International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Cogniscan-Deep Learning Based Brain Tumor Classification

N Sony[1], V Saranya[2], A Slycee Leyona[3], Tokala Sreeja[4]

[1]Assistant Professor; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Telangana, India

[2,3,4]B.Tech Students; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Telangana, India

Mail Id; Nagulapally.Sony@slv-edu.inN[1], saranyabv20@gmail.com[2], slycee310@gmail.com[3], sreejareddy7474@gmail.com[4]

Abstract

CogniScan is a web-based application that employs deep learning techniques to classify brain tumors from magnetic resonance imaging (MRI) scans in an efficient and structured manner. Conventional tumor identification depends heavily on manual interpretation by medical professionals, which is time-consuming and may introduce inconsistencies, particularly when processing large datasets. The proposed system automates this process by allowing users to upload MRI images and receive accurate classification results.The application supports multiple categories, including glioma, meningioma, pituitary tumor, and no tumor, enabling comprehensive diagnostic assistance. A custom Convolutional Neural Network (CNN) model is used for feature extraction and classification. To improve performance, preprocessing steps such as resizing and normalization are applied to ensure consistency across input images.The system incorporates secure user authentication, allowing users to register and log in before accessing prediction features. A user-friendly interface built with Streamlit provides smooth interaction, enabling image uploads and visualization of results along with confidence scores. SQLite is used for secure storage of user credentials and authentication management. Additional features include efficient data handling, rapid prediction generation, and reliable output delivery.Overall, the proposed system offers a scalable and accessible solution for automated medical image analysis, reducing manual effort and supporting healthcare professionals in diagnostic decision-making.

INTRODUCTION

CogniScan is designed to automate the classification of brain tumors using MRI images through deep learning techniques. Traditional tumor identification relies on manual examination by medical specialists, which is both time-intensive and susceptible to variability. The proposed platform simplifies this workflow by allowing users to upload MRI scans and obtain classification results automatically.

The application categorizes images into glioma, meningioma, pituitary tumor, or no tumor. Image preprocessing is performed to enhance prediction accuracy and maintain consistency. The system also provides confidence scores for each prediction, enabling better interpretability. A secure login mechanism and efficient data handling ensure reliable system performance.

Scope

The scope of CogniScan is to develop a web-based platform that enables automated brain tumor classification using deep learning models. Users can register, log in, upload MRI images, and receive classification results in real time. The system supports multiple tumor categories and applies preprocessing to improve model accuracy. It ensures secure authentication, efficient processing, and user-friendly interaction. The platform aims to assist medical professionals by providing quick and reliable diagnostic support.

Existing System

Current approaches for brain tumor detection mainly depend on the manual interpretation of MRI images by healthcare professionals. This method requires significant expertise and is often time-consuming, particularly when dealing with large volumes of medical imaging data. The accuracy of diagnosis may vary depending on the experience and skill level of the practitioner, and identifying tumors at an early stage can be challenging. Furthermore, the absence of automation in traditional workflows reduces scalability and makes the process inefficient for large-scale screening and rapid analysis.

Limitations of Existing System

The traditional system suffers from several drawbacks that affect its efficiency and reliability. Manual MRI analysis demands considerable time and active involvement of medical experts, which limits the ability to process large datasets. The strong dependency on specialists reduces scalability and increases workload. There is also a higher chance of diagnostic errors, especially in complex or early-stage cases. Additionally, the lack of automated detection mechanisms prevents consistent and rapid analysis. The system is also inefficient in handling large volumes of medical imaging data, leading to delays in diagnosis.

Proposed System

The CogniScan platform introduces an automated solution for brain tumor classification using deep learning techniques. The system enables secure user registration and authentication to ensure safe access. Users can upload MRI images, which are processed automatically by the deep learning model to classify tumor types. The system provides prediction results along with confidence scores for better interpretability. Efficient image preprocessing and data handling improve model performance and reliability. Additionally, the platform includes an interactive and user-friendly interface that simplifies user interaction and ensures smooth operation.

REQUIREMENT ANALYSIS

Functional Requirements

The functional requirements of the system are divided into user and system modules. The user module allows individuals to register and log in securely, upload MRI images, view tumor classification results, and access prediction confidence scores along with output reports. The system module handles image preprocessing, execution of the deep learning model, generation of classification results, display of outputs, and overall data processing operations. These functions collectively ensure smooth and accurate system performance.

