Image Based Brain Tumor Detection Using Deep learning

Project Team

Abdul Mohaiman	20P-0580
Haris Ali	20P-0142
Nazim Sulman	20P-0639

Session 2020-2024

Supervised by

Dr.Hafeez Anwar

Department of Computer Science

National University of Computer and Emerging Sciences
Peshawar, Pakistan

June, 2022

Student's Declaration

We declare that this project titled "*Image Based Brain Tumor Detection Using Deep learning*", submitted as requirement for the award of degree of Bachelors in Computer Science, does not contain any material previously submitted for a degree in any university; and that to the best of our knowledge, it does not contain any materials previously published or written by another person except where due reference is made in the text.

We understand that the management of Department of Computer Science, National University of Computer and Emerging Sciences, has a zero tolerance policy towards plagiarism. Therefore, We, as authors of the above-mentioned thesis, solemnly declare that no portion of our thesis has been plagiarized and any material used in the thesis from other sources is properly referenced.

We further understand that if we are found guilty of any form of plagiarism in the thesis work even after graduation, the University reserves the right to revoke our BS degree.

Abdul Mohaiman Signature: ____________________

Haris Ali Signature: ____________________

Nazim Sulman Signature: ____________________

____

Verified by Plagiarism Cell Officer
Dated:

Certificate of Approval

The Department of Computer Science, National University of Computer and Emerging Sciences, accepts this thesis titled *Image Based Brain Tumor Detection Using Deep learning*, submitted by Abdul Mohaiman (20P-0580), Haris Ali (20P-0142), and Nazim Sulman (20P-0639), in its current form, and it is satisfying the dissertation requirements for the award of Bachelors Degree in Computer Science.

Supervisor

Dr.Hafeez Anwar Signature: ________________

____

Name of FYP Coordinator

FYP Coordinator
National University of Computer and Emerging Sciences, Peshawar

____

Dr. Head of Computer Science Department

HoD of Department of Computer Science
National University of Computer and Emerging Sciences

Acknowledgements

Your acknowledgments here

Abdul Mohaiman

Haris Ali

Nazim Sulman

Abstract

The global burden of brain tumors presents a significant health challenge, affecting millions of individuals annually. Timely and accurate detection of brain tumors is crucial for optimal treatment planning and patient care. Traditional diagnostic methods such as MRI and CT scans, while effective, can be time-consuming and heavily reliant on the expertise of radiologists. There is a pressing need for improved diagnostic techniques that enhance efficiency in brain tumor detection.

Advancements in medical imaging, including scaling, resizing, and enhancing blurred MRI images, offer potential for improved accuracy and efficiency. By leveraging deep learning and artificial intelligence, these techniques can facilitate better detection of brain tumors, leading to earlier interventions and more personalized treatment strategies. Addressing these challenges can significantly improve outcomes for patients with brain tumors and advance the field of medical diagnostics.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Brain tumors are one of the most severe medical conditions affecting the human population, with high morbidity and mortality rates. They can be classified into benign (non-cancerous) and malignant (cancerous) tumors, both of which pose significant health challenges. Early diagnosis and accurate classification of these tumors are essential for effective treatment and improved patient survival rates. Medical imaging, particularly Magnetic Resonance Imaging (MRI), has become a pivotal tool in the diagnosis and monitoring of brain tumors. MRI provides high-resolution images of the brain, allowing medical professionals to visualize abnormalities in brain tissues non-invasively. However, the manual interpretation of MRI scans is a complex and time-consuming process that requires the expertise of radiologists and neurologists, and even then, it is prone to errors.

To address these challenges, the integration of computational methods, such as deep learning and machine learning, into medical diagnostics has gained significant attention. These technologies offer a promising solution by automating the detection and classification of brain tumors with high accuracy. Deep learning, a subset of machine learning, leverages neural network architectures, specifically convolutional neural networks (CNNs), to analyze and extract relevant features from MRI images. The ability of these networks to automatically learn from large datasets and improve their accuracy through iterative training has made them valuable tools for image classification tasks, including medical imaging applications.

The objective of this project is to develop an automated system for brain tumor detection and classification using MRI images by employing state-of-the-art deep learning and machine learning algorithms. This system aims to reduce the need for manual image interpretation, thereby increasing diagnostic speed and accuracy. The focus is on building a robust model that can differentiate between various types of brain tumors and normal brain tissue, providing clinicians with a reliable tool for early and precise diagnosis.

In this project, a comprehensive dataset of MRI brain images will be collected, including images of different types of brain tumors such as gliomas, meningiomas, and pituitary tumors. The images will be pre-processed to enhance their quality and ensure consistency across the dataset. This step is crucial, as image quality and uniformity directly impact the performance of the model. The deep learning model, primarily based on CNN architecture, will be trained on this dataset, leveraging its capability to detect patterns and features indicative of tumors.

Alongside deep learning methods, traditional machine learning algorithms such as Support Vector Machines (SVM), Random Forests, and K-Nearest Neighbors (KNN) will also be explored. These algorithms are often used in medical imaging for their simplicity and effectiveness in classification tasks. By comparing the performance of these approaches, the project aims to identify the most efficient and accurate method for brain tumor detection.

The anticipated outcome of this research is a reliable and efficient system capable of assisting healthcare professionals in diagnosing brain tumors with minimal human intervention. This system has the potential to revolutionize clinical diagnostics by providing a quick, non-invasive, and accurate solution for brain tumor detection, ultimately contributing to better patient outcomes. Moreover, the project aims to bridge the gap between advanced computational technologies and practical healthcare applications, demonstrating the effectiveness of combining medical imaging with machine learning and deep learning approaches.

Chapter 2

Review of Literature

]

Paper 1:

Brain Tumor Identification and Classification of MRI images using deeplearning techniques

Introduction:

The detection, segmentation, and extraction from Magnetic Resonance Imaging (MRI) images of contaminated tumor areas are significant concerns; however, a repetitive and extensive task executed by radiologists or clinical experts relies on their expertise. Image processing concepts can imagine the various anatomical structure of the human organ. Detection of human brain abnormal structures by basic imaging techniques is challenging. In this paper, a Fully Automatic Heterogeneous Segmentation using Support Vector Machine (FAHS-SVM) has been proposed for brain tumor segmentation based on deep learning techniques. The present work proposes the separation of the whole cerebral venous system into MRI imaging with the addition of a new, fully automatic algorithm based on structural, morphological, and relaxometry details. The segmenting function is distinguished by a high level of uniformity between anatomy and the neighboring brain tissue. ELM is a type of learning algorithm consisting of one or more layers of hidden nodes. Such networks are used in various areas, including regression and classification. In brain MRI images, the probabilistic neural network classification system has been utilized for training and checking the accuracy of tumor detection in images. The numerical results show almost 98.51in detecting abnormal and normal tissue from brain Magnetic Resonance images that demonstrate the efficiency of the system suggested.

Conclusion: This paper presents a Fully Automatic Heterogeneous Segmentation using Support Vector Machine (FAHS-SVM) for brain tumor identification and segmentation. The accuracy of our automated approach is similar to the values for manual segmentation inter-observer variability. To identify tumor regions by combining intrinsic image structure hierarchy and statistical classification information. The tumor areas described are spatially small and consistent concerning image content and provide an appropriate and robust guide for the consequent segmentation. The proposed method can achieve promising tumor segmentation in conjunction with a semi-supervised approach under a local and globalized accuracy system, as is shown by experiments focused on multi-parametric Magnetic Resonance images. Our experimental results indicate that the method proposed will help to identify the exact location of the brain tumor accurately and quickly. The proposed method is,therefore, critical for MR imagery brain tumor detection. The experimental results showed 98.51 of the accuracy of theproposed technology in the detection of abnormal and normaltissues in Magnetic Resonance images. The findings lead to theend that the suggested approach is sufficient for the inclusion ofprimary diagnostic and radiologist or clinical experts in supportof clinical decision system.

Paper 2: A. smith brain segment classification on anomaly detection using machine learning techniques,IEEE Access 2022:

Introduction:

Most cells in the body grow and then divide in an organized manner to create new cells to maintain the body's health and functionality. Figure 1 shows the pictorial representation of brain lesion. When cells can no longer regulate their growth, they divide too often and randomly; hence every day, our immune system destroys a cell that, had it survived, would have developed into malignant cells [1, 2]. A tumor is generally a mass of tissue made of extra cells. Cell proliferation that is aberrant and out ofcontrol causes brain tumors. According to estimates, 16,500 new lesions in brain were diagnosed in the U.S. in 2000, accounting for 1.4 percent of all cancer cases, 2.4 percent of cancer fatalities, and 20–25 percent of pediatric malignancies. In the end, brain tumors are thought to cause 13,000 annual fatalities. Approaches for improved Segmentation are explored and employed in processing biomedical images [2].Deep Learning technology is ideal for addressing detections in the field of medicine. The possibility of Deep Learning technology in the Medical Field has been a topic of investment. Numerous evaluations that provide a summary of the present state of affairs and a road map for future study have been published. Inspired by these findings, we have analyzed several models of brain tumor detection based on deep learning in the suggested survey

Conclusion:

Appropriate Segmentation of M.R. images is crucial for improved diagnosis and therapy in patients with brain tumors. Accurate diagnosis,planning, and treatment require information from multiple slices. With the abundance of information available, computer processing is necessaryfor decision-making. Researchers prioritize improving the info obtained from collected slice images and optimizing the segmentation process overcomputation speed. This publication highlights substantial recent research efforts in brain tumor identification and Segmentation. Automation ofbrain tumor identification and Segmentation from brain M.R. images is a highly active area of research. Significant efforts have been made overmany years, as evidenced in our literature review. However, few medical community has accepted an automated procedure.In this publication, we tried to discuss a few of the substantial recent research efforts on brain tumor identification and Segmentation.Automating brain tumor identification and Segmentation from brain M.R. images is one of the most active study topics. Our literature examinationshows significant research has been done in this area for many years. However, the medical community currently approves no automated procedure

