KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

KUMASI – GHANA

COLLEGE OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

PROJECT DOCUMENTATION

ON

BRAIN TUMOUR DETECTION SYSTEM

STUDENTS:

NICHOLAS INCOOM -9406219

TEYE EUGENE NUER- 9413919

PROJECT SUPERVISOR: DR. KWABENA OPOKU AGYEMANG

AUGUST 2023

TABLE OF CONTENTS

LIST OF FIGURES

DECLARATION

The project work entitled “Brain Tumour Detection System” was submitted to the Kwame Nkrumah University of Science and Technology, and this project was done under the supervision of Dr. Kwabena Agyemang, in the Department of Computer Science.

……………………………………. …..……………..................

NICHOLAS FAYAH INCOOM DATE

……………………………………. …..……………..................

TEYE EUGENE NUER DATE

SUPERVISOR

……………………………………. …..……………..................

DR. KWABENA OPOKU AGYEMANG DATE

DEDICATION

We dedicate this work to the almighty God for his grace, guidance, and knowledge from the beginning to the completion of this work. To our loving parents who have never stopped believing in us and have supported us all through. We dedicate this work to all lecturers of the Computer Science Department, especially our supervisor, Dr. Kwabena Agyemang

ACKNOWLEDGEMENT

We first acknowledge the almighty God who has been our guide throughout this work. We also acknowledge our project supervisor Dr. Kwabena Agyemang; who gave us this golden opportunity to work on this project.

We owe a great debt to all those who invested countless hours in correction, giving ideas to the content of this project. We express our immense gratitude to all and also to our parents and friends who helped a lot in finalizing this project within a limited time frame.

ABSTRACT

Advancements in deep learning and artificial intelligence (AI) have paved the way for transformative applications in healthcare, particularly in medical imaging. This documentation presents a comprehensive overview of a brain tumor detection project that leverages Convolutional Neural Networks (CNNs) to accurately identify brain tumors from medical images. The project's primary objective is to develop an innovative solution that augments the capabilities of medical professionals in diagnosing brain tumors with a high level of accuracy.

The project encompasses the creation of a diverse and meticulously annotated dataset of brain tumor images, which serves as the foundation for training and evaluating the CNN model. Rigorous experimentation and hyperparameter tuning are conducted to optimize model performance, ensuring reliable predictions across a spectrum of brain tumor variations and imaging modalities. Ethical considerations are paramount, guiding decisions related to patient privacy, data security, and AI's responsible integration within the medical field.

An integral aspect of the project involves collaboration with medical experts for clinical validation. By comparing the model's diagnostic outcomes against expert radiologists' interpretations, the project establishes the model's alignment with real-world medical practices. The documentation further delves into the potential ethical implications, challenges, and considerations surrounding AI in healthcare, underscoring the project's commitment to ethical and responsible AI deployment.

Throughout the documentation, readers will find detailed insights into the technical architecture of the developed brain tumor detection system, its training methodology, preprocessing techniques, model evaluation, and considerations for future work. Additionally, the project's journey highlights lessons learned, recommendations for future enhancements, and strategies for commercialization to bridge the gap between technological innovation and medical impact.

This brain tumor detection project encapsulates the potential of AI to revolutionize medical diagnostics, emphasizing the value of interdisciplinary collaboration, ethical consciousness, and continuous innovation. As AI continues to transform healthcare, this project serves as a

testament to the possibilities of leveraging technology for the betterment of patient outcomes and clinical practices.

CHAPTER 1 - INTRODUCTION

1.0 BACKGROUND

In the ever-evolving landscape of medical diagnostics, advancements in technology have ushered in a new era of accuracy and efficiency. One critical area that has seen remarkable progress is the detection of brain tumors using cutting-edge deep learning methods. Brain tumors, whether benign or malignant, pose significant challenges to healthcare professionals due to their diverse characteristics and potential impact on patients' lives. Timely and accurate detection plays a pivotal role in ensuring effective treatment planning and improved patient outcomes.

Traditionally, the diagnosis of brain tumors relied heavily on manual interpretation of medical imaging scans, such as Magnetic Resonance Imaging (MRI). This process, while crucial, could be time-consuming, subjective, and prone to human error. As a response to these challenges, the integration of deep learning techniques has emerged as a transformative solution. Deep learning, a subset of machine learning, leverages complex neural networks to learn intricate patterns and features from large datasets. When applied to medical image analysis, deep learning models have demonstrated remarkable capabilities in identifying subtle anomalies and patterns that might be missed by the human eye.

The "Brain Tumor Detection System " project is a testament to the convergence of medical expertise and technological innovation. By harnessing the power of deep learning, this project seeks to revolutionize the way brain tumors are detected, diagnosed, and treated. This endeavor is underpinned by the recognition that swift and accurate detection of brain tumors is paramount to the success of medical interventions.

The project acknowledges the limitations of conventional diagnosis methods and aims to bridge the gap between technological advancements and healthcare practices. By developing an automated system capable of processing MRI scans, identifying potential tumor regions, and providing visualizations of the findings, this project aspires to empower medical professionals with a valuable diagnostic tool. This tool is not intended to replace medical expertise but rather to complement it, enabling medical experts to make informed decisions backed by advanced technological insights.

Furthermore, the project's ethical considerations are deeply ingrained in its framework. The responsible usage of patient data, addressing potential biases in algorithmic predictions, and ensuring regulatory compliance are essential pillars of the project's foundation. The ultimate goal is to develop a system that upholds the highest standards of patient confidentiality, fairness, and medical accuracy.

In conclusion, the "Brain Tumor Detection System " project is a convergence of innovation, medical expertise, and technological prowess. It aspires to redefine the landscape of brain tumor diagnostics, accelerate accurate detections, and contribute to the ongoing evolution of healthcare practices. By marrying the intricate complexities of deep learning with the critical demands of medical diagnosis, the project aims to pave the way for improved patient care and enhanced collaboration between medical professionals and advanced technologies.

1.1 PROBLEM STATEMENT

The diagnosis of brain tumors plays a pivotal role in determining treatment strategies and patient outcomes. Magnetic Resonance Imaging (MRI) scans are widely used for the assessment of brain health, including the detection of tumors. However, manual interpretation of these complex images can be time-consuming, prone to human error, and reliant on specialized medical expertise.

The challenge lies in developing a reliable and efficient solution that automates the process of brain tumor detection in MRI scans. This solution should cater to both medical professionals seeking accurate and timely diagnoses and patients seeking prompt answers about their health status. It must address key problems such as:

- Time-intensive diagnosis
- Subjectivity and Variability
- Resource Constraints
- Patient empowerment

1.2 AIM

The aim of this project is to develop a deep learning based system that is; the Convolutional Neural Network (CNN) with improved accuracy for detection and prediction of brain tumor diseases. The Brain Tumor Detection System aims to address these problems by leveraging advanced image processing techniques and machine learning algorithms

1.3 SPECIFIC OBJECTIVES

- Develop a robust Convolutional Neural Network (CNN) model capable of accurately detecting brain tumors from medical images.
- Create a diverse and well-annotated dataset of brain tumor images to train and evaluate the CNN model's performance.
- Explore various preprocessing techniques, including resizing, normalization, and data augmentation, to enhance the quality and diversity of the dataset.
- Optimize hyperparameters such as learning rate, batch size, and layer configurations to achieve optimal model convergence and accuracy.
- Implement a user-friendly interface that allows medical professionals to interact with the model's predictions and integrate it into their clinical workflows.

1.4 PROJECT SCOPE

The scope of the project includes the development of a user-friendly web interface, the implementation of an advanced tumor detection algorithm, and the integration of features that enhance user experience and medical collaboration. The application will encompass the following key aspects:

1. **MRI Scan Upload and Processing:**
    - Enable users to upload MRI scans in various common formats.
    - Develop an algorithm to preprocess and analyze uploaded scans for tumor detection.
2. **Automated Tumor Detection:**
    - Implement a deep learning algorithm for accurate tumor detection within MRI scans.

.

The project's scope is centered on creating a reliable and efficient brain tumor detection web application that caters to both medical professionals and patients. While the application aims to streamline and enhance the tumor detection process, it does not intend to replace expert medical judgment. Instead, it serves as a valuable tool to aid medical professionals in making informed decisions and empowering patients with timely information about their health.

1.5 PROJECT JUSTIFICATION

The development of the Brain Tumor Detection Web Application is driven by the imperative to address significant challenges in the field of medical diagnostics, particularly in the realm of brain tumor detection using MRI scans. The following factors underscore the importance and relevance of this project:

Timely and Accurate Diagnosis:

Early detection of brain tumors is critical for effective treatment and improved patient outcomes.

