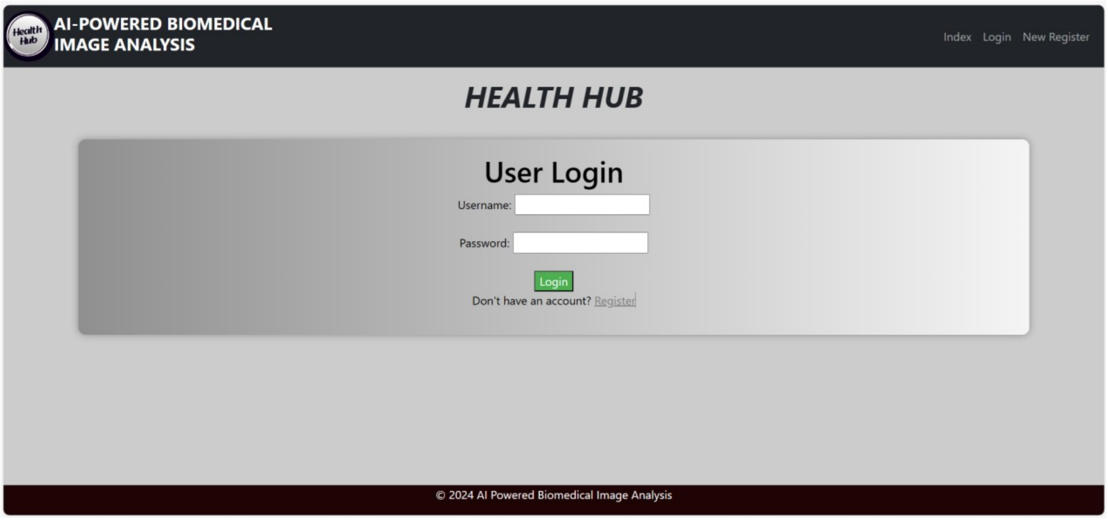

Fig 7.2.3: Login Page

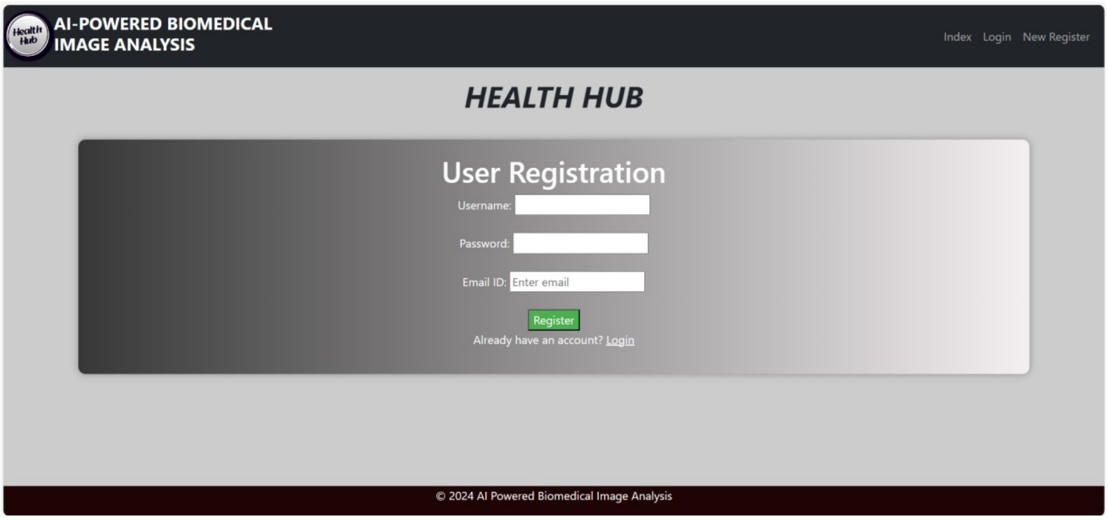

Fig 7.2.4: Registration Page

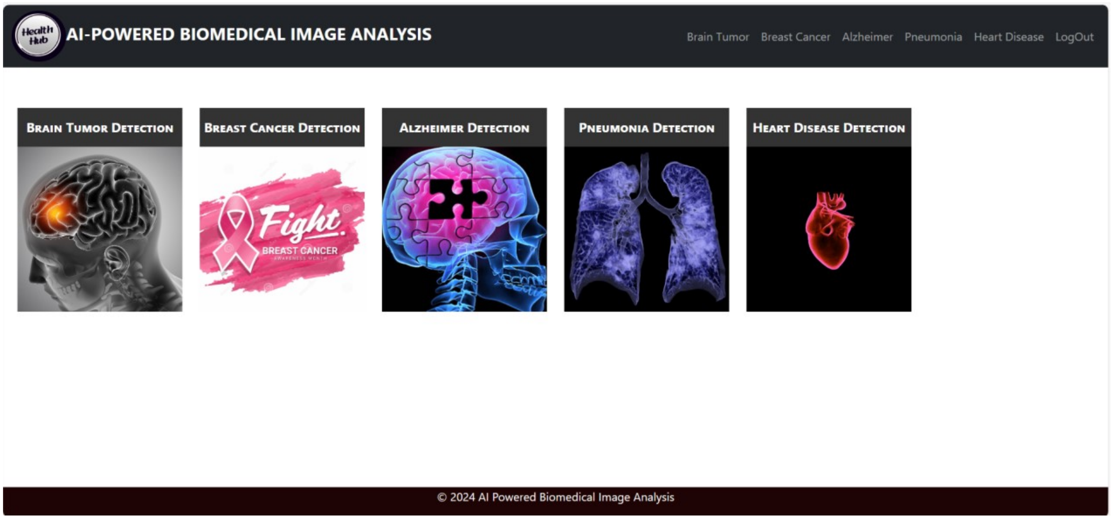

Fig 7.2.5: Home Page

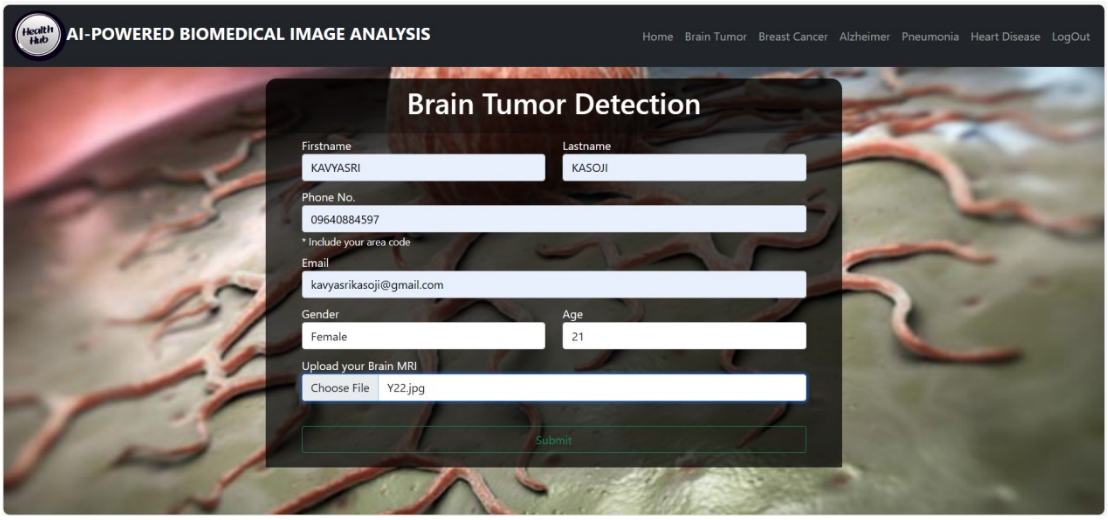

Fig 7.2.6: Patient Details for Brain Tumor

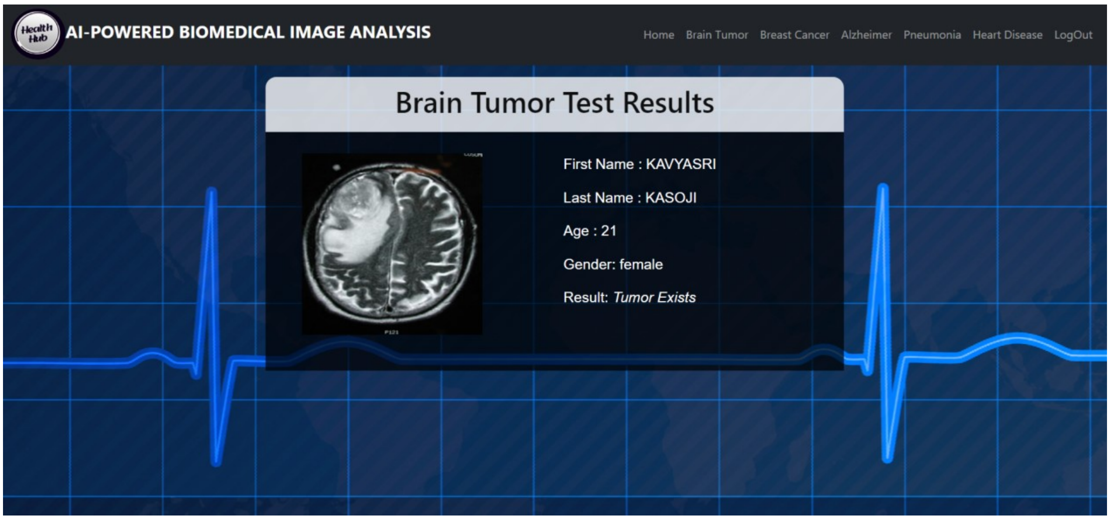

Fig 7.2.7: Result Page for Brain Tumor

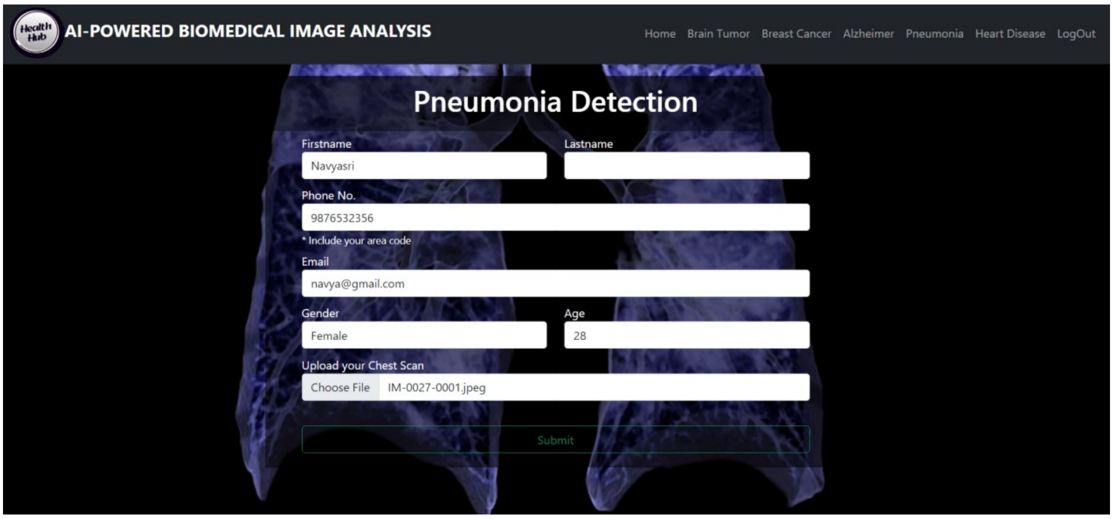