Paper3:

Deep learning based semantic segmentaion for brain tumor detection in MRI images,IEEE transactions on Medical Imaging,2022

Introduction:

Brain tumorsare a big problem in medical testing because they need to be found quickly and correctly so that they can be treated effectively. This opening gives a full background. It starts with a short look at brain tumors, their different symptoms, and how important it is to find them early. Brain tumors are made up of cells that grow in a way that isn't normal in the brain. They can look like a lot of different things[1]. They can show up in different parts of the brain, affecting movement skills, cognitive function, and the health of the brain as a whole. Understanding how different they are is important for coming up with good diagnosis methods. Early [2]discovery becomes very important because it directly leads to better treatment results and a better outlook for the patient. Magnesium-based magnetic resonance imaging (MRI) is the best way to find brain cancer. MRI is very important in neuroimaging because it can show detailed pictures of soft brain cells without hurting the person. The main job of MRI is to findand describe brain tumors, and this part goes into detail about that. It looks into how MRI, with its better contrast in soft tissues, can help find abnormalities and tumors that other imaging methods might miss. Deep learning is being looked into as a possible way to find brain tumors because of the problems with current diagnosis methods[7]. Traditional methods often involve reading pictures by hand, which can be subjective and cause small problems to be missed. Because brain tumor anatomy is so complicated, we need a smarter and more automatic way to analyze it. This drives them because they know that deep learning, which can find complex patterns and features in medical pictures, could greatly improve the precision and speed of brain tumor detection.Brain imaging is very important for diagnosing and treating brain-related illnesses, and brain tumor spotting is one of the most important parts of this field. Manually looking at medical pictures is what traditional methods do, which takes a lot of time and can lead to mistakes [1]. Deep learning is changing quickly, especially in computer vision. This opens up a huge chance for brain tumor detection to be done automatically and accurately [2]. Convolutional Neural Networks (CNNs), one type of deep learningmethod, has become one of the most useful and effective tools for picture analysis tasks like object recognition and segmentation [3]. Researchers are looking into how CNNs can help find and classify brain cancers from Magnetic Resonance Imaging (MRI) data. They are using deep learning to improve the accuracy of diagnosis and treatment [4]. The main goal is to make CNNs that are good at finding and grouping brain tumors, which shows how flexible deep 1Department of ECE, Bharath Institute of Higher

Education and Research Tamil Nadu, India and AISSMS Institute of Information and Technology, Pune,India.*jadhavrn@gmail.com2Department of ECE, Bharath Institute of Higher Education and Research, Tamil Nadu, Indiasudhagar.ece@bharathuniv.ac.in International Journal of Intelligent Systems and Applications in EngineeringIJISAE, 2024, 12(13s), 586–602l587learning can be in medical imaging [5]. One great thing about deep learning is that it can learn complex, structured features on its own from raw data, so you don't have to use rule-based methods or make features by hand [6]. Because they are built to finds patial links and local patterns in pictures, convolutional neural networks work really well for medical image processing jobs. Because of this, deep learning has become an important tool for analyzing medical pictures, which helps doctors diagnose diseases more accurately and find problems in imaging data. Deep learning can also be used to make medical decisions automatically, which could make the work of doctors easier [7]. Using deep learning to its full potential makes it possible to get more accurate diagnoses than with traditional methods. Automating medical findings not only makes things run more smoothly, but it also has the potential to make healthcare better generally. In conclusion, using deep learning to find brain tumors is a big step forward that could lead to faster and more accurate medical picture analysis and identification.People are moving from traditional[8]methods to deep learning in medical picture analysis because they are becoming more aware of the problems with the old methods. Even though traditional methods are useful, they have trouble with the complexity and variability of brain tumor images. Deep learning, which can automatically pull features and recognize patterns, is a game-changer that can help us get past these problems. This part goes into more detail about how the built-in abilities of deep learning models work well with the complex needs of brain tumor research. Setting the goals of this poll is important for describing its reach and expected contributions. The goal of the study is to look at all the different ways that deep learning can be used to find brain tumors using MRI pictures. Its goal is to bring together what is already known, point out current trends, and find study holes. The poll wants to be a useful tool for academics, therapists, and other people interested in the area where deep learning and MRI meet. The main goals are to explain the pros and cons of different deep learning models, evaluate the datasets that were used, and give information that will help guide future study in this important area.The paper objective is given as:•To discuss and review the different dataset available for the MRI images of brain tumor patients, get MRI images of patients and put them into a data set. •To explore and provide comprehensive review the latest techniques and methods to detect the brain tumors using MRI images datasets, with maximum accuracy of deep learning model. Also to provide the review based the different deep learning model with its evaluation parameter with the pitfall and advantages of it2.BackgroundA. Traditional Methods for Brain Tumor DetectionBrain tumors have been found in the past using complicated methods that include picture preparation, feature extraction, and classification techniques. This section goes into great depth about each of these parts.1.Techniques for Image preparation: The traditional way of doing[9]things starts with image preparation to make the MRI data better and more useful. During this step, a set of tasks are done to reduce noise, make sure that levels are all the same, and improve the general quality of the pictures. Noise reduction with filtering, intensity normalization, and spatial normalization are all common preparation steps. Noise reduction is especially important for MRI pictures, which can be different because of things like different tools and the patient moving around during the scan. By using these preparation methods, the information can be used for later steps of research in a more reliable and consistent way.2.Feature Extraction and Selection: After preparation, the focus moves to feature extraction and selection, which is an important step in standard methods for finding brain tumors. Features are unique things or patterns in the images that can help tell the difference between healthy tissues and tissues that have been damaged by a tumor. Handcrafted features are often used in traditional methods, which involve figuring out things like shape, color, and strength changes. While this hand picking works in some situations, it's not very good at catching complex, non-linear patterns. Also, relying on traits that have already been decided upon could mean missing small but important signs of a growth. Even with these problems, standard methods have been shown to be effective at pulling important traits that can then be used for classification.

Conclusion:

The deep learning techniques to find brain tumors in MRI pictures is a huge step forward in medical imaging. Researchers have made a lot of progress in automating and improving the accuracy of tumor spotting using tools like Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, YOLOv7, ResNet-50, VGG16, and Inception V3. These models are very good at recording spatial hierarchies, picking out complex patterns, and changing to different datasets, which makes them useful for doctors. But the field faces problems that need to be looked at. The ability to understand deep learning models is still a problem. This is why explainable AI methods are needed to make clear decisions in important medical situations. There isn't a lot of labeled data, which makes it harder for more people to use it. This shows how important it is for people to work together to collect large datasets and look into other ways to learn. The best way to find brain tumors in the future is to combine deep learning with different imaging methods, such as functional MRI, diffusion tensor imaging, and positron emission tomography. Multi-modal methods offer a more complete picture of how tumors work, which will improve both sensitivity and precision. Also, new trends show thatmedical imaging is moving toward AI that can be explained. Making clear models is important for making sure that clinical integration works well and that models are always getting better and being confirmed. The field of deep learning for brain tumor identification is likely to become more popular and used in clinical practice as it gets better at combining interpretability, data availability, and multi-modal integration

Paper 4:

Accurate brain tumor detection using deep convolutional neural network Khan, Md. Saikat Islam et al. Computational and Structural Biotechnology Journal Volume 20, 4733 - 4745

Introduction:

A brain tumor is one of the deadliest illnesses which occurs due to the sudden and unregulated brain tissue growth inside the skull. It can be either benign or malignant. Malignant tumors can expand quickly and disperse across the surrounding brain tissue, whereas benign tumors tend to grow slowly. However, benign tumors can also be dangerous as their proliferation may affect surrounding brain tissues. About 70 of the tumors are benign, and 30are malignant [1]. So far, more than 120 different brain tumors including meningioma, glioma, and pituitary as the most popular ones have been detected and identified.