1.6 PROJECT MOTIVATION

Behind the development of the Brain Tumor Detection System is rooted in it; the desire to address critical challenges in brain tumor diagnosis, enhance medical care, and leverage technology to improve patient outcomes. The following factors serve as the driving force for embarking on this innovative project:

Medical Diagnostic Advancements:

The fusion of medical diagnostics with cutting-edge technology opens new avenues for accurate and efficient disease detection.

1.7 PROJECT BENEFICIARIES

The Brain Tumor Detection Web Application is poised to benefit a diverse range of stakeholders, all of whom stand to gain from its innovative approach to brain tumor diagnosis and patient care. The beneficiaries of this project include:

1. **Medical Professionals:**
    - Radiologists, neurologists, and oncologists can utilize the application to enhance their diagnostic accuracy and efficiency, leading to better treatment planning and patient care.

- Medical experts in regions with limited resources can leverage the application to access advanced diagnostic capabilities, bridging the gap between medical expertise and geographical constraints.

2. **Patients and Their Families:**
   - Patients undergoing MRI scans can receive timely information about potential brain tumors, empowering them to make informed decisions about their health and treatment options.
   - Families of patients can gain reassurance and understanding through clear visualizations and reports provided by the application.
3. **Healthcare Institutions:**
   - Hospitals and medical centers can incorporate the application into their diagnostic workflows to streamline processes, reduce delays, and optimize resource utilization.
   - The application's collaborative features can facilitate interdisciplinary discussions among medical teams, leading to more comprehensive and effective patient care.
4. **Medical Education and Research Community:**
   - Medical students can use the application as a valuable educational tool to learn about medical image analysis and deep learning techniques.
   - Researchers in the field of medical imaging can explore the potential of advanced algorithms for improving brain tumor detection accuracy.

1.8 PROJECT LIMITATIONS

While the Brain Tumor Detection Web Application aims to provide innovative solutions for brain tumor diagnosis, there are certain limitations and challenges that should be considered:

1. **Dependency on Quality of Input Scans:**

    - The accuracy of tumor detection is dependent on the quality of the input MRI scans. Low-resolution or poorly captured scans may affect the performance of the detection algorithm.

2. **Limited to Detection, Not Diagnosis:**

    - The application provides automated tumor detection, but it does not replace the need for expert medical diagnosis. Detected regions should be reviewed and confirmed by medical professionals.

3. **Variability in Tumor Characteristics:**

    - Brain tumors can have varying sizes, shapes, and appearances. The application's performance may vary based on the diversity of tumor characteristics.

4. **False Positives and False Negatives:**

    - Automated algorithms can produce false positives (detecting tumors where none exist) and false negatives (missing actual tumors). Regular updates and refinement of the algorithm are necessary to minimize these errors.

5. **Training Data Bias:**

    - The machine learning algorithm's accuracy depends on the quality and diversity of the training dataset. A biased or unrepresentative dataset can lead to biased results.

6. **Interpretable Deep Learning Models:**

- Deep learning models, while powerful, can lack interpretability. Understanding the rationale behind predictions can be challenging, especially for medical professionals seeking diagnostic insights.

7. **Data Privacy and Security Concerns:**

    - Handling sensitive medical data requires stringent data privacy and security measures. Ensuring compliance with regulations like HIPAA is crucial to maintaining user trust.

8. **Dependency on Computational Resources:**

    - Machine learning algorithms, particularly deep learning models, demand substantial computational resources for training and inference. Access to sufficient computing power is essential.

9. **User Experience and Usability:**

    - The success of the application depends on its usability and user experience. A complex or unintuitive user interface can hinder adoption and hinder effective usage.

10. **Technical Expertise for Maintenance:**

    - Ongoing maintenance, bug fixing, and updates require technical expertise. A dedicated team is needed to address technical issues that may arise.

11. **Algorithm Generalization:**

    - The algorithm's performance might vary across different datasets and populations. Fine-tuning and retraining may be necessary to ensure generalizability.

12. **Internet Connectivity Requirement:**

- Users must have reliable internet connectivity to access and use the web application. Limited or unstable internet access may impact usability.

13. **Regulatory Approval and Medical Validation:**

    - Before deployment for clinical use, the application may require regulatory approvals and rigorous validation to ensure its safety, accuracy, and reliability.

14. **Ethical Considerations:**

    - Ethical considerations regarding patient consent, data usage, and potential biases in the algorithm's predictions should be thoroughly addressed.

Despite these limitations, the Brain Tumor Detection System aims to provide valuable support to medical professionals and patients in the field of brain tumor detection. Regular updates, continuous monitoring, and collaboration with healthcare experts can help mitigate these challenges and enhance the application's performance over time.

CHAPTER 2 - REVIEW OF RELATED SYSTEMS

The following is a review of similar systems to Brain Tumour detection web application

2.1 Overview of System 1: Automated Glioma Detection System

This system focuses on detecting gliomas, a type of brain tumour. It employs feature extraction and classification techniques to analyze MRI scans and identify tumour presence.

2.1.1 Good features

- . Focuses on a specific tumor type (gliomas), allowing for specialized detection.
- Incorporates both feature extraction and classification for comprehensive analysis.

2.1.2 Challenges

- Limited to the detection of gliomas, potentially missing other tumor types.
- Feature extraction and classification pipelines can be computationally intensive.

2.2 Overview of System 2: U-Net Architecture for Image Segmentation

U-Net is a popular architecture for medical image segmentation, including brain tumor detection. Its design allows for precise localization of tumor regions.

2.2.1. Good features

- U-Net's design is particularly well-suited for accurate segmentation tasks.
- Can effectively capture fine details and localize tumor regions.

2.2.2 Challenges

- May struggle with irregular or overlapping tumor shapes.
- Requires thorough hyperparameter tuning for optimal performance.

2.3 EXISTING SYSTEM AND PROPOSED WORK OVERFLOW

2.3.1 OVERVIEW OF EXISTING SYSTEM

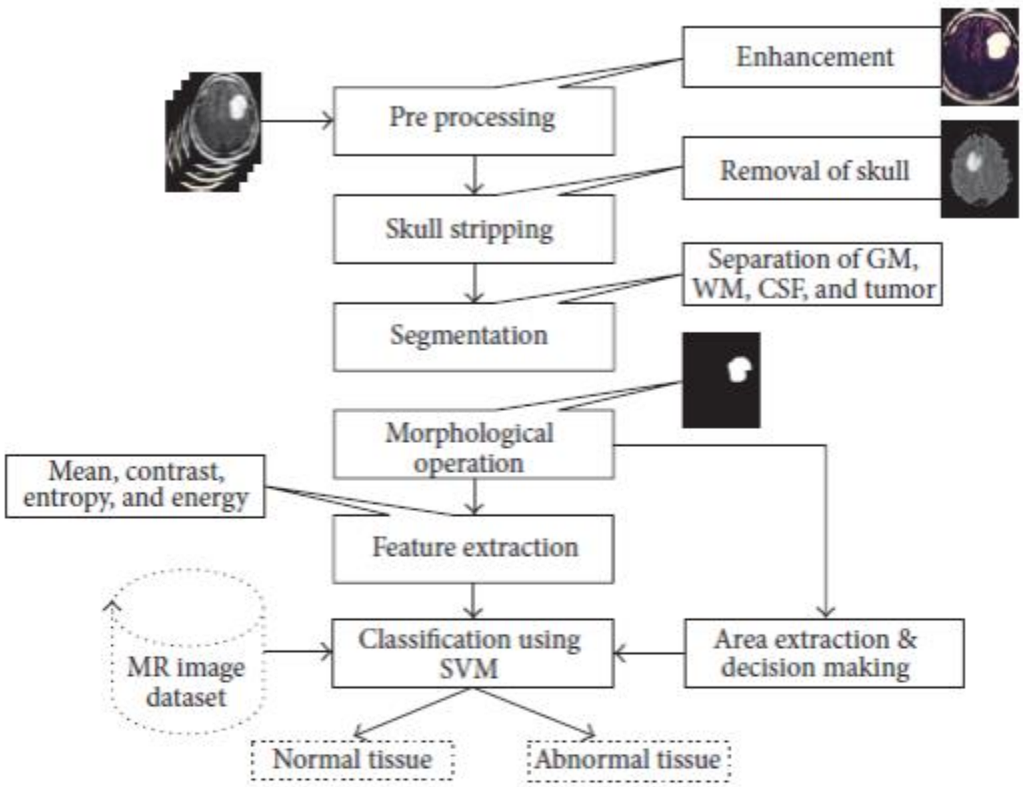


Figure 2.1 Diagram overview of existing system

- In the first stage, there is a computer-based procedures to detect tumor blocks and classify the type of tumor using Artificial Neural Network Algorithm for MRI images of different patients.