Non-Functional Requirements

The non-functional requirements focus on performance, scalability, usability, reliability, security, and compatibility. The system is designed to provide fast responses during image upload and prediction generation. It should support multiple users and handle large datasets without performance degradation. The interface must remain simple and intuitive for ease of use. Reliable performance with consistent prediction accuracy is essential. Security measures ensure protected user authentication and safe data storage. Additionally, the application should be compatible across different devices and web browsers to maximize accessibility.

Software Requirements

The software requirements define the tools used for developing and running the system. The application operates on Windows 10 or Windows 11. Python is used as the programming language, while Streamlit serves as the framework for building the web interface. Deep learning libraries such as TensorFlow and Keras are used for model development. Image processing is handled using OpenCV and NumPy. SQLite is utilized as the database for storing user credentials and managing authentication.

Hardware Requirements

The hardware requirements specify the physical resources necessary for efficient system execution. The system requires a processor equivalent to Intel i5 or higher for smooth computation. A minimum of 8 GB RAM is recommended to handle image processing tasks effectively. Additionally, at least 512 GB of storage is required for system files, datasets, and application deployment.

Software Process Model

A software process model provides a structured approach to software development. Several models such as Waterfall, V-Model, Incremental, Spiral, and Agile can be adopted depending on project needs. For the CogniScan system, the Waterfall model is selected due to its sequential and systematic development approach. This model ensures clear requirement analysis, proper design, structured implementation, and well-documented testing phases, making it suitable for projects with well-defined objectives.

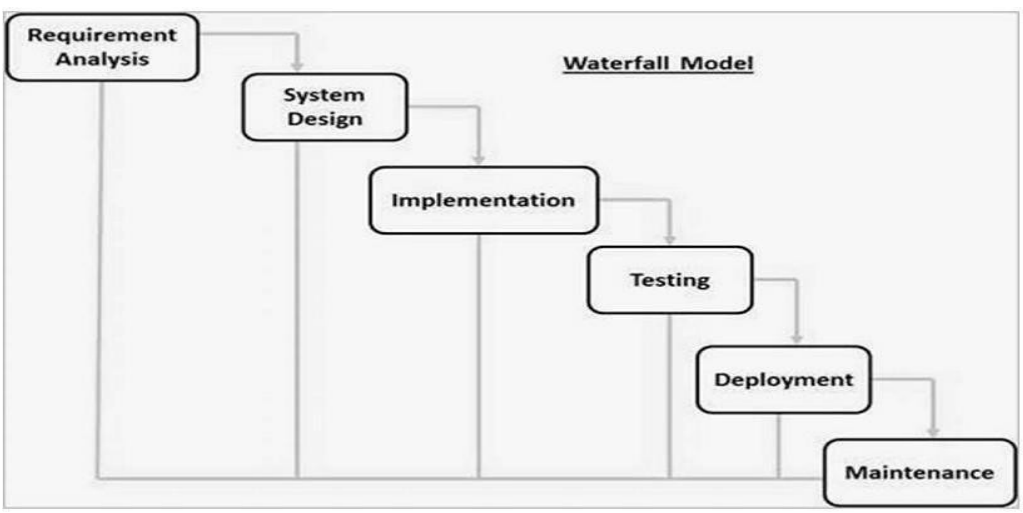


Fig.1 Waterfall Model

DESIGN

The design phase focuses on planning system components and defining workflows to achieve efficient brain tumor classification. It includes steps such as data preprocessing, feature extraction, model selection, training strategy, and evaluation. The system is designed with a user-friendly interface to ensure smooth interaction. Deployment planning is also considered to integrate all modules effectively and maintain reliable performance.

Architecture

The system architecture illustrates the components involved and the flow of data between them. It consists of software architecture and technical architecture, both of which define how the system operates and how technologies interact.

Software Architecture

The software architecture defines the workflow of the application. It includes modules for user interaction, image preprocessing, feature extraction, and classification. These components operate sequentially to process MRI images and generate predictions. This structured approach enhances maintainability, scalability, and reliability while ensuring efficient data processing.