Among these three, meningioma tumors are perhaps the most prominent primary brain tumor in the meninges and affect the brain and spinal cord [2]. On the other hand, glioma tumors grow from glial cells called astrocytes. The most prominent tumor of glioma is an astrocytoma, a low-risk tumor that suggests slow development. However, high-risk glioma is one of the most severe brain tumors. Pituitary is another type of tumor that is due to excessive growth of brain cells in the pituitary gland of the brain. Therefore, early diagnosis of a brain tumor is essential due to its deadly aspect. According to the International Association of Cancer Registries (IARC), there are more than 28,000 people diagnosed with brain tumors every year just in India in which more than 24,000 people die [3]. Another study reported that there are approximately 5,250 deaths recorded annually in the United Kingdom due to brain tumors [4]. In the United States, the impact of brain tumors is even more significant than in other countries. Just in 2019, about 86,970 cases of benign and malignant brain tumors are diagnosed [5]. The radiologist uses different experimental procedures for diagnosing brain tumors, including biopsy, Cerebrospinal fluid (CSF) analysis, and X-ray analysis. In the biopsy procedure, a small fragment of tissue is removed by surgery. The radiologist then determines whether the tissue holds a tumor or not. However, the biopsy process introduces many risks including inflammation and severe bleeding. It also has just 49.1 accuracy [6]. CSF is a colorless fluid that illustrates inside the brain. The radiologist tests the liquid to detect a brain tumor. However, similar to biopsy, it introduces many risks including bleeding from the incision site to the bloodstream and perhaps an allergic reaction after the treatment [7]. Similarly, using X-rays on the skull can lead to an increase in the risk of cancer due to the radiation. Nowadays, image modalities are becoming more popular for radiologists since they are more accurate and introduce much less risk to patients. There are different methods for capturing medical imaging data including radiography, magnetic reasoning imaging (MRI), tomography, and echocardiography. Among them, MRI is the most prominent as it provides higher resolution images without any radiation. MRI is a non-invasive procedure that provides the radiologist with useful knowledge of medical image data to diagnose brain abnormalities [8,9]. On the other hand, the Computer-Aided Diagnosis (CAD) method is designed for detecting brain tumors in the early stages without any human intervention. CAD systems can produce diagnostic reports based on MRI images and offer guidance to the radiologist [10]. The CAD process has improved dramatically using machine learning (ML) and deep learning (DL) applications in the medical imaging field [11–13]. Such techniques lead to better accuracy in terms of detecting brain tumors in the CAD system. Machine learning techniques are based on feature extraction, feature selection, and classification approaches. Different feature extraction techniques, including thresholding-based, clustering-based, contour-based, and texture-based are used for segmenting the tumor region from the human skull [14]. Such techniques extract the features from the MRI images where the important features are selected through the feature selection process. Extracting features with significant discriminatory information lead to achieving high accuracy [15]. However, using features extraction, it is possible to discard important information from the original image [16]. On the other hand, DL methods address this issue by using the original image as input[17]. In other words, they do not require handcrafted features for classification purposes. Among DL models, Convolutional Neural Network (CNN) provides[18] different convolution layers which will automatically extract features from the images[19]. CNN performed well when working with a large dataset which is not always easy to obtain in the medical imaging field [20]. One method to address this issue is to use transfer learning. In transfer learning[21], a model that has been previously trained with another large dataset related to another domain is used for the classification purpose[22]. Such knowledge helps the model to achieve high accuracy on a small dataset [23]. In this paper, we propose a system for automatically classifying brain tumors based on two deep learning models. A "Fine-tuned proposed model with the attachment of the transfer learning based VGG16" architecture is used for classifying normal and abnormal brain images. Four dense layers are employed in place of the completely connected layers during the tuning process, with the last dense layer equipped with a softmax activation function being used to identify brain tumors. To transform the two-dimensional matrix into a vector, we use Global Average Pooling 2D instead of flattening layers. A total of 71 normal and 81 abnormal MRI images are used in this classification to address the data imbalance problem. On the other hand, we propose a "23-layers CNN" architecture for classifying multiclass brain tumors. In this work, a total of 3064 MRI images are used for training the CNN model. A dropout layer is applied to solve the over fitting issue. In addition, different kernel sizes are integrated with the model to extract the complex features from the MRI images, making the model more robust. Our experimental results indicate that our models reach up to 97.8 and 100 prediction accuracies for our employed, exceeding all other previous studies found in the literature. To summarize, the main contributions of this study are as follows: The "23-layer CNN" framework provides segmentation-free feature extraction techniques that do not require any handcrafted feature extraction method relative to the conventional machine learning methods. In this model, we replace the fully connected layers with four dense layers which facilitate the tuning process. Data imbalance issue is solved in the Harvard Medical dataset by taking an almost equal number of MRI slices in both normal and abnormal tumor classes. The overfitting issue is solved in this study by increasing the number of MRI slices using a data augmentation strategy and introducing the dropout layers within both models. The proposed "23-layers CNN" framework performance is evaluated on both large and small datasets. Results indicate that our framework is able to outperform previous studies found in the literature. To prevent overfitting in a small image dataset, we merged the "23-layers CNN" framework with the transfer learning-based VGG16 model. Results show that the suggested technique performs splendidly in the test images without experiencing any overfitting problems

Conclusion:

This research introduces two deep learning models for identifying brain abnormalities as well as classifying different tumor grades, including meningioma, glioma, and pituitary. The "proposed 23-layer CNN" architecture is designed to work with a relatively large volume of image data, whereas the "Fine-tuned CNN with VGG16" architecture is designed for a limited amount of image data. A comprehensive data augmentation technique is also conducted to enhance the "Fine-tuned CNN with VGG16" model's performance. Our experimental results demonstrated that both models enhance the prediction performance of diagnosis of brain tumors. We achieved 97.8 and 100 prediction accuracy for dataset 1 and dataset 2, respectively outperforming previous studies found in the literature. Therefore, we believe that our proposed methods are outstanding candidates for brain tumor detection.

Sr No	Year	Title	Methodology	Limitation
1)	2020	Brain Tumor Identification and Classification of MRI images using deep learning techniques	Limited evaluation on real-world datasets and potential scalability issues.	The paper proposes a deep learning-based approach for video anomaly detection using spatio-temporal feature extraction and reconstruction.
2)	2022	Brain Tumor Classification and Detection Using Hybrid Deep Tumor Network	The paper presents a novel method for brain classification using machine learning algorithms and feature extraction techniques.	Limited validation on diverse datasets from different geographical locations and grid conditions.
3)	2022	Brain tumor MRI images identification and classification based on the recurrent convolutional neural network	Utilizes deep learning algorithms for semantic segmentation of brain tumor regions in MRI images.	May not account for potential variations in tumor appearance and characteristics across different patient populations.
4)	2022	Accurate brain tumor detection using deep convolutional neural network	Deep learning models for identifying brain abnormalities as well as classifying different tumor grades,	The study may be limited by the availability and quality of the MRI dataset used for training the deep learning model.

Figure 2.1: Literature Review

Chapter 3

Project Vision

3.1 Problem Statement

Brain tumors are serious medical conditions that can significantly impact an individual's quality of life and, in many cases, be life-threatening. Early detection and accurate diagnosis of brain tumors are critical for effective treatment and improved patient survival rates. However, the manual analysis of MRI scans, which is the standard practice, poses challenges. It requires experienced radiologists and is time-consuming, often leading to diagnostic delays and variability in accuracy. In low-resource settings or rural areas where access to radiologists may be limited, these challenges are even more pronounced. Therefore, there is a pressing need for a reliable, automated system that can detect and classify brain tumors with minimal human intervention and high accuracy, ultimately improving the speed and quality of patient care.

3.2 Business Opportunity

The development of an automated brain tumor detection system using MRI images and advanced computational techniques such as deep learning and machine learning presents a significant business opportunity. The healthcare sector is increasingly adopting AI-driven diagnostic tools to enhance clinical decision-making processes. An effective and efficient automated brain tumor detection system can be commercialized as a diagnostic software solution, targeting hospitals, diagnostic centers, and clinics. This technology has the potential to reduce operational costs, minimize diagnostic errors, and increase the efficiency of medical practitioners, making it an attractive solution in the healthcare industry. Furthermore, the system can be integrated into telemedicine platforms, expanding its reach to underserved areas, and improving accessibility to quality healthcare.

3.3 Objectives

Develop a deep learning-based model capable of detecting and classifying brain tumors from MRI images with high accuracy and efficiency.To explore and evaluate the performance of various machine learning algorithms alongside deep learning methods to determine the most effective approach for brain tumor detection.Optimize the models using techniques such as hyperparameter tuning, data augmentation, and model refinement to achieve high precision and minimize false positives and negatives.Create a simple and intuitive user interface for healthcare professionals, allowing them to upload MRI images and receive diagnostic results with minimal effort.To ensure the system is scalable and can be integrated into existing healthcare information systems and telemedicine platforms for broader application and accessibility.

3.4 Project Scope

Gathering a comprehensive dataset of MRI brain images that includes various types of brain tumors as well as normal brain images for model training and evaluation. Preprocessing the images to enhance quality and consistency.Designing, training, and testing deep learning models (e.g., CNN architectures) and comparing their performance with traditional machine learning algorithms.Implementing strategies for improving the model's accuracy, including data augmentation, hyperparameter tuning, and exploring different neural network architectures. Creating a user-friendly interface where healthcare professionals can input MRI images and receive diagnostic results seamlessly.Conducting extensive testing and validation of the model using unseen MRI images to assess its performance in real-world scenarios.Ensuring the developed system can be deployed on different platforms and integrated into hospital management systems or telemedicine solutions.

3.5 Constraints

Obtaining a diverse and comprehensive MRI dataset with labeled brain tumors may be challenging, as access to medical imaging databases is often restricted due to privacy concerns. Deep learning models require significant computational power for training, which may limit the scale of experimentation if sufficient resources (e.g., GPUs) are unavailable.The developed system must comply with healthcare regulations and standards such as the Health Insurance Portability and Accountability Act (HIPAA) for data privacy and security, which may require additional resources for compliance verification.Achieving high accuracy is important, but understanding how the model makes decisions is equally critical in a medical context. Ensuring that the model provides inter pretable results can be challenging.

3.6 Stakeholders Description

Radiologists use medical imaging tools for brain tumor detection and diagnosis. Their feedback on the usability and effectiveness of AI models is crucial. These specialists rely on accurate tumor detection and classification for treatment planning. They benefit from improved diagnostics and may provide insights on clinical needs. Individuals with brain tumors or suspected brain tumors are directly affected by the project's outcomes. Their privacy and informed consent must be protected. Organizations representing patients' interests can provide valuable perspectives on the need for improved diagnostics and potential patient concerns. Hospital and clinic administrators oversee the integration of AI tools into existing systems and processes. They may prioritize projects based on cost-effectiveness and patient outcomes.

3.6.1 Stakeholders Summary

The stakeholders involved in this project include healthcare professionals, hospitals and diagnostic centers, patients, software developers and data scientists, as well as healthcare administrators and policy makers. Healthcare professionals, such as radiologists and neurologists, are the primary users of the system, relying on the automated tool to assist in the detection and diagnosis of brain tumors, which aids their clinical decision-making processes. Hospitals and diagnostic centers implement and deploy the system, benefiting from increased diagnostic accuracy and operational efficiency. Patients, although not direct users of the system, are the ultimate beneficiaries, as they gain access to early and accurate diagnoses that improve their treatment outcomes. Software developers and data scientists play a crucial role in developing, training, and optimizing the models, ensuring the system is accurate, reliable, and user-friendly. Healthcare administrators and policy makers are responsible for overseeing the integration and deployment of the system within healthcare facilities, ensuring that the technology complies with healthcare regulations and aligns with policies to enhance clinical workflows. Each of these stakeholders plays an essential role in the success and effective implementation of the project.