- The second stage involves the use of different image processing techniques such as histogram equalization, image segmentation, image enhancement, morphological operations and feature extraction are used for brain tumor detection in the MRI images for the cancer-affected patients.

- This work is introduced as one automatic brain tumor detection method to increase the accuracy and decrease the diagnosis time.

- **Image Preprocessing**: As input for this system is MRI, scanned image and it contain noise. Therefore, our first aim is to remove noise from input image. As explained in system flow we are using high pass filter for noise removal and preprocessing.

- **Segmentation**: Region growing is the simple region-based image segmentation technique. It is also classified as a pixel based image segmentation technique since it is involve the selection of initial seed points.

- **Morphological operation**: The morphological operation is used for the extraction of boundary areas of the brain images. This operation is only rearranging the relative order of pixel value, not mathematical value, so it is suitable for only binary images. Dilation and erosion is basic operation of morphology. Dilation is add pixels to the boundary region of the object, while erosion is remove the pixels from the boundary region of the objects.

- **Feature Extraction**: The feature extraction is used for edge detection of the images. It is the process of collecting higher level information of image such as shape, texture, color, and contrast.

- **Connected component labeling**: After recognizing connected components of an image, every set of connected pixels having same gray-level values are assigned the same unique region label.

• **Tumor Identification**: In this phase, we are having dataset previously collected brain MRIs from which we are extracting features. Knowledge base is created for comparison.

2.3.2 PROPOSED WORKFLOW

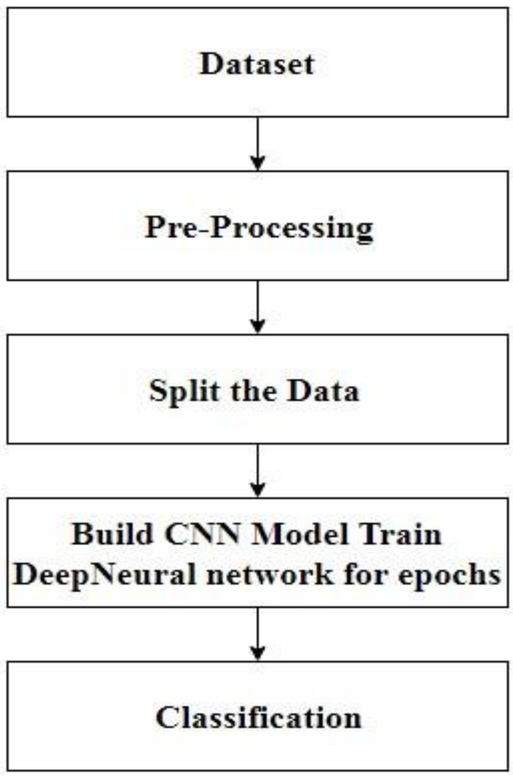


Figure 2.2 Proposed workflow

The proposed system has mainly five modules; Dataset, Pre-processing, Split the data, Build CNN model train Deep Neural network for epochs, and classification. In dataset we can take multiple MRI images and take one as input image. In pre-processing image to encoded the label and resize the image. In split the data we set the image as 80% Training Data and 20% Testing Data. Then build CNN model train deep neural network for epochs. Then classified the image as yes or no if tumor is positive then it returns yes and the tumor is negative the it returns no.

2.3.3 WORKING OF CNN MODEL

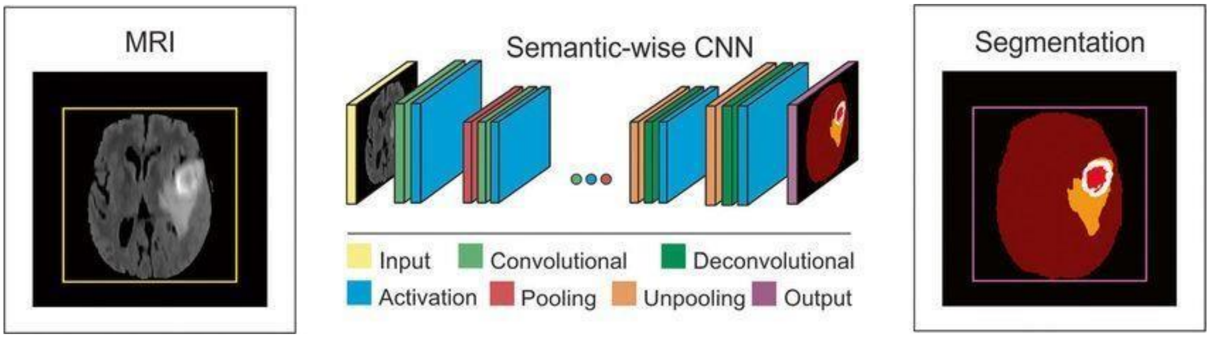


Figure 2.3 Representation of working CNN model

- Layers of CNN model
  - Convolution 2D
  - MAXpooling 2D
  - Dropout
  - Flatten
  - Dense
  - Activation

- **Convolution 2D:** In the Convolution 2D extract the featured from input image. It given the output in matrix form.

- **MAX Poolig2D:** In the MAX polling 2D it take the largest element from rectified feature map.

- **Dropout:** Dropout is randomly selected neurons are ignored during training.

- **Flatten:** Flatten feed output into fully connected layer. It gives data in list form.

- **Dense:** A Linear operation in which every input is connected to every output by weight. It followed by nonlinear activation function.

- **Activation:** It used Sigmoid function and predict the probability 0 and 1.
- In the compile model we used binary cross entropy because we have two layers 0 and 1.

- We used Adam optimizer in compile model.

Adam:-Adaptive moment estimation. It used for non convex optimization problem like straight forward to implement.

  - Computationally efficient.
  - Little memory requirement

2.3.4 PROPOSED SYSTEM FEATURES

1. **Upload and Preprocess MRI Scans:**
   - Enable users to upload MRI scans for analysis.
   - Preprocess uploaded scans by normalizing intensities, resizing, and applying data augmentation techniques.
2. **Deep Learning Model Integration:**
   - Integrate a pre-trained or custom-designed deep learning model, such as a CNN, for tumor detection.
   - Utilize the model to analyze preprocessed MRI scans and identify potential tumor regions.

3. **Tumor Detection and Visualization:**

- Process MRI scans through the deep learning model to detect tumor regions.

- Generate visualizations that highlight detected tumor areas within the scans

4. **Simple User Interface:**

- Design a simple web-based interface for model deployment to enable user interaction

- Provide options for uploading scans and viewing detection feedback.

2.4 DEVELOPMENT TOOLS

- **Python:** Python was the language of selection for this project. This was a straightforward call for many reasons.

  - Python as a language has a vast community behind it. Any problems which may be faced is simply resolved with a visit to Stack Overflow. Python is among the foremost standard language on the positioning that makes it very likely there will be straight answer to any question
  - Python has an abundance of powerful tools prepared for scientific computing Packages like NumPy, Pandas and SciPy area unit freely available and well documented. Packages like these will dramatically scale back, and change the code required to write a given program. This makes iteration fast.

- **Jupyter Notebook:** The Jupyter Notebook is an open-source web application that enables you to make and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more.

- **Noise Removal and Sharpening:** Unwanted data of element are remove using filter and image Can be sharpen and black and white gray scale image is used as a input.

Deep Learning Frameworks:

- **TensorFlow:** An open-source machine learning framework developed by Google that supports deep learning tasks.

  .
- **Keras:** A high-level neural networks API that can run on top of TensorFlow, Theano, or Microsoft Cognitive Toolkit.

Web Development**:**

- **HTML, CSS, JavaScript:** For building the user interface and designing the web application.

- **Web Frameworks:** Flask or Django (Python) for building the backend of the web application.

IDEs (Integrated Development Environments):

- **PyCharm:** A popular IDE for Python development.
- **Data Preprocessing and Visualization: OpenCV-**For image processing tasks.

2.5 COMPONENT DESIGN AND COMPONENT DESCRIPTIONS

Each component of the architecture contributes to the overall functionality of the "Brain Tumor Detection System." Their seamless integration creates a powerful diagnostic tool that combines advanced technology with medical expertise, ultimately aiming to improve the accuracy, speed, and effectiveness of brain tumor detection in medical MRI scans.

Below is how each component of the architecture of the "Brain Tumor Detection System " works:

- **Preprocessing Module:** The preprocessing module is responsible for preparing the uploaded MRI scans for analysis. It includes normalization techniques to standardize the pixel intensities and ensure consistent data. Additionally, the module handles image resizing to ensure that the input images are of a uniform size suitable for the deep learning model. Data augmentation techniques might also be applied to increase the diversity of training data, improving the model's robustness.

- **Deep Learning Model:** At the core of the system is the deep learning model, typically a Convolutional Neural Network (CNN). The model is trained on a diverse dataset of labeled MRI scans. During training, the CNN learns to identify patterns and features indicative of brain tumor regions. Once trained, the model is capable of processing preprocessed MRI scans and generating pixel-level predictions, highlighting potential tumor regions.