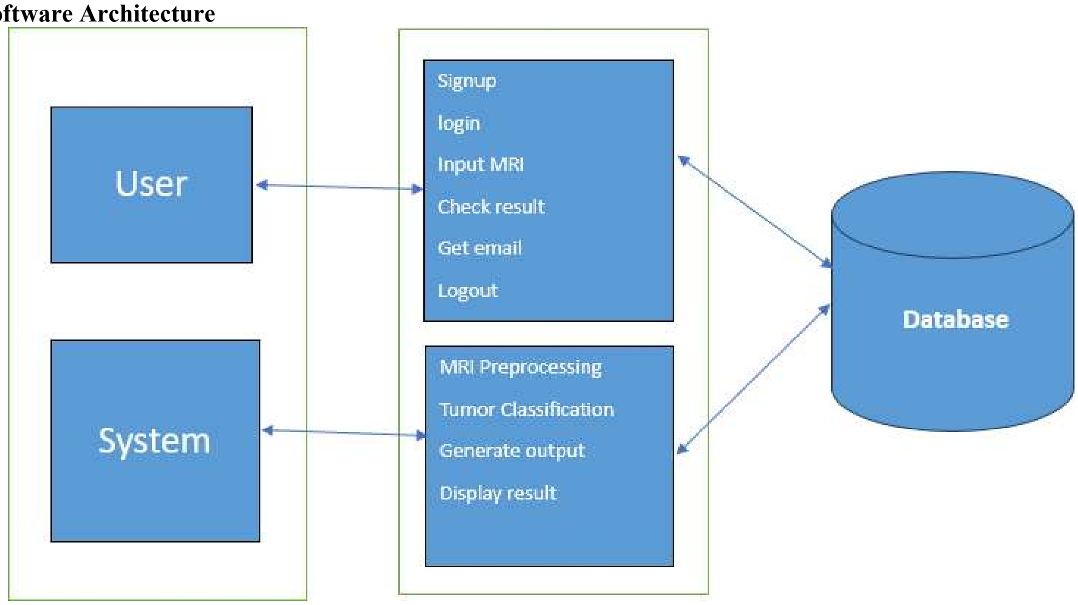


Fig.2 Software Architecture

Technical Architecture

The technical architecture describes the technologies used in system implementation. The frontend is developed using Streamlit along with HTML and CSS to provide an interactive user interface. Python is used in the backend to handle processing logic and model execution. SQLite is used to manage user authentication data securely. The communication between frontend, backend, and database ensures smooth data flow and efficient overall system operation.

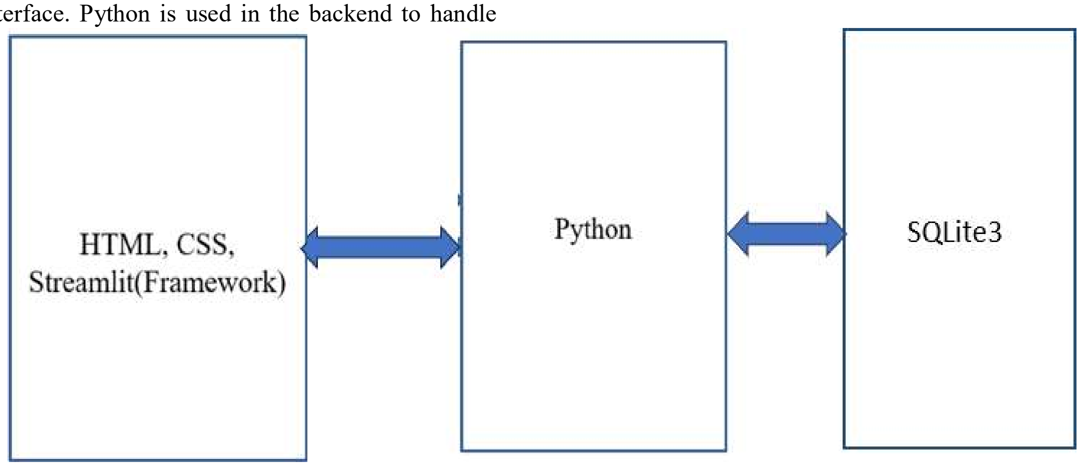


Fig.3 Technical Architecture

IMPLEMENTATION

Technologies

The CogniScan system is implemented using a combination of modern web and deep learning technologies to ensure efficient processing and user interaction. These technologies support model execution, user authentication, and result visualization in a structured manner.