3.6.2 Key High Level Goals and Problems of Stakeholders

To achieve high diagnostic accuracy in detecting and classifying brain tumors from MRI images.To develop a fast and reliable system that can produce diagnostic results in a short time frame, improving the speed of clinical workflows.To make the system accessible to healthcare facilities and telemedicine platforms, especially in low-resource areas. To design the system in a way that allows for scaling up and integration with existing medical information systems.To ensure that the system adheres to healthcare regulations and data privacy standards, providing a secure and trustworthy solution.problems of stack holders To ensure that the system adheres to healthcare regulations and data privacy standards, providing a secure and trustworthy solution.Experience operational inefficiencies and high costs associated with manual MRI image interpretation, as it requires highly skilled radiologists and neurologists.uffer from delayed diagnosis or misdiagnosis, affecting the timeliness and effectiveness of treatment, which can lead to worse health outcomes.Encounter challenges in developing models that are both accurate and interpretable while also ensuring compliance with healthcare regulations.Face difficulties in integrating new technologies within healthcare systems due to regulatory, financial, and technological constraints.

Chapter 4

Software Requirements Specifications

This project is a web-based system for brain tumor detection using MRI images. The user will upload an MRI image through the web interface, and the system will process the image using deep learning models to detect the presence and type of brain tumor. The system will then generate a diagnostic report based on the analysis. The software requirements for this system include:

4.1 List of Features

- Image Upload
- Image Processing and Analysis
- Report Generation

4.2 Functional Requirements

- Image Upload: The user can upload an image to detect tumor
- Model Processing
- Report Generation

4.3 Quality Attributes

- **Reliability:** The system must provide accurate and consistent diagnostic results for MRI images, with minimal downtime.
- **Performance:** The model processing time should be optimized to deliver results within a few seconds after image upload.
- **Security:** User data, including MRI images and diagnostic reports, must be securely stored and transmitted, adhering to data protection standards like HIPAA.
- **Usability:** The user interface should be intuitive, allowing easy navigation for users with varying levels of computer literacy.

4.4 Non-Functional Requirements

- **Performance Requirements:** The system must be able to process and analyze an MRI image within 10 seconds.
- **Availability Requirements:** The system should have an uptime of 99.9

4.5 Use Cases/Use Case Diagram

Actors: Doctor

Description: The Doctor logs in and uploads an MRI image through the web interface for analysis.

Post condition: The image is processed, and a report is generated.

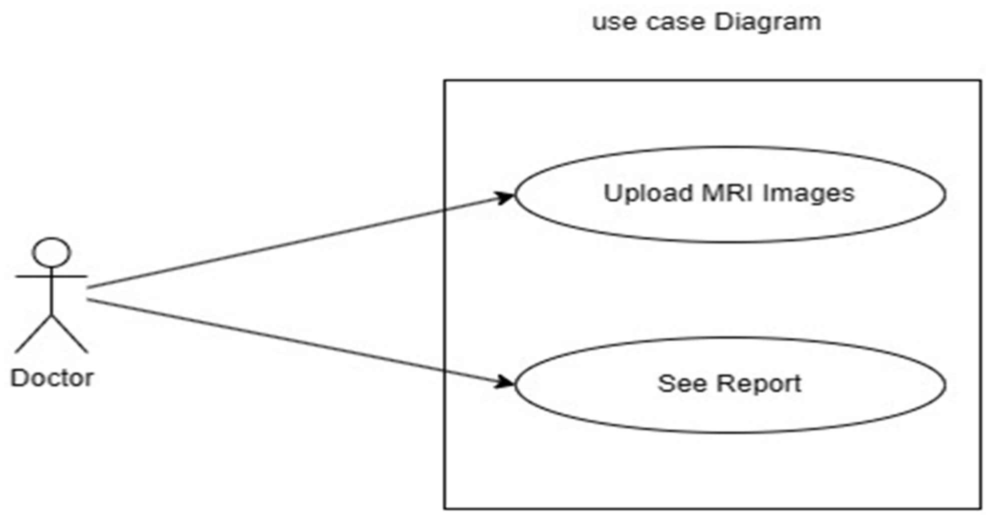


Figure 4.1: Use Case Diagram

4.6 Test Plan (Test Level, Testing Techniques)

- **Unit Testing:** Test individual components like the image upload functionality, model processing, and report generation separately to ensure they work as expected.
- **Integration Testing:** Test the integration between the front-end and back-end components, ensuring smooth data flow and correct interactions.
- **System Testing:** Perform end-to-end testing on the entire system to verify that all components work together and meet requirements.
- **Acceptance Testing:** Test the system with sample MRI images to validate that it meets the acceptance criteria set by healthcare professionals.

Testing Techniques: Automated testing tools (e.g., Selenium for web testing) and manual testing methods will be used to ensure the reliability and quality of the system.

4.7 Software Development Plan

Development Methodology: The project will follow the Agile development methodology, allowing for iterative progress, flexibility, and continuous integration of features.

Milestones:

- Week 1-3: Requirement analysis, planning, and environment setup.
- Week 4-6: Front-end development and initial back-end setup (user authentication and image upload).
- Week 7-9: Model integration and testing.
- Week 10-12: Report generation, UI enhancements, and testing.
- Week 13-15: Deployment and system testing, feedback, and iterations.

4.8 Wire-frames

Wireframes will be designed for each key part of the user interface, including:

- Image Upload Page: A drag-and-drop or file-upload interface for users to upload MRI images, with options for file type and format information.
- Diagnostic Report Page: A layout that displays the report with tumor type, location, and confidence level, along with download options.