Fig 7.2.8: Patient Details for Pneumonia

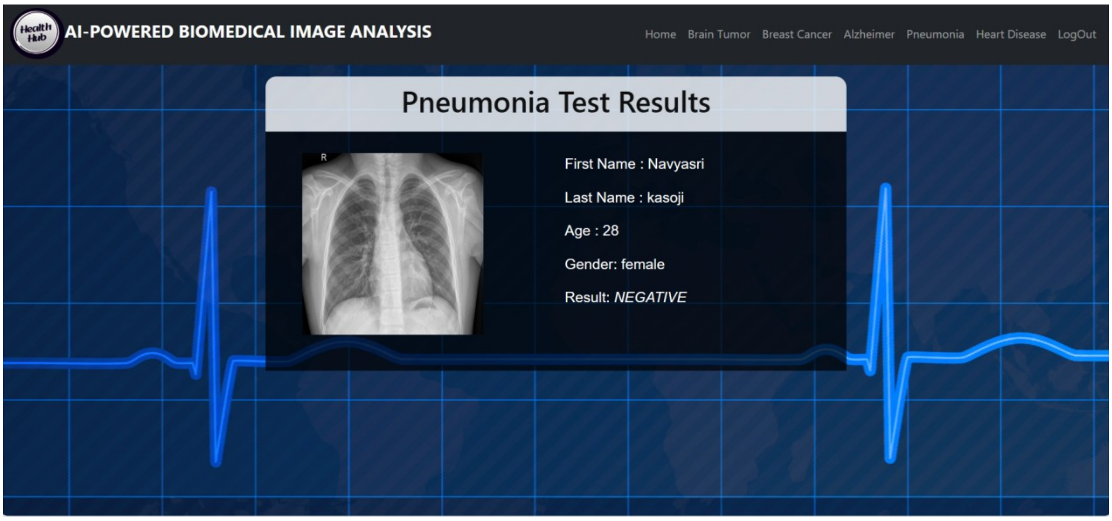

Fig 7.2.9: Result Page for Pneumonia

CONCLUSION

In conclusion, this project presents a robust framework for automated biomedical image analysis, demonstrating the potential of machine learning and web technologies in revolutionizing health care diagnostics. The system's accuracy and user-friendly interface make it a valuable tool for healthcare professionals, contributing to early disease detection and improved patient care. It's like a super-savvy assistant for doctors, helping them look at medical pictures and quickly figuring out if there's anything to be concerned about. The friendly website makes it a breeze for doctors to use, and behind the scenes, our smart system keeps everything tidy and safe.