- **Tumor Detection and Localization:** When an MRI scan is passed through the trained deep learning model, the model generates predictions that indicate potential tumor regions. These predictions consist of pixel-level labels, effectively outlining the areas in the image where tumors are suspected. This step involves convolutional layers for feature extraction and pooling layers for down-sampling, enabling the model to capture relevant patterns.

- **Visualization and Interpretation:** The system generates visualizations that overlay the model's predictions onto the original MRI scans. These visualizations provide medical professionals with an intuitive representation of the detected tumor regions. The visualizations aid in interpretation and decision-making, allowing experts to validate and collaborate on the findings.
- **Collaboration and Validation:** The system supports collaboration among medical professionals by enabling them to review and validate the detected tumor regions. Medical experts can interact with the visualizations, discuss findings, and provide

insights. Collaboration tools enhance the reliability of diagnoses and contribute to a collective understanding of potential tumor regions.

- **Continuous Improvement:** The system is designed to support continuous improvement over time. Mechanisms are in place to update and retrain the deep learning model as new data becomes available. Feedback from medical professionals can be incorporated to refine the model's accuracy and performance.

2.5.1 COMPONENT DESCRIPTION ON THE ALGORITHM OF THE MODEL CODE FOR THE SYSTEM

Algorithm and Components of Brain Tumor Detection System:

The brain tumor detection system is built using a Convolutional Neural Network (CNN), a type of deep learning model specialized for image analysis tasks. The primary goal of this system is to accurately classify brain tumor images as either containing a tumor or being tumor-free.

1. Data Preprocessing:

- The system starts by loading and preprocessing the dataset, which consists of brain tumor images. Images are loaded, resized to a fixed input size, and converted into arrays of pixel values.
- Labels are assigned to each image, indicating whether it contains a tumor or not.

2. Model Architecture:

- The model architecture is constructed using the Keras Sequential API.
- The architecture consists of several layers, including Conv2D, Activation, MaxPooling2D, Flatten, Dense, Dropout, and Activation.

3. Feature Extraction:

- The model's Conv2D layers perform feature extraction. Each Conv2D layer applies convolutional filters to the input images, extracting patterns and features.
- ReLU (Rectified Linear Unit) activation functions introduce non-linearity and allow the model to learn complex relationships between features.
- MaxPooling2D layers downsample the feature maps, retaining important information while reducing spatial dimensions.

4. Dense Layers and Classification:

- After feature extraction, the model flattens the feature maps and passes them through Dense (fully connected) layers.
- These dense layers help in learning higher-level representations of the extracted features.
- Dropout layers are introduced to prevent overfitting by randomly deactivating neurons during training.
- The final dense layer produces the model's output logits.

5. Activation Function and Output:

- The final dense layer is followed by a sigmoid activation function.
- The sigmoid activation squashes the output logits into the range [0, 1], representing the probability of the input image containing a tumor.

6. Loss Function and Optimization:

- The model uses binary cross-entropy loss, suitable for binary classification problems.
- The optimization algorithm used is Adam, a popular gradient-based optimizer that adapts learning rates during training.

7. Training and Validation:

- The model is trained on the training dataset with specified hyperparameters such as batch size and number of epochs.

- The validation dataset is used to monitor the model's performance during training and prevent overfitting.

8. Model Evaluation:

- Once training is complete, the model is evaluated on an independent test dataset to assess its generalization performance.
- Metrics such as accuracy and loss are calculated to measure the model's effectiveness.

9. Model Deployment:

- Once the model is trained and evaluated, it can be deployed for inference in real-world scenarios.
- The model can take new brain tumor images as input and classify them as containing a tumor or not.

The brain tumor detection system employs a Convolutional Neural Network to automatically learn and extract relevant features from brain tumor images. The architecture of the system, including Conv2D layers, ReLU activations, MaxPooling2D layers, and dense layers, collectively enables the model to identify intricate patterns in the images. The system's training process optimizes the model's parameters to make accurate predictions. The system's design and architecture empower it to perform brain tumor detection tasks with high accuracy, benefiting medical professionals and patients alike.

2.6 BENEFITS OF THE IMPLEMENTAITON OF THE PROPOSED SYSTEM

The implementation of the proposed "Brain Tumor Detection System Using Deep Learning Methods" offers a range of benefits to medical professionals, patients, and healthcare institutions. Here are some key benefits:

2.6.1 Benefits of the Proposed System

1. **Accurate and Early Detection:**
    - The deep learning-based system enhances the accuracy of brain tumor detection, aiding in early diagnosis and intervention.
    - Early detection can lead to improved treatment outcomes and enhanced patient survival rates.
2. **Efficiency and Time Savings:**
    - Automation of tumor detection reduces the time required for manual analysis of MRI scans.
    - Medical professionals can focus on interpreting results rather than spending excessive time on the detection process.
3. **Objective and Consistent Results:**
    - The system's automated approach reduces subjectivity, leading to consistent and objective detection outcomes.
    - Results are not influenced by individual variations in interpretation.
4. **Enhanced Diagnostic Process:**

- Medical professionals can leverage the system's insights to make informed decisions during diagnosis and treatment planning.
- The system complements medical expertise by providing additional diagnostic support.

5. **Collaborative Healthcare:**
    - The system's collaborative features enable medical experts to review and validate detected tumor regions.
    - Seamless collaboration enhances the quality of diagnosis and treatment recommendations.
6. **Educational Value:**
    - Researchers and medical students can gain insights into the application of deep learning in medical diagnostics.
    - The system serves as an educational tool for understanding technology's role in healthcare.
7. **Timely Intervention:**
    - The system's efficiency allows for prompt detection and timely intervention, minimizing the progression of tumors and potential complications.
8. **Accessibility and Convenience:**
    - The web-based user interface enables medical professionals to access the system from various locations and devices.
    - Remote access facilitates convenience and extends the reach of diagnostic capabilities.

9. **Reduced Workload and Stress:**

    - The automated tumor detection process reduces the workload on medical professionals, alleviating stress and burnout.

10. **Resource Optimization:**

    - Efficient analysis of MRI scans optimizes the use of healthcare resources, including medical equipment and personnel time.

11. **Quantitative Analysis:**

    - The system provides quantitative insights into the size, location, and characteristics of detected tumor regions.

    - This information aids in treatment planning and monitoring.

12. **Continuous Improvement:**

    - The system can be updated and refined based on new data and feedback, ensuring continuous improvement in accuracy and performance.

13. **Regulatory Compliance:**
    - The system adheres to data privacy regulations, ensuring that patient information is handled securely and ethically.

14. **Technological Innovation:**
    - Implementation of advanced deep learning techniques showcases the innovative potential of technology in healthcare.

CHAPTER 3 - METHODOLOGY

3.0 METHODOLOGY OVERVIEW

Brain tumor disease detection depends on extracting features from chronic pictures. The many phases would be depicted on a flowchart, allowing a better understanding of the procedures. This approach will cover a research gap in which a separate model has been built for each chronic image of the disease.

The motivation is that training, implementing, and maintaining an expert system for brain tumor disease in large networks, especially deep learning techniques, might quickly become problematic. This study focuses on a more recent benchmark dataset that has recently been available to researchers. On MATLAB R2019b, the Kaggle Dataset [48] will be used with a convolutional neural network. This convolutional neural network will be the same size and number of layers as the previous one.

Their threshold formula will be used, but it will be modified by adding a parameter. This parameter is subjected to optimization. Some specifics, such as the choice of activation function and optimization algorithm, are unique to this work. In addition, this research will use deep learning to forecast the accuracy of feature selection parameters using a feature selection mechanism. This research will allow for a more accurate evaluation of convolutional neural network method accuracy on smaller feature subsets and a more balanced comparison of illness detection and classification for brain tumors using a more sophisticated deep learning method.

3.1 REQUIREMENT SPECIFICATIONS

3.1.1 HARDWARE AND SOFTWARE SPECIFICATION

HARDWARE REQUIREMENTS

● Hard Disk : 500GB and Above

● RAM : 4GB and Above

● Processor : Core I3 and Above

SOFTWARE REQUIREMENTS

✔ Operating System : Windows 7, 8, 10 (64 bit)

✔ Software : Python

✔ Tools : Anaconda (Jupyter Notebook IDE

3.2 STAKE HOLDERS OF THE SYSTEM.

Stakeholders are individuals or groups who have an interest, influence, or investment in a project or system. In a brain tumor detection system using a trained CNN model, the stakeholders include:

1. **Patients and Their Families:**

Patients diagnosed with brain tumors and their families are primary stakeholders. They are directly affected by the accuracy and reliability of the system's tumor detection. Accurate results can lead to timely medical interventions, while false negatives or positives could have significant consequences.