4.9 UI Screens

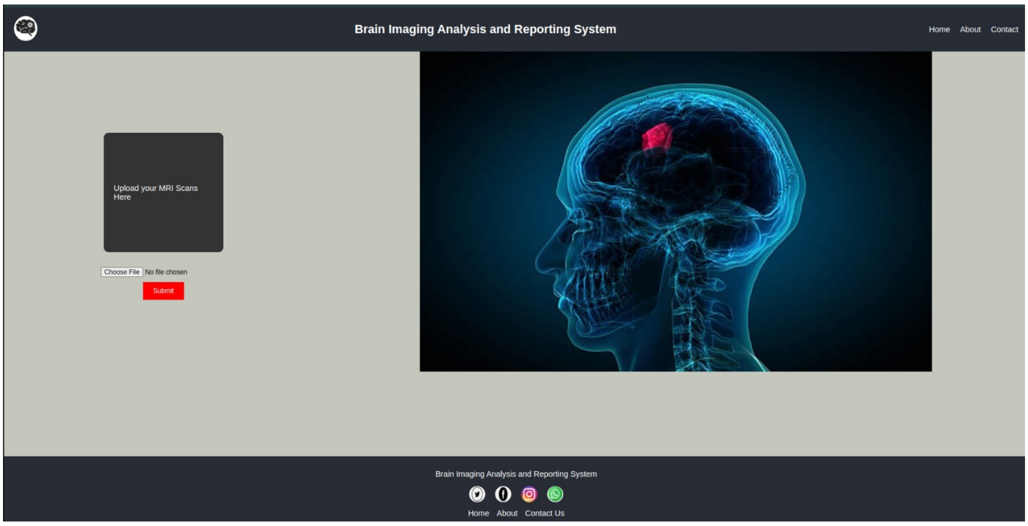


Figure 4.2: UI Screen of the Application

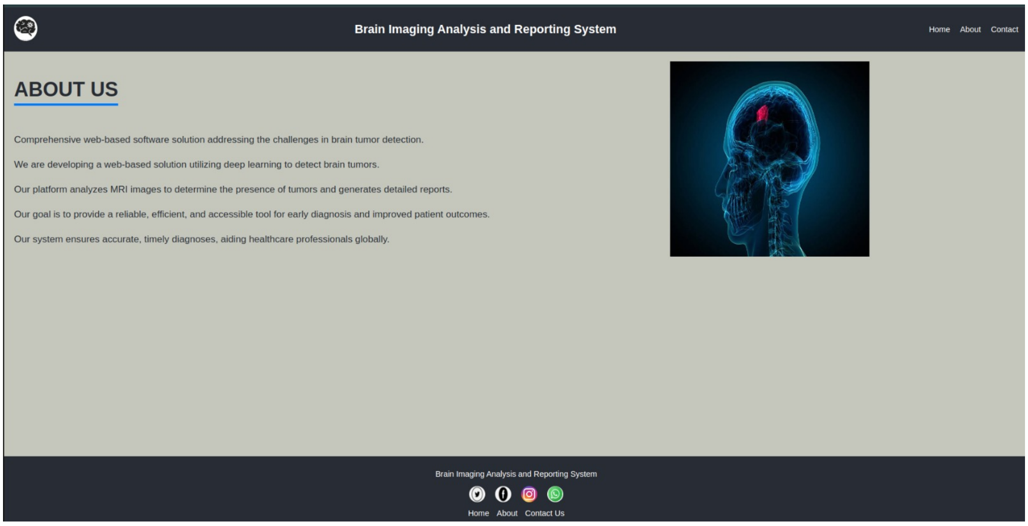


Figure 4.3: About page

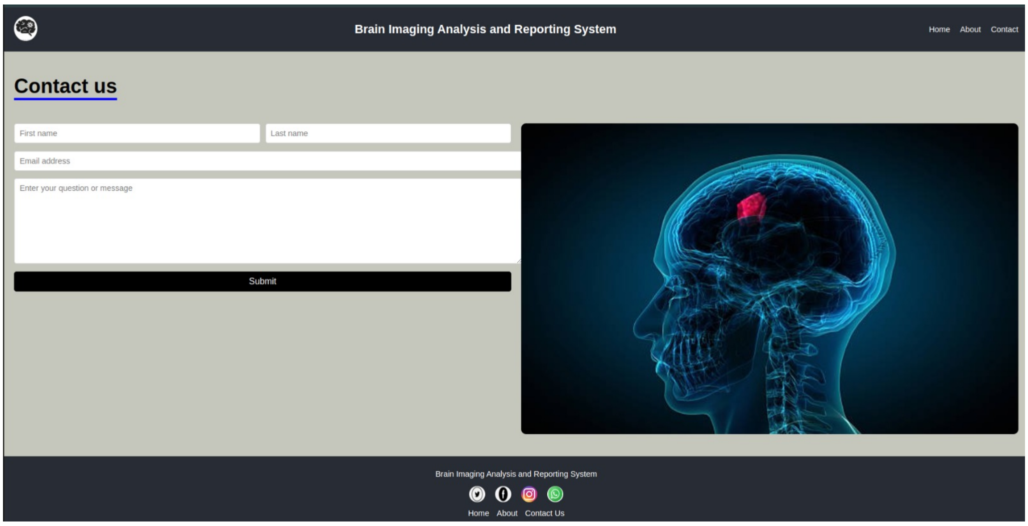


Figure 4.4: Contact page

Chapter 5

Iteration Plan

This chapter outlines the iteration plan of the project, detailing how the development process progresses through various phases to complete all requirements. The project is structured into several key phases, ensuring a systematic approach to meeting the objectives. The development of the project is broken down into the following iterations:

- **Midterm FYP 1:**

  In the initial phase of the project (Midterm FYP 1), the focus was on designing and developing the user interface (UI) and user experience (UX). This phase was crucial as it set the foundation for the project's frontend and visual elements. Key activities during this phase included:

  UI/UX Design:

  Designing the layout and wireframes for the web application, ensuring an intuitive and user-friendly experience for users who will upload MRI images and view diagnostic reports.

  Frontend Development:

  Initial development of the frontend using HTML, CSS, and JavaScript (React/Vue.js) to implement the designed UI/UX.

  Dataset Collection:

  Researching and collecting a comprehensive dataset of MRI images, ensuring it included a variety of brain tumor types and normal cases for training and testing the deep learning model.

This phase was essential for laying down the project structure and gathering the necessary data to develop the machine learning model in later stages.

- **Final FYP 1:**

  In the Final FYP 1 phase, the focus remained on finalizing the frontend development and preparing the dataset for the next stage of model development. Key activities during this phase included:

  Frontend Completion:

  Refining the frontend elements, ensuring all UI components were functional and visually aligned with the project goals. The image upload functionality was tested to validate that users could interact with the system correctly.

- **Midterm FYP 2:**

  In the Midterm FYP 2 phase, the focus shifted towards the development of the core deep learning model and image processing capabilities. This phase marked the start of implementing the main functionality of the system. Key activities included:

  Model Development:
  Developing and training a deep learning model using convolutional neural networks (CNN)and YOLO on the preprocessed MRI dataset. This involved experimenting with different architectures and tuning hyperparameters to optimize the model's performance.

  Image Processing Module:
  Implementing the image processing pipeline that accepts input images from users, preprocesses them, and passes them through the trained model to detect and classify brain tumors.

  Backend Development:
  Initial development of the backend using Python (Django/Flask), integrating the model with the backend server to process the uploaded images and generate diagnostic results.

Testing the Model:

Conducting testing with sample images to validate the accuracy and reliability of the model, identifying areas for further improvement and optimization. This phase was critical for establishing the technical backbone of the project, allowing for further refinement and integration of the system components.

- **Final FYP 2:**

  The final phase (Final FYP 2) focused on completing the backend development and integrating all components to deliver a fully functional web application. The key activities during this phase included:

  Frontend and Backend Integration:
  Integrating the frontend interface with the backend system, ensuring that image uploads by users are processed seamlessly through the model and that diagnostic reports are generated and displayed correctly.

  Backend Completion:
  Finalizing the backend by ensuring secure data handling, efficient processing, and integration of all necessary APIs for smooth communication between the frontend and backend.

  System Testing and Optimization:
  Conducting extensive system testing to validate the end-to-end functionality of the system. This included testing user interactions, image processing capabilities, and the accuracy of the diagnostic reports generated.

  Deployment Preparation:
  Preparing the system for deployment on a cloud platform to ensure scalability and reliability, setting up databases and configuring necessary security measures (e.g., SSL certificates) for secure communication.

  {Final Refinements:
  Applying final optimizations to both the model and the user interface based on testing feedback, ensuring that the system meets all performance and usability requirements. The completion of this phase resulted in a fully integrated and functional

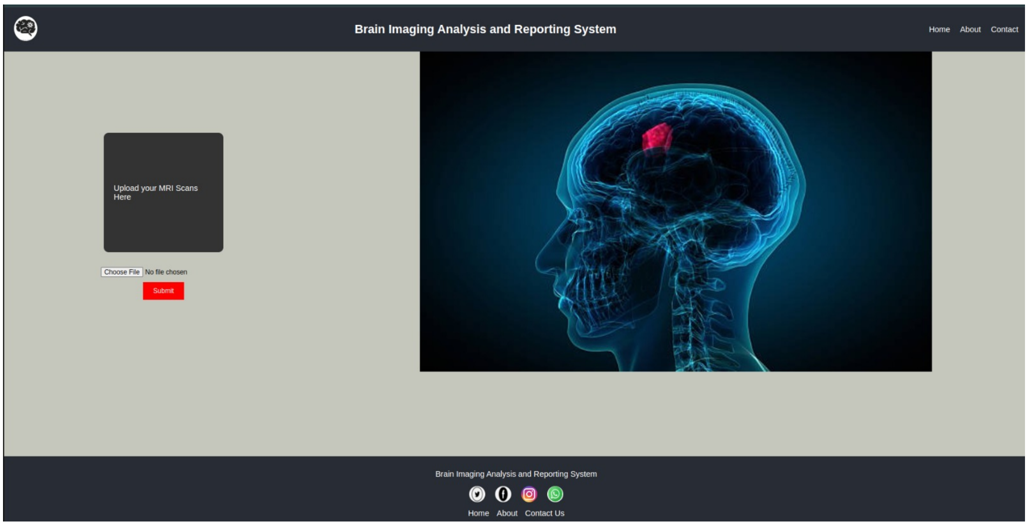


Figure 5.1: main sacreen

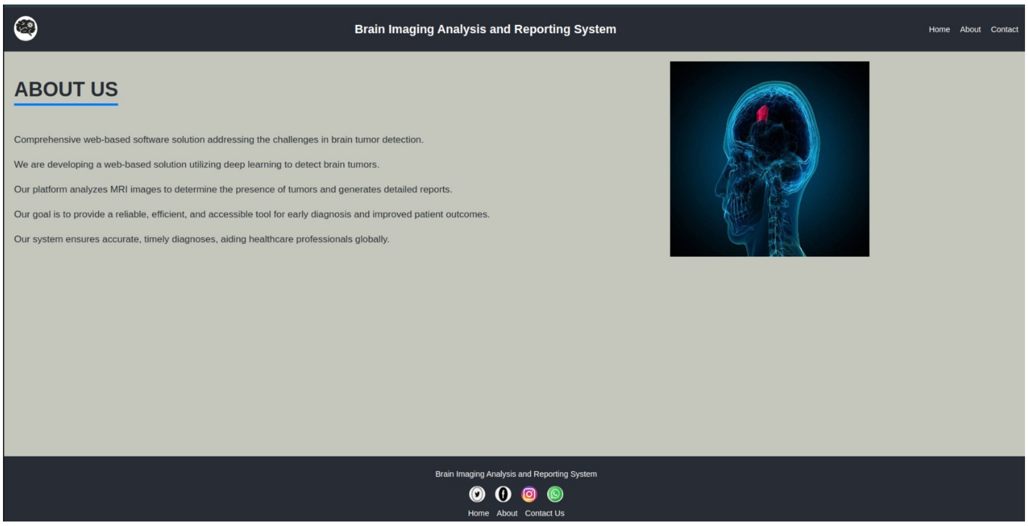


Figure 5.2: About software

system ready for deployment and testing in a real-world scenario.