Frontend – Streamlit

Streamlit is used to build the frontend of the CogniScan application. It provides an interactive and lightweight interface that allows users to upload

MRI images and view classification outputs. Streamlit enables rapid development of web.

Backend – Python

Python serves as the backend technology for handling system operations. It manages image preprocessing, deep learning model execution, and prediction generation. User inputs are processed in Python, where MRI images undergo normalization and resizing before being passed to the trained convolutional neural network. Python is well suited for this application due to its readability, flexibility, and extensive support for machine learning libraries.

Database – SQLite

SQLite is used as the database for storing user credentials and managing authentication. It provides a lightweight and embedded data storage solution without requiring complex configuration. SQLite supports secure storage, efficient retrieval, and structured management of user information, making it suitable for small to medium-scale applications.

Algorithm / Pseudocode

The implementation begins with database initialization, followed by construction of the convolutional neural network architecture. The database stores user details for authentication, while the CNN model performs feature extraction and classification.

The system first establishes a connection to the SQLite database and creates a table for storing user information. Each record contains a unique identifier, username, user ID, and password. After committing database changes, the deep learning model is initialized using a sequential architecture.

The convolutional neural network consists of multiple convolution blocks. Each block includes convolution layers for feature extraction, pooling layers for dimensionality reduction, and dropout layers to prevent overfitting. As the network depth increases, the number of filters also increases to capture complex spatial features. After convolutional operations, a flatten layer converts feature maps into a one-dimensional vector. Fully connected dense layers are then used to learn high-level representations. Finally, a softmax output layer classifies the input MRI image into four categories: glioma, meningioma, pituitary tumor, and no tumor.

TESTING

Software testing is an essential phase in the development lifecycle that ensures the system meets functional requirements and operates without defects. It verifies correctness, reliability, and performance of the application. In modern digital environments, software systems play a critical role in data processing and decision-making. Any malfunction can lead to inaccurate results and reduced trust. Therefore, thorough testing is necessary to deliver a robust product.

In the CogniScan system, testing ensures that MRI image upload, preprocessing, model prediction, and output display function correctly. Proper testing improves system quality, enhances security, and ensures user satisfaction. The major objectives of testing include cost efficiency, improved product quality, enhanced security, and increased reliability.

Dimensions of Testing

Testing is performed across multiple dimensions to evaluate different aspects of the system. These include testing various layers of the application such as user interface, processing logic, and database operations. Testing is also conducted at different scales, including unit, module, integration, and scenario-based testing. Furthermore, different testing types such as functional, performance, and security testing are considered. Both manual and automated methodologies are applied to ensure comprehensive coverage.

Stages of Testing

Unit testing focuses on validating individual components of the application. Each module, such as image preprocessing or login validation, is tested independently to verify correct functionality. White-box testing techniques are commonly used in this phase to ensure proper execution of code logic.

Integration Testing

Integration testing evaluates the interaction between different modules. In the CogniScan system, modules such as image upload, preprocessing, and prediction are combined and tested together. This stage helps identify interface issues and ensures smooth communication between components.

System Testing

System testing validates the entire application as a complete unit. The objective is to confirm that the system satisfies functional and technical requirements. Testing is performed in an environment that simulates real-world usage. This stage verifies performance, reliability, and overall system behavior.

Types of Testing

Black Box Testing

Black box testing evaluates system functionality without examining internal code structure. Testers provide inputs and verify outputs based on expected behavior. This method is suitable for validating user interactions such as login, image upload, and prediction display.