2. **Medical Professionals:**

Radiologists, neurologists, and medical practitioners are essential stakeholders. They use the system's predictions to assist in diagnosis and treatment planning. The system's accuracy and ease of use impact their workflow and decisions.

3. **Healthcare Institutions and Hospitals:**

Hospitals and healthcare institutions are invested in the system's performance as it can enhance their diagnostic capabilities. Reliable tumor detection could lead to improved patient care, while inaccurate results could affect their reputation and patient outcomes.

4. **Software Developers and Data Scientists:**

The individuals responsible for developing, maintaining, and optimizing the system are stakeholders. They are invested in the system's technical aspects, including its efficiency, accuracy, maintainability, and scalability.

5. **Regulatory Authorities:**

Regulatory bodies overseeing medical software and devices are stakeholders. They ensure that the system complies with medical standards, safety regulations, and ethical guidelines.

6. **Researchers and Academia:**

Researchers in the field of medical imaging and machine learning might be stakeholders. They could benefit from the system's insights, contribute to its improvement, or study its impact on medical practices

3.3 FUNCTIONAL REQUIREMENTS.

Image Input and Preprocessing:

- The system should allow users to input medical images (MRI scans) in supported formats.
- The system should ensure the image is preprocessed to ensure consistent size, resolution, and normalization for accurate model predictions.

Tumor Detection and Prediction:

- The system should utilize the trained CNN model to predict the presence of brain tumors in input images.
- Predictions should provide a confidence score or probability indicating the likelihood of tumor presence.

Accuracy Assessment:

- The system should calculate and display the accuracy of its predictions based on ground truth labels.

Visualization and Interpretation:

- The system should visually display the input image along with the prediction.

Support for Multiple Image Formats:

- The system should support various image formats commonly used in medical imaging, such as JPEG, and PNG.

3.4 UML DIAGRAMS

Unified Modeling Language (UML) is a universal modeling language that has been standardized in the field of software development. This standard is managed and created by the asset management team. It is used to define, visualize, modify, construct and document the artifacts of intensively developed object-oriented software systems

ARCHITECTURE DIAGRAM:

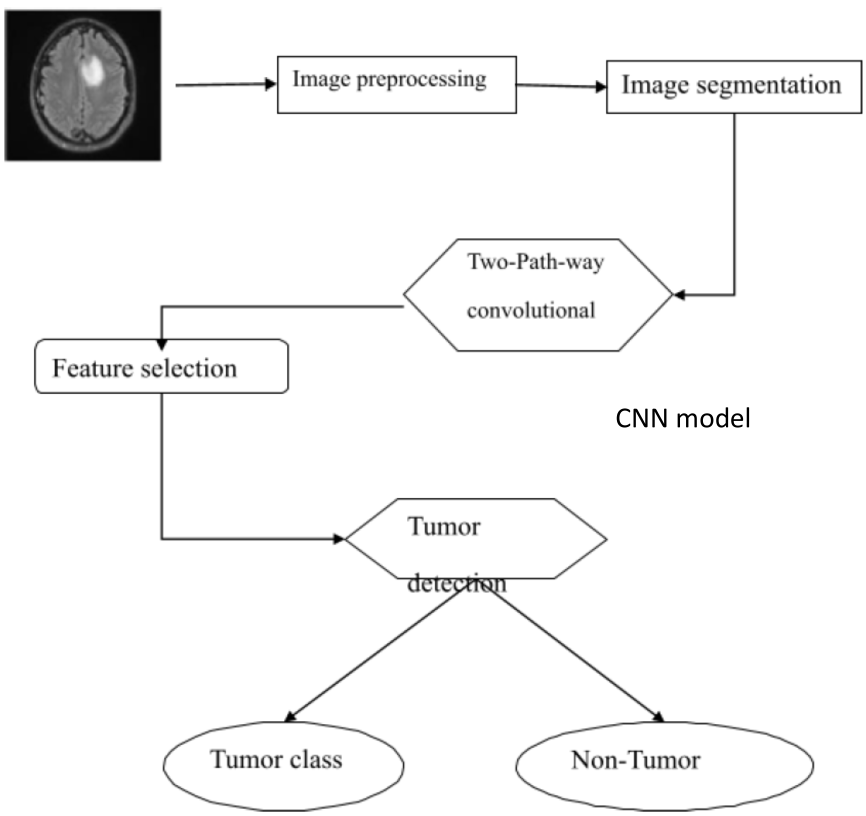


Figure 3.1 Architecture Diagram

3.4.1 SEQUENCE DIAGRAM:

A **s**equence diagram is an interaction diagram that shows how and in what order the processes interact**.** This is the construction of message sequence diagrams**,** sometimes called. event diagrams, event scenarios**,** and sequence diagram**s**.

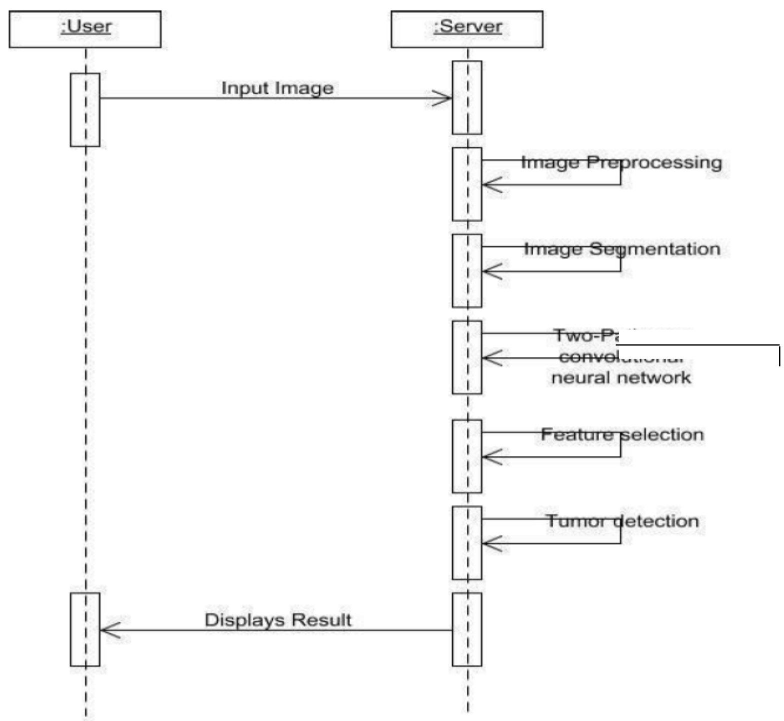


Fig 3.2 Sequence diagram

3.4.2 USE CASE DIAGRAM

Use case diagrams are used to graphically describe the functions provided by the system based on participants, their goals, and any dependencies between these use cases. The use case diagram consists of two parts:

Use case: A use case describes a series of actions through which the subject can be measured. and drawn as a horizontal ellipse.

Participants: Participants are individuals, organizations, or external systems that play a role in one or more interactions with the system.