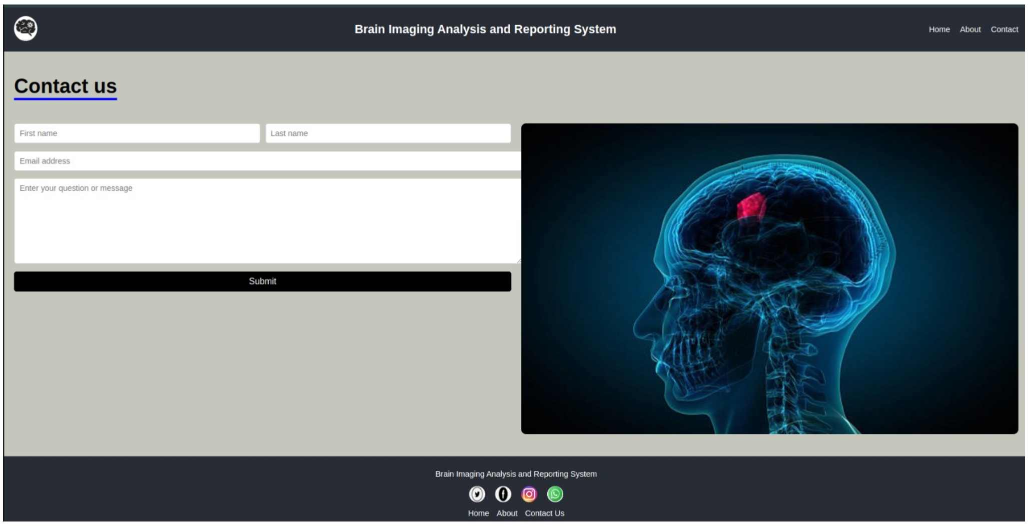


Figure 5.3: contact page

Chapter 6

Iteration 1

Coding structure:

Directory structure of a project, and based on the organization of files and folders, it appears to follow a React-based frontend architecture with some backend elements. Here's a breakdown of the structure

Structural design:

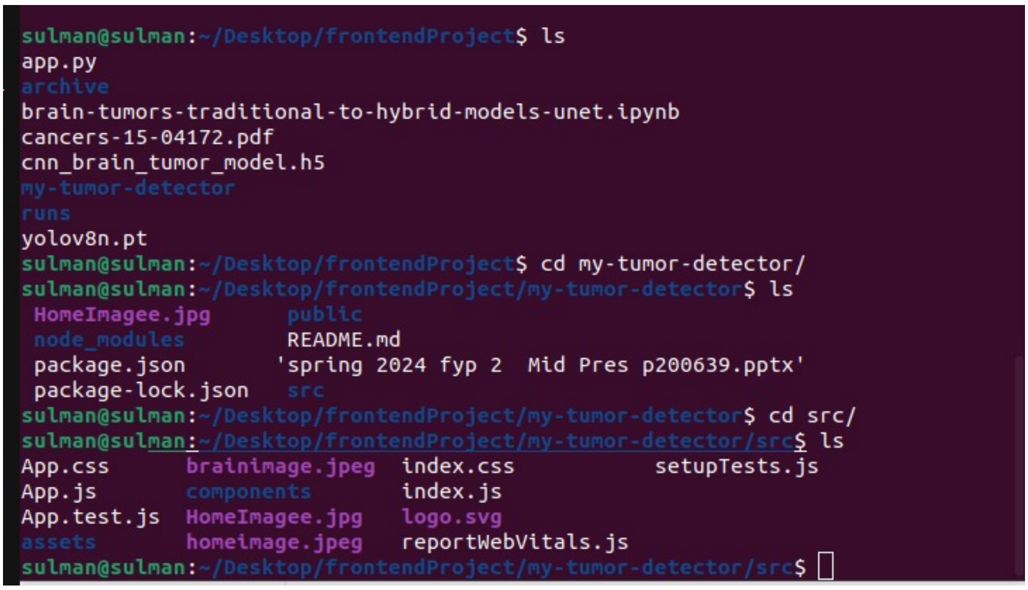
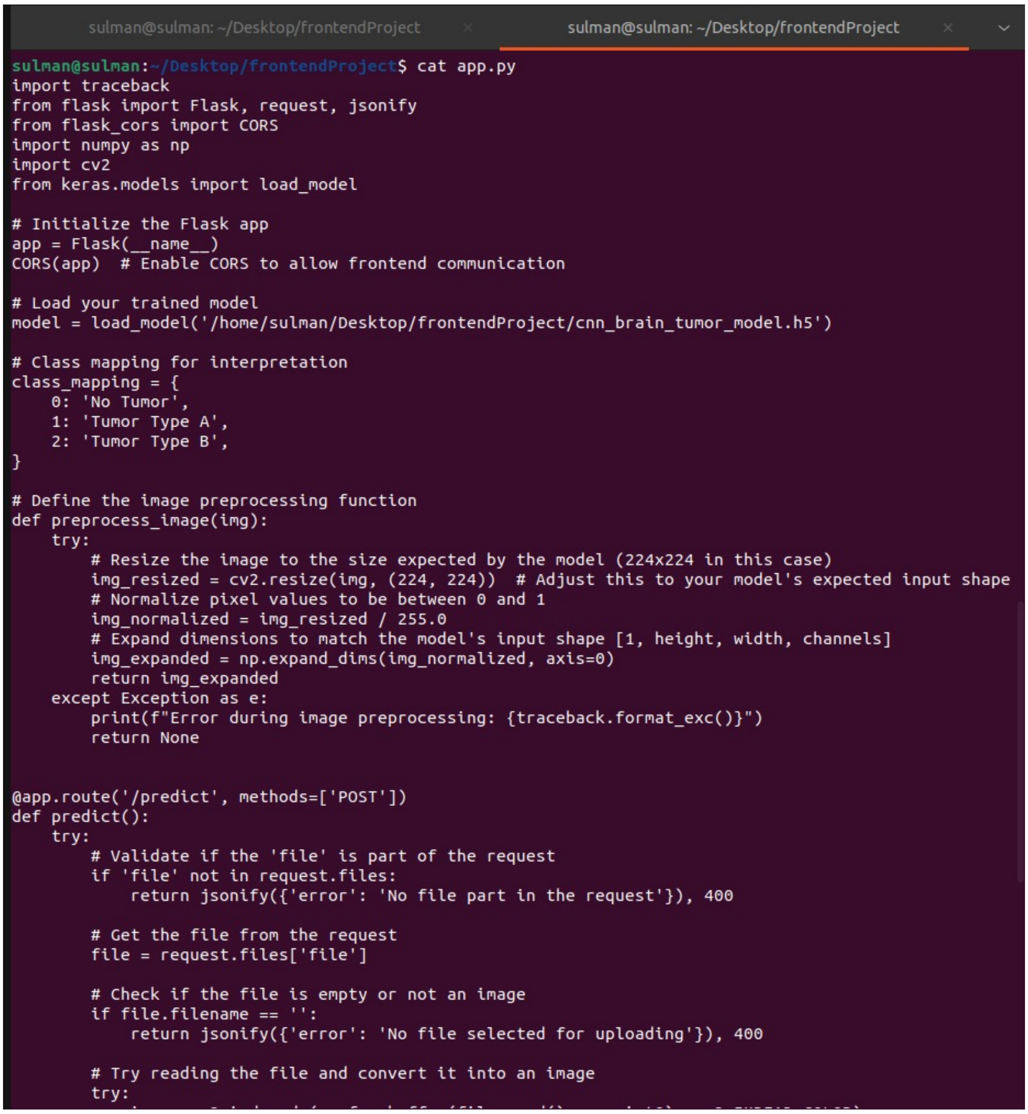


Figure 6.1: Directory Locations



```
sulman@sulman:~/Desktop/frontendProject$ cat app.py
import traceback
from flask import Flask, request, jsonify
from flask_cors import CORS
import numpy as np
import cv2
from keras.models import load_model

# Initialize the Flask app
app = Flask(__name__)
CORS(app)  # Enable CORS to allow frontend communication

# Load your trained model
model = load_model('/home/sulman/Desktop/frontendProject/cnn_brain_tumor_model.h5')

# Class mapping for interpretation
class_mapping = {
    0: 'No Tumor',
    1: 'Tumor Type A',
    2: 'Tumor Type B',
}

# Define the image preprocessing function
def preprocess_image(img):
    try:
        # Resize the image to the size expected by the model (224x224 in this case)
        img_resized = cv2.resize(img, (224, 224))  # Adjust this to your model's expected input shape
        # Normalize pixel values to be between 0 and 1
        img_normalized = img_resized / 255.0
        # Expand dimensions to match the model's input shape [1, height, width, channels]
        img_expanded = np.expand_dims(img_normalized, axis=0)
        return img_expanded
    except Exception as e:
        print(f"Error during image preprocessing: {traceback.format_exc()}")
        return None


@app.route('/predict', methods=['POST'])
def predict():
    try:
        # Validate if the 'file' is part of the request
        if 'file' not in request.files:
            return jsonify({'error': 'No file part in the request'}), 400

        # Get the file from the request
        file = request.files['file']

        # Check if the file is empty or not an image
        if file.filename == '':
            return jsonify({'error': 'No file selected for uploading'}), 400

        # Try reading the file and convert it into an image
        try:
```

Figure 6.2: Python Flask Code

```
sulman@sulman:~/Desktop/frontendProject/my-tumor-detector/src$ cat App.js
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import Navbar from './components/Navbar';
import Home from './components/Home';
import Footer from './components/Footer';
import './App.css';  // Ensure you're importing the CSS file

const App = () => {
  return (
    <Router>
      <div className="app-container">  {/* Wrap all content in a flex container */}
        <Navbar />
        <div className="content-wrap">  {/* This div will hold the page content */}
          <Routes>
            <Route path="/" element={<Home />} />
            {/* You can add more routes here */}
          </Routes>
        </div>
        <Footer />
      </div>
    </Router>
  );
};

export default App;
```

Figure 6.3: APP.js Code

```
sulman@sulman:~/Desktop/frontendProject/my-tumor-detector/src/components$ cat Contact.js
import React, { useState } from 'react';
import brainImage from '../brainimage.jpeg'; // Make sure the image path is correct

const Contact = () => {
  const [formData, setFormData] = useState({
    firstName: '',
    lastName: '',
    email: '',
    message: '',
  });

  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData({
      ...formData,
      [name]: value,
    });
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    // You can handle the form submission here, e.g., send the data to an API
    alert(`Message submitted by ${formData.firstName} ${formData.lastName}`);
    // Reset form
    setFormData({ firstName: '', lastName: '', email: '', message: '' });
  };

  return (
    <div className="contact-container" style={{ padding: '20px', color: '#000' }}>
      <h1 style={{ fontSize: '2.5rem', borderBottom: '5px solid #00f', display: 'inline-block' }}>Con
tact us</h1>
      <div style={{ display: 'flex', justifyContent: 'space-between', alignItems: 'flex-start', margi
nTop: '20px' }}>
        <div style={{ flex: 1, paddingRight: '20px' }}>
          <form onSubmit={handleSubmit} style={{ display: 'flex', flexDirection: 'column', gap: '15px
' }}>
            <div style={{ display: 'flex', gap: '10px' }}>
              <input
                type="text"
                name="firstName"
                placeholder="First name"
                value={formData.firstName}
                onChange={handleChange}
                style={inputStyle}
                required
              />
              <input
                type="text"
                name="lastName"
                placeholder="Last name"
                value={formData.lastName}
                onChange={handleChange}
                style={inputStyle}
```