Based on the accuracy obtained for each disease dataset in the AI-powered biomedical image analysis project, it is evident that the system demonstrates high performance in disease detection. The accuracy rates range from 89.76% for Breast Cancer to 94.89% for Pneumonia, showcasing the system's effectiveness across diverse medical conditions. These accuracy rates reflect the system's ability to accurately classify biomedical images and assist healthcare professionals in diagnosing diseases with a high level of confidence. Overall, the project's results underscore its potential to significantly impact healthcare by providing reliable and precise disease detection capabilities, thereby enhancing diagnostic accuracy and patient care outcomes.

Picture it like having a tech-savvy helper in a doctor's toolkit – someone who can process lots of information in the blink of an eye. The goal is simple: to speed up the process of finding health issues in images, making it simpler for doctors and ultimately improving the way we take care of people's health. It's a small but powerful step toward a future where technology works hand-in-hand with healthcare professionals to bring better and faster care to everyone.

FUTURE ENHANCEMENT

Looking ahead, there are numerous avenues for enhancing our project's capabilities. One potential enhancement involves exploring advanced deep learning architectures, such as recurrent neural networks (RNNs) or attention mechanisms, to improve the accuracy and sensitivity of disease detection. Additionally, incorporating multimodal data fusion techniques, which combine information from various sources such as medical images and clinical records, could provide a more comprehensive understanding of patients' health conditions. Real-time image analysis could be achieved through optimization techniques and parallel processing, enabling rapid diagnosis and treatment decisions. Integrating decision support systems and personalized medicine approaches would empower healthcare professionals to tailor treatments based on individual patient characteristics and medical histories. Furthermore, deploying our system on scalable and resilient cloud platforms could enhance its accessibility, scalability, and reliability. Finally, investing in user experience design and patient engagement features would ensure that our system remains intuitive, user-friendly, and inclusive, ultimately improving healthcare outcomes for patients worldwide.

REFERENCES

[1].A. S. Panayides et al., "AI in Medical Imaging Informatics: Current Challenges and Future Directions," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 7, pp. 1837-1857, July 2020, doi: 10.1109/JBHI.2020.2991043.

[2].R. Buettner, M. Bilo, N. Bay and T. Zubac, "A Systematic Literature Review of Medical Image Analysis Using Deep Learning," 2020 IEEE Symposium on Industrial Electronics & Applications (ISIEA), TBD, Malaysia, 2020, pp. 1-4, doi: 10.1109/ISIEA49364.2020.9188131.

[3].J. Ker, L. Wang, J. Rao and T. Lim, "Deep Learning Applications in Medical Image Analysis," in IEEE Access, vol. 6, pp. 9375-9389, 2018, doi: 10.1109/ACCESS.2017.2788044.

[4].Alvin Rajkomar, Sneha Lingam, and Andrew Taylor, "Machine learning in medical imaging: A review", Medical Imaging and Augmented Reality, 2018, doi: 10.1016/j.media.2017.06.005

[5].Ayushi Garg and Sanjay Kr Singh, "Biomedical Image Analysis with Deep Learning", Procedia Computer Science, 2018, doi: 10.1016/j.procs.2018.04.320

[6].Geert Litjens et al, "A survey on deep learning in medical image analysis", Medical Image Analysis, 2017, doi: 10.1016/j.media.2017.07.005

[7].Ayesha Heena, Najmuddin M. Maroof, Surbhi Bhatia, Rashmi Agarwal, Kanta Prasad,"Machine learning based biomedical image processing for echocardiographic images",July 2022,Multimedia Tools and ApplicationsVolume 82Issue 25Oct 2023pp 39601–39616https://doi.org/10.1007/s11042-022-13516-5

[8].Arslan Khalid, Arif Mehmood, Amerah Alabrah, Bader Fahad Alkhamees, Farhan Amin, Hussain AlSalman, Gyu Sang Choi,"Breast Cancer Detection and Prevention Using Machine Learning",October 2023,Diagnostics (Basel) 2023 Oct 2;13(19):3113. doi:10.3390/diagnostics13193113.

[9].Sonia Ben Hassen, Mohamed Neji, Zain Hussain, Amir Hussain, Adel M. Alimi, Mondher Frikha,"Deep learning methods for early detection of Alzheimer’s disease using structural MR images: a survey",June 2020,https://doi.org/10.1016/j.neucom.2024.12732.