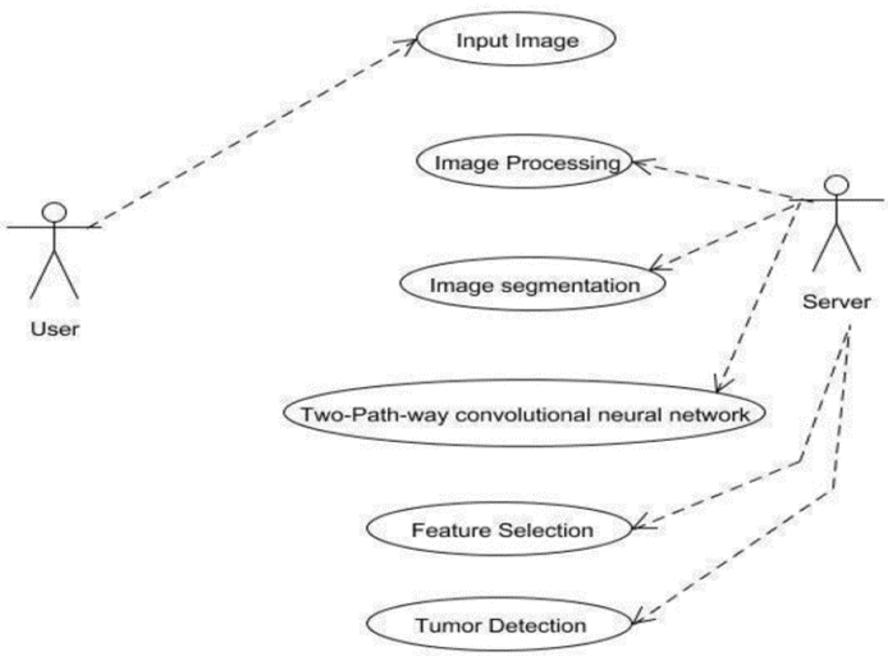


Fig 3.3 Use Case Diagram

3.4.3 ACTIVITY DIAGRAM:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

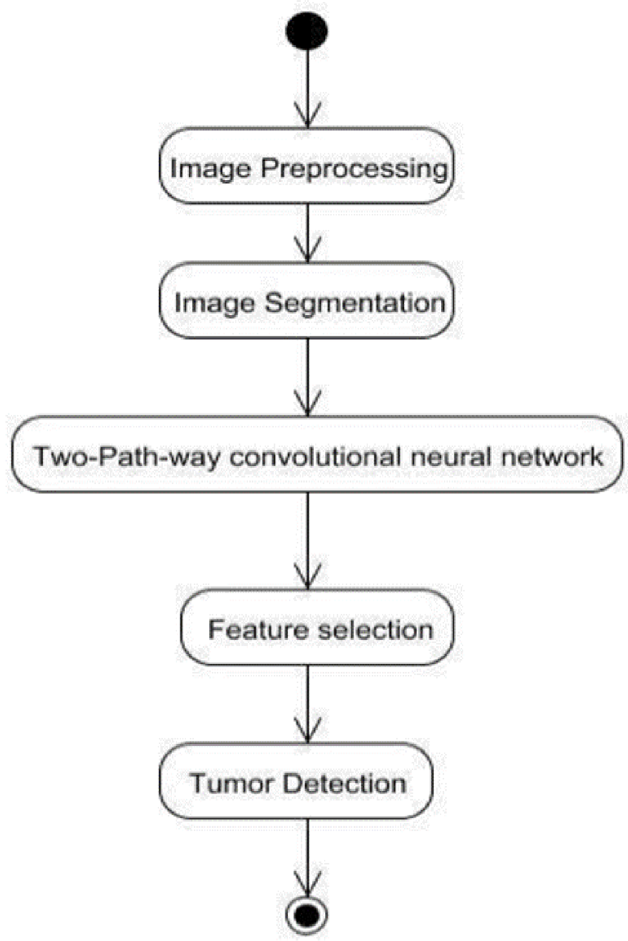
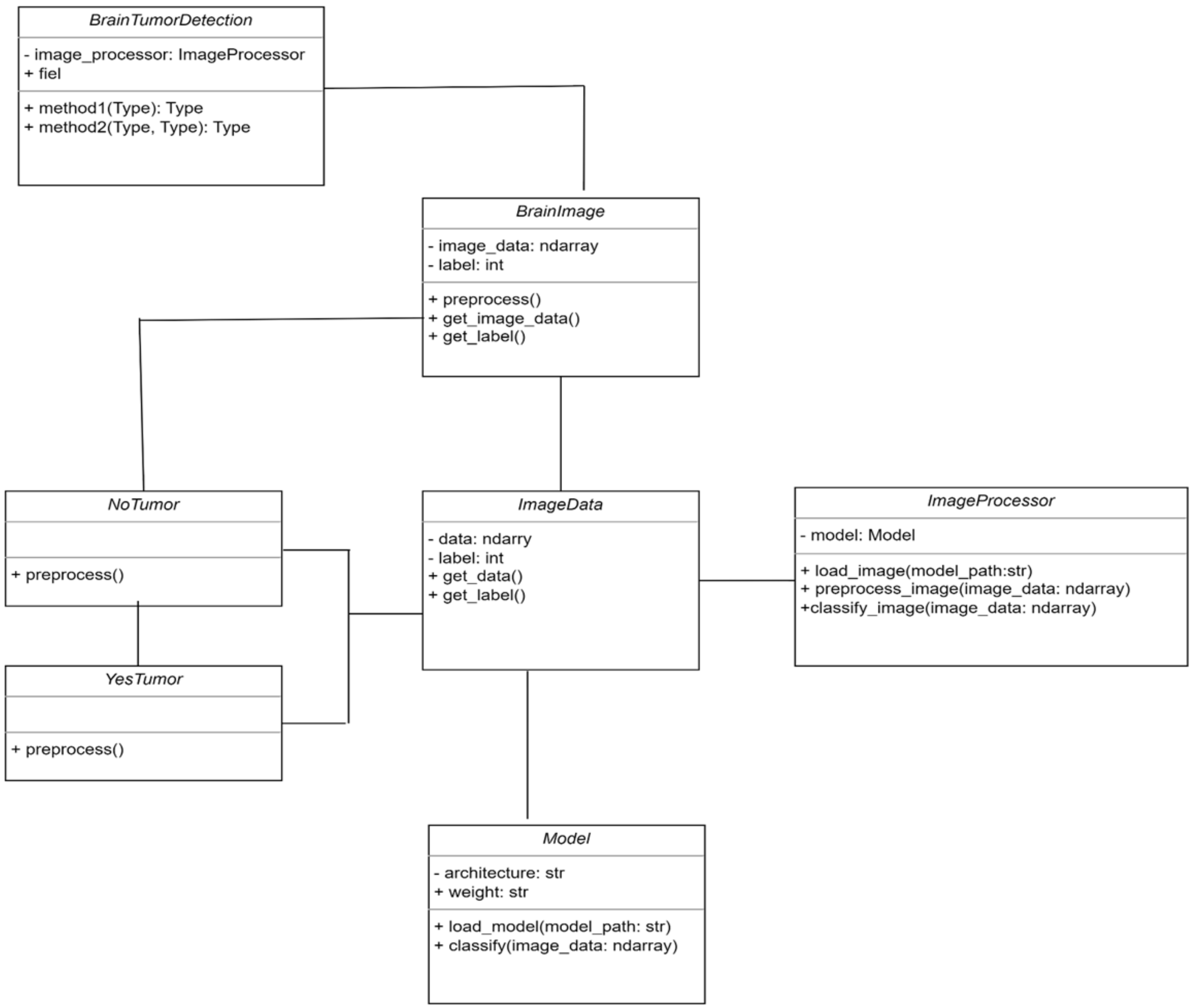


Fig 3.4 Activity diagram

3.4.4 CLASS DIAGRAM

A class diagram is a type of diagram in software engineering that visually represents the structure of a system or application in terms of classes, their attributes, methods, and the relationships between classes. It is a fundamental component of the Unified Modeling Language (UML), which is used to model and document software systems.

Fig 3.5 class diagram



3.4.5 COLLABORATION DIAGRAM:

UML collaboration diagram illustrates the relationship and interaction between software objects. that use cases, system usage contracts and domain models already exist. The collaboration diagram shows the messages sent between the class and the object

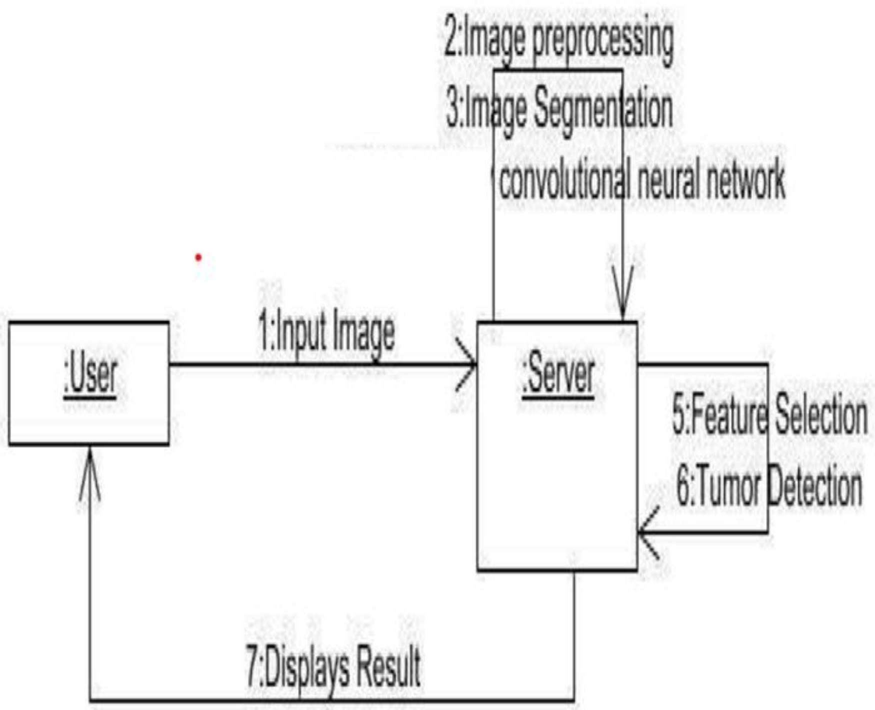


Fig 3.6 Collaboration diagram

3. NONFUNCTIONAL REQUIREMENTS

1. **Performance:**
    - **Response Time:** Specifies the maximum time a system should take to respond to a user request.
    - **Latency:** Defines the acceptable delay in processing or communication.
2. **Reliability:**
    - **Availability:** Specifies the percentage of time the system should be operational and available for use.
    - **Fault Tolerance:** Describes the system's ability to continue functioning despite failures.
3. **Usability:**
    - **User Interface:** Describes the user interface's ease of use, intuitiveness, and consistency.
    - **User Documentation:** Defines the availability and comprehensiveness of user manuals and guides.