Figure 6.4: home.js Code

```
sulman@sulman:~/Desktop/frontendProject/my-tumor-detector/src/components$ ls
About.css  About.js  Contact.css  Contact.js  Footer.js  Home.css  Home.js  Navbar.css  Navbar.js
sulman@sulman:~/Desktop/frontendProject/my-tumor-detector/src/components$ cat About.js
import React from 'react';
import brainImage from '../brainimage.jpeg'; // Make sure the image path is correct

const About = () => {
  return (
    <div className="about-container" style={{ display: 'flex', justifyContent: 'space-between', paddi
ng: '20px', color: '#282c34' }}>
      <div style={{ flex: 1, paddingRight: '20px' }}>
        <h2 style={{ fontSize: '2.5rem', fontWeight: 'bold', borderBottom: '5px solid #007BFF', displ
ay: 'inline-block', paddingBottom: '5px' }}>ABOUT US</h2>
        <p style={{ marginTop: '20px', fontSize: '1.2rem', lineHeight: '1.6' }}>
          Comprehensive web-based software solution addressing the challenges in brain tumor detectio
n.
        </p>
        <p style={{ fontSize: '1.2rem', lineHeight: '1.6' }}>
          We are developing a web-based solution utilizing deep learning to detect brain tumors.
        </p>
        <p style={{ fontSize: '1.2rem', lineHeight: '1.6' }}>
          Our platform analyzes MRI images to determine the presence of tumors and generates detailed
 reports.
        </p>
        <p style={{ fontSize: '1.2rem', lineHeight: '1.6' }}>
          Our goal is to provide a reliable, efficient, and accessible tool for early diagnosis and i
mproved patient outcomes.
        </p>
        <p style={{ fontSize: '1.2rem', lineHeight: '1.6' }}>
          Our system ensures accurate, timely diagnoses, aiding healthcare professionals globally.
        </p>
      </div>
      <div style={{ flex: 1, display: 'flex', justifyContent: 'center' }}>
        <img src={brainImage} alt="Brain MRI scan" style={{ width: '100%', height: 'auto', maxWidth:
'400px' }} />
      </div>
    </div>
  );
};

export default About;
```

Figure 6.5: Header and Footer Code

Chapter 7

Iteration 2

The goal in the mid of FYP2 was to start working on the backend and working on model development and testing.

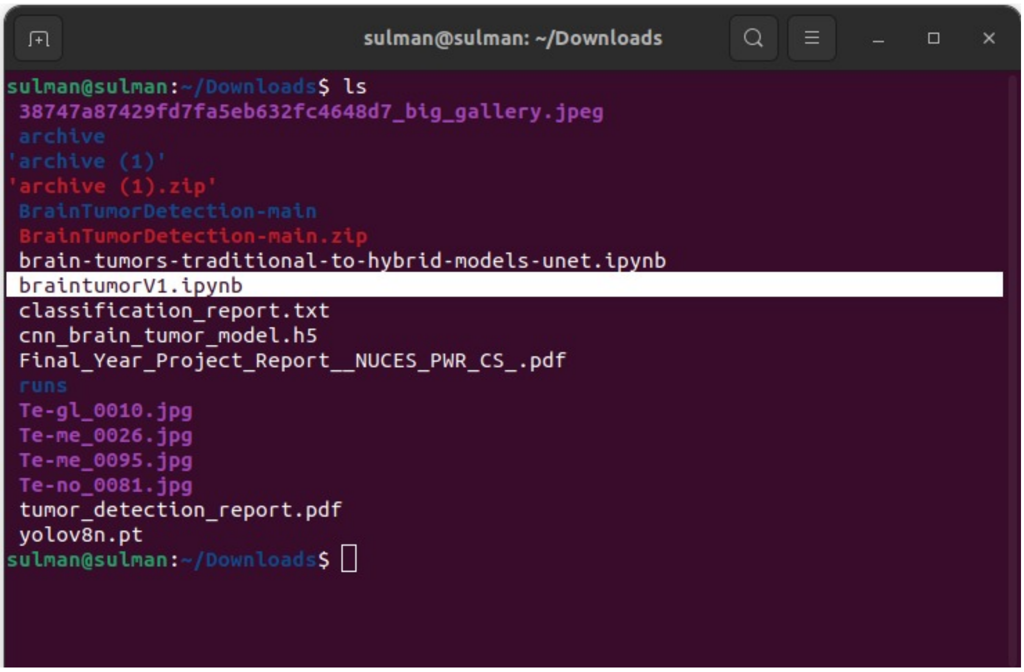


Figure 7.1: Backend ipynb file

```
In [1]: import tensorflow as tf
import keras
import numpy as np
import os
import cv2
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, Flatten, Dense, Dropout

# YOLO model import (ultralytics)
from ultralytics import YOLO

# Load YOLOv8 model pre-trained on COCO dataset
yolo_model = YOLO('yolov8n.pt')  # You can replace with other YOLOv8 variants if needed

def load_and_preprocess_images(directory, target_size=(224, 224)):
    X, y = [], []
    class_folders = os.listdir(directory)
    class_folders.sort()  # Ensure consistent class order

    for class_index, class_folder in enumerate(class_folders):
        class_path = os.path.join(directory, class_folder)
        if os.path.isdir(class_path):
            for image_file in os.listdir(class_path):
                image_path = os.path.join(class_path, image_file)
                try:
                    image = cv2.imread(image_path)
                    image = cv2.resize(image, target_size)
                    X.append(image)
                    y.append(class_index)
                except Exception as e:
                    print(f"Error loading {image_path}: {e}")

    X = np.array(X)
    y = np.array(y)
    return X, y

# Load and preprocess images
path = "archive/Training"
X, y = load_and_preprocess_images(path)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Define a simple CNN model
def create_cnn(input_shape, num_classes):
```

Figure 7.2: Load and preprocessing and model code

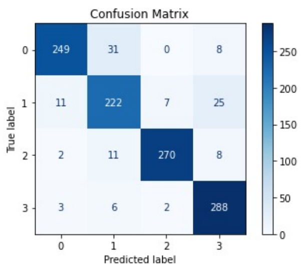
```
In [7]: from fpdf import FPDF

# Function to generate a PDF report
def generate_pdf_report(image_path, prediction):
    pdf = FPDF()
    pdf.add_page()

    # Title
    pdf.set_font("Arial", size=16)
    pdf.cell(200, 10, txt="Brain Tumor Detection Report", ln=True, align="C")

    # Add prediction result
    pdf.set_font("Arial", size=12)
    pdf.cell(200, 10, txt=f"Prediction: {prediction}", ln=True, align="L")

    # Insert the image
    pdf.image(image_path, x=10, y=30, w=100)

    # Save PDF
    output_path = "tumor_detection_report.pdf"
    pdf.output(output_path)
    print(f"PDF report generated: {output_path}")

# Example usage:
image_path = "Te-me_0026.jpg"  # Replace with the actual image input path
prediction = handle_user_image_input(image_path)
if prediction:
    generate_pdf_report(image_path, prediction)

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
Prediction: No Tumor
PDF report generated: tumor_detection_report.pdf

In [8]: # Load YOLOv8 model pre-trained on COCO dataset
yolo_model = YOLO('yolov8n.pt')  # You can replace with other YOLOv8 variants if needed

# Load your trained CNN model
cnn_model = tf.keras.models.load_model('cnn_brain_tumor_model.h5')

# Function to preprocess image for CNN
def preprocess_image(image):
    image = image.resize((224, 224))  # Resize to match model input
    image = np.array(image)
    image = image / 255.0  # Normalize pixel values
    image = np.expand_dims(image, axis=0)  # Add batch dimension
```

Figure 7.3: Report generation code

```
In [14]: from sklearn.metrics import confusion_matrix, classification_report, ConfusionMatrixDisplay
         from sklearn.metrics import precision_score, recall_score, f1_score

         # After evaluating the CNN model on the test set
         y_pred = cnn_model.predict(X_test)
         y_pred_classes = np.argmax(y_pred, axis=1)

         # Generate the confusion matrix
         cm = confusion_matrix(y_test, y_pred_classes)

         # Display the confusion matrix
         ConfusionMatrixDisplay(confusion_matrix=cm).plot(cmap='Blues')
         plt.title("Confusion Matrix")
         plt.show()

         # Generate a classification report
         report = classification_report(y_test, y_pred_classes, target_names=os.listdir(path))
         print(report)

         # Save classification report to a text file
         with open("classification_report.txt", "w") as f:
             f.write(report)

         # Calculate precision, recall, and F1 score
         precision = precision_score(y_test, y_pred_classes, average='weighted')
         recall = recall_score(y_test, y_pred_classes, average='weighted')
         f1 = f1_score(y_test, y_pred_classes, average='weighted')

         print(f"Precision: {precision:.2f}")
         print(f"Recall: {recall:.2f}")
         print(f"F1 Score: {f1:.2f}")
```

Figure 7.4: Accuracy Code

```
36/36 ━━━━━━━━━━━━━━━━━━━━ 6s 157ms/step
```




```
              precision    recall  f1-score   support

      glioma       0.94      0.86      0.90       288
  meningioma       0.82      0.84      0.83       265
     notumor       0.97      0.93      0.95       291
   pituitary       0.88      0.96      0.92       299

    accuracy                           0.90      1143
   macro avg       0.90      0.90      0.90      1143
weighted avg       0.90      0.90      0.90      1143

Precision: 0.90
Recall: 0.90
F1 Score: 0.90
```