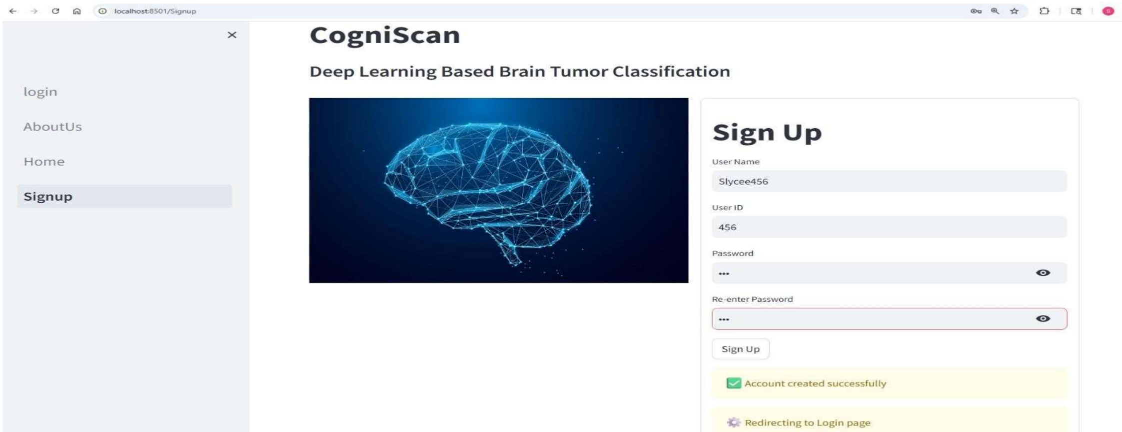
White Box Testing

White box testing analyzes the internal logic of the application. It is typically performed at the unit level and requires programming knowledge. This method ensures correct execution paths and code coverage. Techniques used include statement coverage, branch coverage, and path coverage.

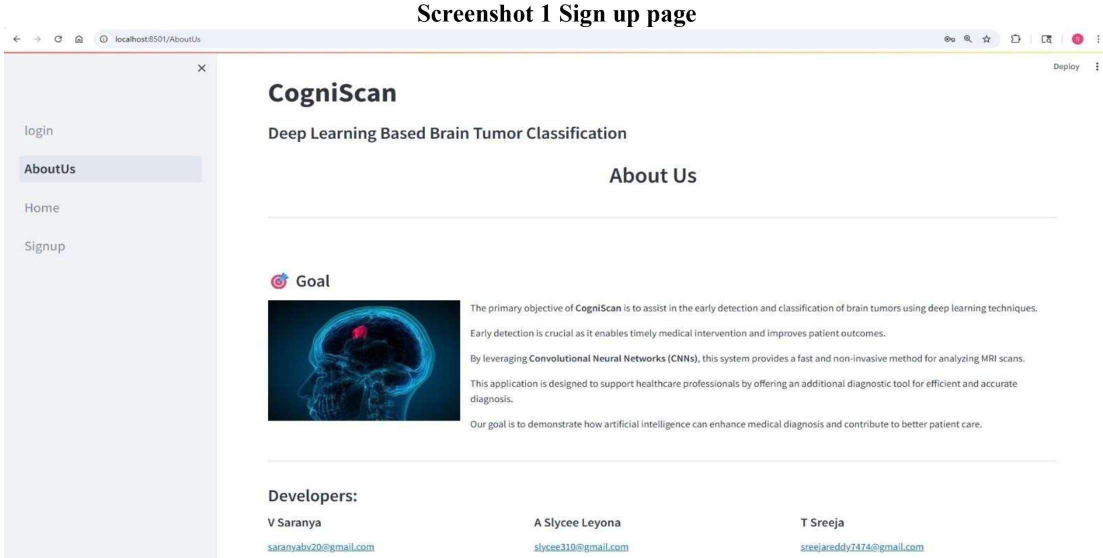
Test Cases

Test cases were developed to validate various functionalities of the CogniScan system. These include user authentication, image upload, preprocessing, tumor classification, and result display. The system successfully handled both valid and invalid inputs, generating appropriate outputs in each case. All tested scenarios produced expected results, indicating reliable performance.