.

4. **Maintainability:**
    - **Extensibility:** Describes how easily the system can be extended with new features or functionalities.
    - **Readability and Documentation:** Specifies the code's readability and the availability of documentation for developers.
5. **Performance:**

- **Response Time:** Specifies the maximum time a system should take to respond to a user request.

3.5 PROJECT METHODS.

The brain tumor detection using a trained model and input images leans more toward a plan-driven approach rather than an agile approach because of the following:

1. Sequential Workflow: The code follows a sequential and structured workflow with well-defined steps. Each step must be completed before moving to the next one. There isn't room for parallel development or iterations, which aligns with plan-driven methodologies.

2. Fixed Process Flow: The code follows a fixed process flow, where steps like loading the model, preprocessing images, making predictions, and calculating accuracy are carried out in a specific order. Agile methods typically allow more flexibility to reorder or adapt steps based on feedback.

3. Emphasis on Accuracy Calculation: The code places a significant emphasis on calculating accuracy, which is a characteristic of plan-driven approaches. Accuracy calculation is treated as a distinct step rather than an ongoing iterative improvement process.

4. Detailed Conceptual Design: The code includes a detailed conceptual design that outlines specific components, interactions, and responsibilities. Agile methodologies often rely on lightweight documentation and may favor communication over extensive upfront design.

5. Predictable Timeline: The code execution follows a predictable timeline without iterations or incremental updates. Agile methodologies allow for iterative development, enabling the incorporation of feedback and adjustments in real-time.

6. Limited Collaboration Emphasis: While the code might involve collaboration between developers, it doesn't emphasize continuous collaboration with stakeholders or frequent iterations based on their feedback. Agile methodologies encourage ongoing interaction and adaptation based on user input.

7. Quality Control and Validation: The code integrates quality control aspects, such as calculating accuracy and comparing predictions with ground truth. These aspects are typical of plan-driven approaches that prioritize accuracy verification.

8. Absence of Continuous Improvement: Agile methodologies stress continuous improvement through feedback loops, retrospectives, and iterative enhancements. The code lacks provisions for ongoing improvements based on user feedback or changing requirements.

10. Minimal Adaptability: The code doesn't readily accommodate changes in real-time. Agile methodologies allow for more adaptability to changes, even during development, based on user needs.

3.6 SOFTWARE PROCESS MODELS.

For CNN brain tumor detection, several software process models can be considered. Each has its own advantages and suitability based on the project's scope, requirements, and team dynamics. Below are brief descriptions of a few relevant software process models:

1. **Waterfall Model:** The Waterfall model is a linear and sequential approach. It involves distinct phases like requirements analysis, design, implementation, testing, and deployment. Each phase is completed before moving to the next. This model can be suitable if the project's requirements are well-defined and unlikely to change significantly.

2. **Iterative Model:** The Iterative model involves repeating cycles of development, where each cycle includes phases like planning, requirements, design, implementation, and testing. At the end of each cycle, a working prototype is delivered. This approach accommodates incremental improvements and allows for changes in requirements.

3. **Agile Model :** Agile methodologies prioritize flexibility, collaboration, and iterative development. Teams work in short iterations (sprints) to deliver small increments of functionality. The project evolves based on continuous feedback from stakeholders. This approach is suitable when requirements may evolve and when collaboration is crucial.

4. **V-Model (Validation and Verification Model):** The V-Model aligns testing phases with development phases. Each development phase has a corresponding testing phase, ensuring that each requirement is thoroughly tested. This model is particularly useful

when quality assurance and verification are critical, such as in medical applications like brain tumor detection.

5. **Spiral Model:** The Spiral model combines iterative development with risk assessment. It involves cycles of planning, risk analysis, engineering, and evaluation. It's particularly useful when there's a need to manage uncertainties and risks associated with developing advanced systems like CNN-based medical applications.

6. **Hybrid Models:** Hybrid models combine elements of multiple process models to suit specific project needs. For CNN brain tumor detection, a hybrid model could combine aspects of Waterfall, Iterative, and Agile models to balance the need for accuracy, iteration, and adaptability.

7. **RAD Model (Rapid Application Development):** The RAD model emphasizes rapid prototyping and quick development cycles. It's useful when time-to-market is crucial and when the project requires frequent user feedback and validation.

3.6.1 WATERFALL MODEL

The Waterfall Model is a linear and sequential approach to software development. It divides the development process into distinct phases, with each phase building upon the outcomes of the previous one. The phases are completed in a linear order, and there is minimal overlap or iteration between them. The phases are as follows:

1. **Loading the Model:** The code begins by loading the pre-trained Convolutional Neural Network (CNN) model. This phase is analogous to the initial "Requirements Analysis" and "System Design" phases in the Waterfall Model, where the foundational components are set up.

2. **Preprocessing Images:** The next step involves preprocessing the input images, including resizing and normalization. This corresponds to the "Implementation" phase in the Waterfall Model, where the detailed coding and implementation of the system's functionality take place.

3. **Model Prediction:** The code then utilizes the loaded model to predict the presence of a brain tumor in the input image. This step aligns with the "Testing" phase in the Waterfall Model, where the developed system is rigorously tested to ensure its correctness.
4. **Calculating Accuracy:** After making predictions, the code calculates the accuracy of the system's predictions. This corresponds to the "Validation" phase in the Waterfall Model, where the system's performance is validated against predefined criteria.
5. **Completion and Delivery:** The code wraps up by saving the trained model and might involve other post-processing tasks. This finalization step aligns with the "Deployment" phase in the Waterfall Model, where the complete system is delivered to users or stakeholders.

3.7 DATASET DETAILS

Kaggle Dataset is concerned with the method and format in which data is often collected from the Kaggle repository. The data can be in either.JPEG or.PNG format. The performance of networks with identical architecture will be compared to isolate the training component. These are the pre-trained model, the network with only the last layer retrained using machine learning and the network with only the last layer retrained using machine learning. Their performances are presented depending on the level of occlusion applied to the instance they had to classify. Having a high cluttering factor also allows a large portion of the self-interference signal to remain and not be canceled. The dataset can be accessed from the link:
https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection

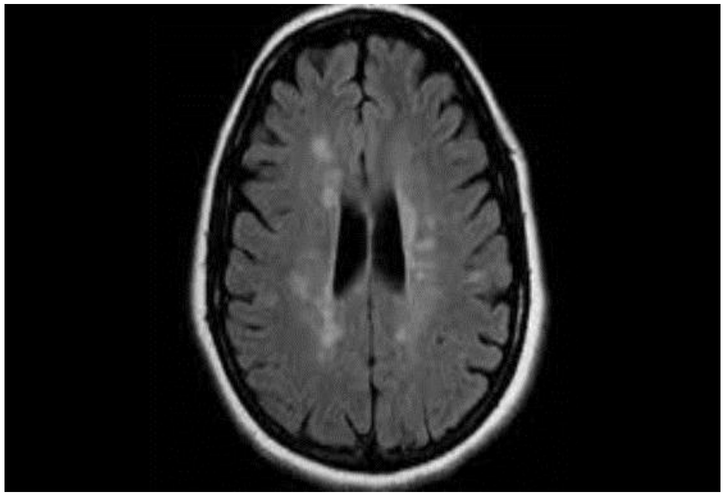


Fig 3.6 A sample of brain image being acquired from Kaggle.

3.7.1 CONVOLUTIONAL NEURAL NETWORK (CNN).

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects or objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand engineered, with enough training, ConvNet have the ability to learn these filters or characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cd over the entire visual area. The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first Layer is responsible for capturing the Low-level features such as edges, color, gradient orientation etc. With added layers, the architecture adapts to the High-Level features as well, giving us a network, which has the wholesome understanding of images in the dataset, similar to how we would. The proposed system has mainly five modules. Dataset, Pre-processing, Split the data, Build CNN model train Deep Neural network for epochs, and classification. In dataset we can take multiple MRI images and take one as input image. In pre-processing image to encoded the label and resize the image. In split the data we set the image as 80% Training Data and 20% Testing Data. Then build CNN model train deep neural network for epochs. Then classified the image as yes or no if tumor is positive then it returns yes and the tumor is negative the it returns no.