Figure 7.5: Accuracy result

Chapter 8

Iteration 3

8.1 Flow Diagram

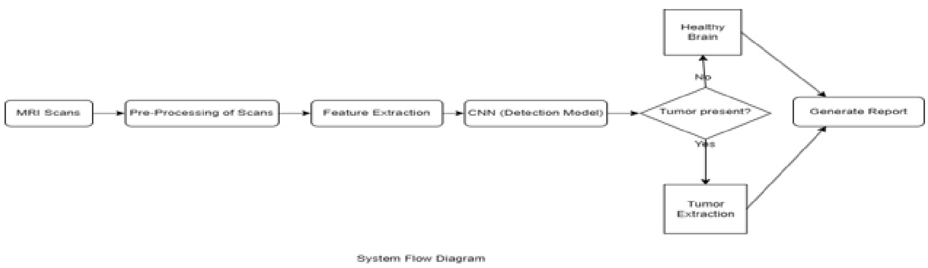


Figure 8.1: Flow diagram

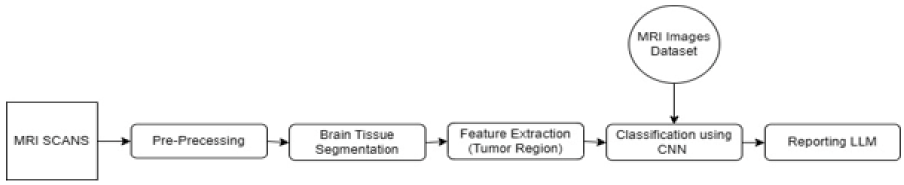


Figure 8.2: Data Flow diagram

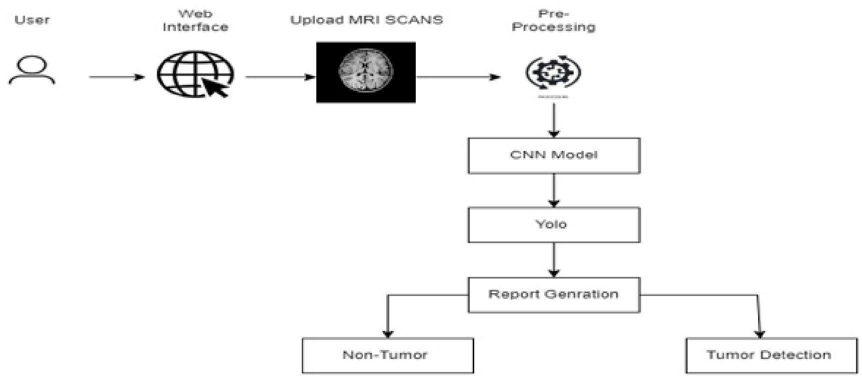


Figure 8.3: Architecture diagram

8.2 Data Flow Diagram (DFD)

8.3 Interaction Overview Diagram

8.4 Algorithm Design

The user accesses a web interface where they can upload MRI scans of the brain. This interface serves as the entry point for data input. Data Upload:

The user uploads MRI scans, which could be in various formats (e.g., DICOM, JPEG). These scans need to be processed to be used as input for the model. Pre-Processing:

Before feeding the scans to the model, pre-processing is performed. Pre-processing can include: Resizing images to a uniform size. Normalization of pixel values. Data augmentation (e.g., flipping, rotating) to improve model generalization. Segmentation to isolate the brain region from the rest of the image. Noise reduction techniques such as Gaussian filtering or median filtering. CNN Model:

The pre-processed MRI scans are fed into a Convolutional Neural Network (CNN). This model is designed to detect patterns in images and is trained to classify MRI scans as potentially having a tumor or not. The CNN extracts features from the images and identifies areas of interest that may indicate the presence of a tumor. YOLO (You Only Look Once) Detection:

After the CNN processes the images, a YOLO model is used for more precise localization and detection of tumors. YOLO is a popular object detection model that predicts bounding boxes around objects of interest in a single pass through the network, making it suitable for real-time detection tasks. It outputs bounding boxes that localize the detected tumor areas within the MRI scans. Report Generation:

Based on the output from the CNN and YOLO models, a report is generated. The report will classify the scan as either "Non-Tumor" (no tumor detected) or "Tumor Detection" (tumor detected). In the case of tumor detection, additional details may include the location, size, and confidence score of the detected tumor. Outcome:

The final output could be presented to the user as: Non-Tumor: No abnormal growth detected in the MRI scans. Tumor Detection: Abnormal growth is detected, with further information about the suspected tumor location and size provided in thereport.

8.5 Development Phase

8.5.1 Unit Test

8.5.2 Suites or Test Cases

8.5.3 System-Level Test Suites, Test Cases

8.5.4 Configuration/ Setup and Tool Manual (Optional)

Chapter 9

Iteration 4

This chapter will have some of the artifacts based on system design. The requirements analysis section is same for all the systems while the design may vary. There may have two types of designs the structural design or . First section is for the structural design.

9.1 Flow Diagram

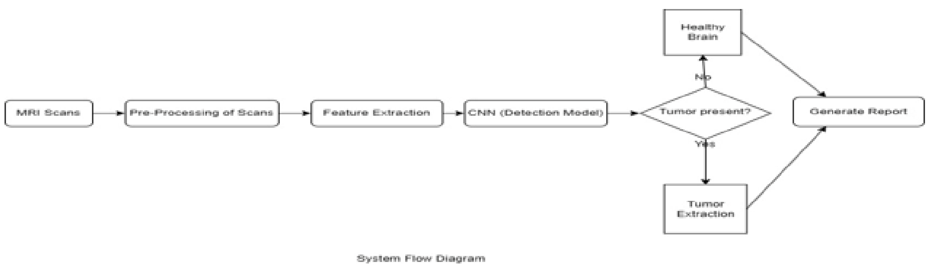


Figure 9.1: Flow diagram

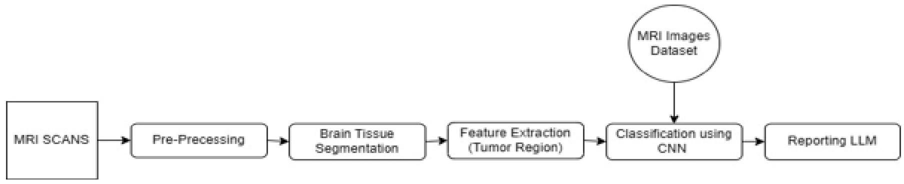


Figure 9.2: Data Flow diagram

9.2 Data Flow Diagram (DFD)

9.3 Interaction Overview Diagram

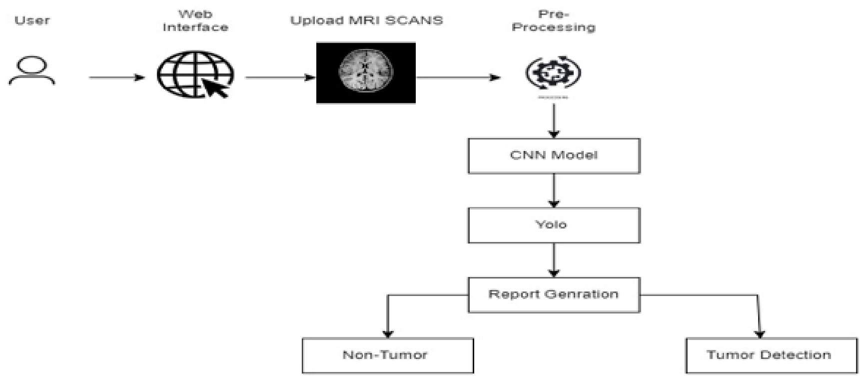


Figure 9.3: Architecture diagram

9.4 Algorithm Design

The user accesses a web interface where they can upload MRI scans of the brain. This interface serves as the entry point for data input. Data Upload:

The user uploads MRI scans, which could be in various formats (e.g., DICOM, JPEG). These scans need to be processed to be used as input for the model. Pre-Processing:

Before feeding the scans to the model, pre-processing is performed. Pre-processing can include: Resizing images to a uniform size. Normalization of pixel values. Data augmentation (e.g., flipping, rotating) to improve model generalization. Segmentation to isolate the brain region from the rest of the image. Noise reduction techniques such as Gaussian filtering or median filtering. CNN Model:

The pre-processed MRI scans are fed into a Convolutional Neural Network (CNN). This model is designed to detect patterns in images and is trained to classify MRI scans as potentially having a tumor or not. The CNN extracts features from the images and identifies areas of interest that may indicate the presence of a tumor. YOLO (You Only Look Once) Detection:

After the CNN processes the images, a YOLO model is used for more precise localization and detection of tumors. YOLO is a popular object detection model that predicts bounding boxes around objects of interest in a single pass through the network, making it suitable for real-time detection tasks. It outputs bounding boxes that localize the detected tumor areas within the MRI scans. Report Generation:

Based on the output from the CNN and YOLO models, a report is generated. The report will classify the scan as either "Non-Tumor" (no tumor detected) or "Tumor Detection" (tumor detected). In the case of tumor detection, additional details may include the location, size, and confidence score of the detected tumor. Outcome:

The final output could be presented to the user as: Non-Tumor: No abnormal growth detected in the MRI scans. Tumor Detection: Abnormal growth is detected, with further information about the suspected tumor location and size provided in the report.

9.5 Development Phase

9.5.1 Unit Test

9.5.2 Suites or Test Cases

9.5.3 Deployment Diagram

9.5.4 System-Level Test Suites, Test Cases

9.5.5 Configuration/ Setup and Tool Manual (Optional)

Chapter 10

Implementation Details

Chapter 11

User Manual

This chapter will have the user manual.

Chapter 12

Conclusions and Future Work

Bibliography