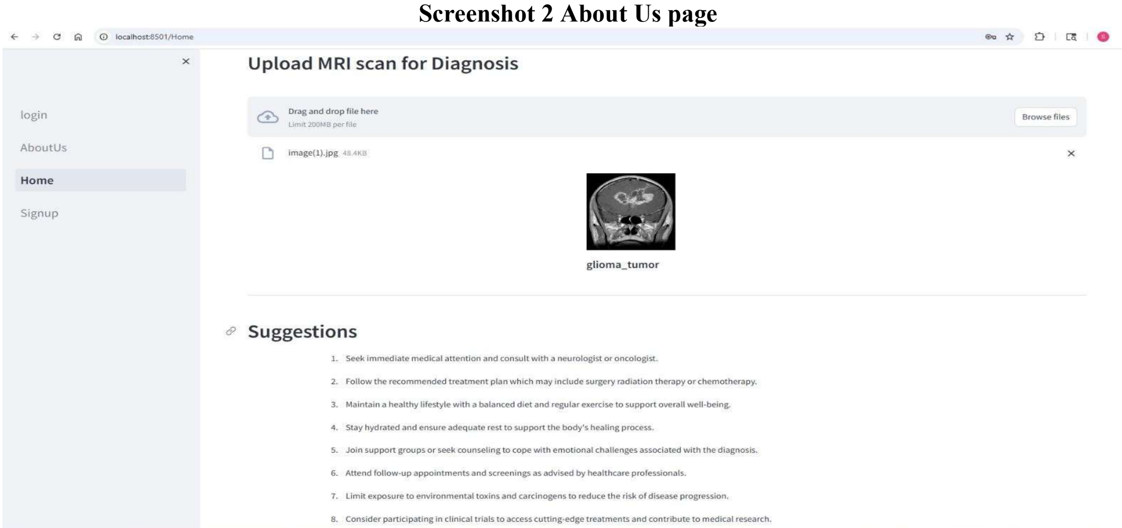
Screenshots



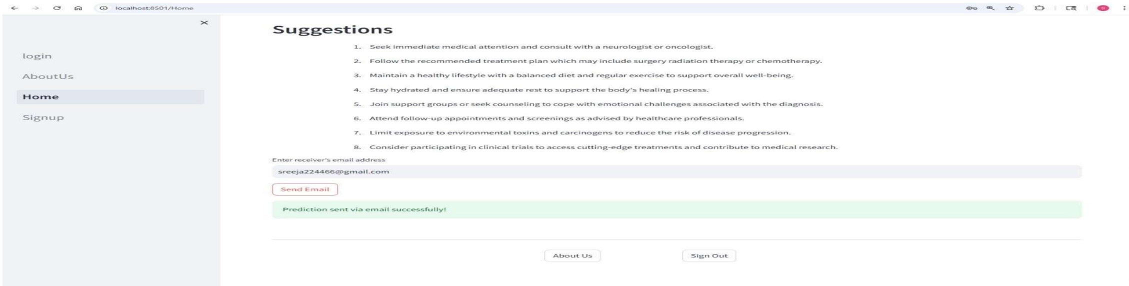
Screenshot 1 Sign up page



Screenshot 2 About Us page



Screenshot 3 MRI Scan upload and Tumor Detection



Screenshot 4 Email sent confirmation page

Screenshot 5 Email of health report

Conclusion

The CogniScan system simplifies the process of brain tumor classification through a deep learning-based digital platform. It allows users to upload MRI images and obtain accurate classification results efficiently. The system automates image processing and prediction, reducing manual effort and improving diagnostic support. It ensures reliable performance through secure user authentication and efficient data handling. Features such as quick prediction and user-friendly interface enhance usability. Overall, the system provides an effective and accurate solution for brain tumor detection using advanced machine learning techniques.

Future Scope

The CogniScan system can be enhanced by developing a mobile application for easier access and wider usability. The system can be extended to detect additional brain conditions beyond the current tumor categories, making it more comprehensive. Advanced features such as real- time result notifications and improved user interaction can be added. The application can be integrated with hospital systems to assist doctors in diagnosis and decision-making. The system can also be deployed on cloud platforms to support multiple users and improve accessibility. Security and performance can be further enhanced to ensure reliable and efficient operation in real-world scenarios.

REFERENCES

1) Karim et al.,"Developments in Brain Tumor Segmentation Using MRI: Deep Learning Insights and Future Perspectives,"IEEE Access, 2024.
2) Fahim et al.,"Brain Tumor Detection, Classification and Segmentation Using Deep Learning Models,"Array Journal (Elsevier), 2025.
3) U. Baid et al.,"The RSNA-ASNR-MICCAI BraTS Challenge on Brain Tumor Segmentation,"Medical Image Analysis, 2021.
4) Majib et al., VGG-SCNet: A VGG-based Deep Learning Framework for Brain Tumor Detection on MRI Images, IEEE Access, 2021.
5) Ahmad & Choudhury, Performance of Deep Transfer Learning Networks for Brain Tumor Detection, IEEE Access, 2022.
6) Kumar & Singh, A Hybrid Deep Learning Model for Brain Tumor Detection, IEEE Access, vol. 10, 2022.
7) Litjens et al., A Survey on Deep Learning in Medical Image Analysis, Medical Image Analysis, 2017.