3.7.2 MAGNETIC RESONANCE IMAGING (MRI)

The MRI is a diagnostic tool used for analyzing and studying the human anatomy. The medical images acquired in various bands of the electromagnetic spectrum. The wide variety of sensors used for the acquisition of images and the physics behind them, make each modality suitable for a specific purpose. In MRI, the pictures are produced using a magnetic field, which is approximately 10,000 times stronger than the earth's magnetic field. The MRI produces more detailed images than other techniques, such as CT or ultrasound. The MRI also provides maps of anatomical structures with a high soft-tissue contrast. Basically, the magnetic resonance of hydrogen (1H) nuclei in water and lipid is measured by an MRI.

Layers of CNN model:

- Convolutional Layer (Convolution 2D)
- Pooling Layer (MAX Poolig2D)
- Dropout
- Flatten
- Dense
- Activation

Convolutional Layer: The Convolutional layer is the core block of the Convolutional Neural Network. It has some special properties. It does most of the computational heavy lifting. The CONV layer's parameters consist of a set of learn-able filters. Every filter is small spatially (along width and height) but extends through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume and dot products are computed between the entries of the filter and the input at any position. As the filters are slid over the width and height of the input volume, a 2-dimensional activation map will be produced that gives the responses of that filter at every spatial position. Intuitively, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation. These activation maps are stacked along the depth dimension and the output volume is produced.

Pooling Layer: Pooling layer is another building block of CNN. Usually, the pooling layer is placed after the convolutional layer. Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network, and hence to also control over fitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially. Pooling does not affect the depth dimension of the input volume.

- Max Pooling: It returns the maximum value from its rectangular neighborhood.

Dropout: Dropout is an approach to regularization in neural networks which helps reducing interdependent learning amongst the neurons. Dropout forces a neural network to learn more robust features that are useful in conjunction with many different random subsets of the other neurons. During the forward pass, Dropout temporarily removes some of the neurons. For example, if we fix dropout to 20%, then every 1 out of 5 neurons will be inactivated during the forward pass. During the backward pass, any weigh update will not be applied to these neurons.

Flatten Layer

After the pooling layer, a pooled feature map is obtained. Flatten layer is one of the essential layers after the pooling because we have to transform the whole matrix representing the input images into a single column vector and its imperative for processing. It is then fed to the Neural Network for the processing.

Dense Layer or Fully Connected Layer: In fully connected layer, every neuron is connected to its previous layer neuron like the neural network. Its activation is also computed by matrix multiplication with its weight followed by bias as like neural network. Usually, a fully connected layer is a column vector.

Activation Function: The Activation or Activation functions are used to introduce non-linearity to neural networks. It squashes the values in a smaller range. For example, a sigmoid activation function squashes values between a range 0 to 1. The Activation function used in the model is sigmoid function and ReLU.

Sigmoid: The Sigmoid function bounds the input value in between 0 to 1 range. For large positive numbers, it returns 1 and for large negative number, it returns 0.

ReLU (Rectified Linear Unit): ReLU refers to Rectified Linear Unit. It simply thresholds the input value to zero. For positive, it returns the number and for negative, it returns 0.

3.8 JUSTIFICATION OF CHOSEN MODEL.

Convolutional Neural Networks have a different architecture than other neural networks such as regular Neural Networks. Regular Neural Networks transform an input by putting it through a series of hidden layers. Every layer is made up of a set of neurons, where each layer is fully connected to all neurons in the layer before and where neurons in a single layer function completely independently and do not share any connections between themselves. Finally, there is a last fully connected layer, which is the output layer that represent the predictions. Regular Neural Networks do not scale well to full images. Convolutional Neural Networks are a bit different. First of all, the layers are organized in three dimensions: width, height, and depth. Further, the neurons in one layer do not connect to all the neurons in the next layer but only to a small region of it. Lastly, the final output will be reduced to a single vector of probability scores, organized along the depth dimension. Moreover, CNNs perform convolution operation in case of matrix multiplication.

DEVELOPMENT TOOLS.

Python:

Python is the programming language that we used to develop your brain tumor classification system using Convolutional Neural Networks (CNNs). Python's versatility, ease of use, and extensive libraries make it a popular choice for machine learning and deep learning tasks. Here's how Python contributed to our work:

1. **Libraries and Frameworks**: Python boasts powerful libraries and frameworks for machine learning, such as TensorFlow and Keras, which we utilized in our code. These libraries provide pre-built functions and classes for creating and training neural networks, reducing the complexity of implementing complex algorithms from scratch.

2. **Data Manipulation**: Python's extensive data manipulation libraries, like NumPy and Pandas, allowed us to efficiently handle, preprocess, and manipulate the image data.

3. **Visualizatio**n: Python has libraries like **Matplotlib** that enabled us to visualize data and results effectively. Visualization helps you gain insights into the performance of your model and present your findings.

4. **Integration and Deployment**: Python is versatile in integrating with other technologies and helped in easily deploying our trained models as part of a larger application or system. Python's integration capabilities make it feasible to use your model for real-world applications.

PyCharm

PyCharm is an integrated development environment (IDE) specifically designed for Python programming. It provides a range of features that can greatly assist in developing machine learning projects like our brain tumor classification system. Here's how PyCharm helped in our work:

1. **Code Editing and Productivity:** PyCharm offers advanced code editing features such as auto-completion, code suggestions, and error highlighting. These features improve coding speed and accuracy, helping you write code more efficiently**.**
2. **Integrated Debugger:** PyCharm's integrated debugger allows you to identify and fix errors and issues in your code step by step. This is crucial in machine learning projects where debugging complex neural network architectures is important.

Keras

Keras is a high-level neural network library written in Python that provides a user-friendly interface to build, train, and evaluate deep learning models. It abstracts the complexities of low-level libraries like TensorFlow , making it easier to develop neural network architectures like the brain tumor detection. Here's how Keras helped in our project:.

1. **Layer Abstraction:** Keras abstracts the implementation details of neural network layers. You could add convolutional, pooling, dense, and activation layers to your model with just a few lines of code.
2. **Activation Functions**: Keras offers a variety of activation functions that can be easily added to layers. Activation functions like ReLU, sigmoid, and softmax are crucial for introducing non-linearity into our model**.**
3. **Optimizers and Loss Functions:** Keras provides a range of optimizers (e.g., Adam, SGD) and loss functions (e.g., binary cross-entropy) that we used. These components are essential for training a neural network effectively.
4. **Model Compilation**: Keras simplifies model compilation by allowing you to specify the loss function, optimizer, and evaluation metrics in a single step. This makes the process of configuring the training process more straightforward.

5. **Training and Evaluation:** Keras streamlines the training process with the fit method. You can provide training and validation data, set the batch size and number of epochs, and monitor performance metrics during training.
6. **Model Saving and Loading:** Keras enabled us to save your trained models to disk and load them for further evaluation or deployment.

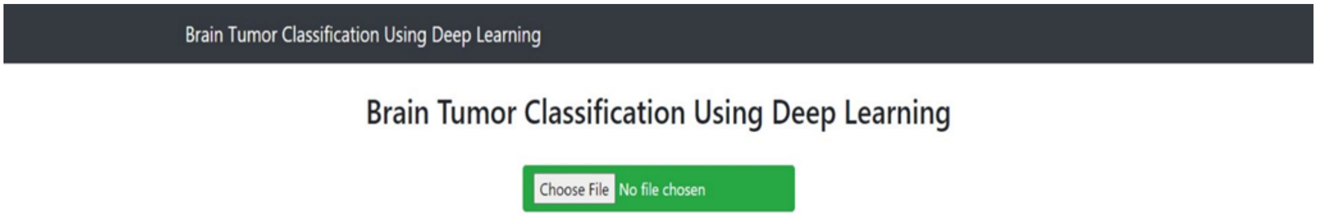
OpenCV:

OpenCV (Open-source computer vision) is a library of programming functions mainly aimed at real-time computer vision. In the code, OpenCV library played a crucial role in preprocessing and working with the image data. OpenCV is a versatile library widely used for various computer vision tasks, including image manipulation, processing, and analysis. Here's how OpenCV helped in the work:

1. **Loading and Reading Images: We used OpenCV's cv2.imread function to load and read image files from the specified directories. This function converts image data into a format that can be easily processed and analyzed.**
2. **Resizing Images: After reading the images, we used OpenCV's cv2.resize function to resize them to a uniform size (INPUT_SIZE). Resizing is important to ensure that all images have the same dimensions before feeding them into the neural network for training or prediction.**
3. **Converting Images to NumPy Arrays: OpenCV enabled us to convert images into NumPy arrays using the np.array function. This format is suitable for numerical operations and compatibility with neural network frameworks.**

3.9 UI DESIGN.

The CNN model was deployed in a web application using Flask. Below is the UI design of the web application.



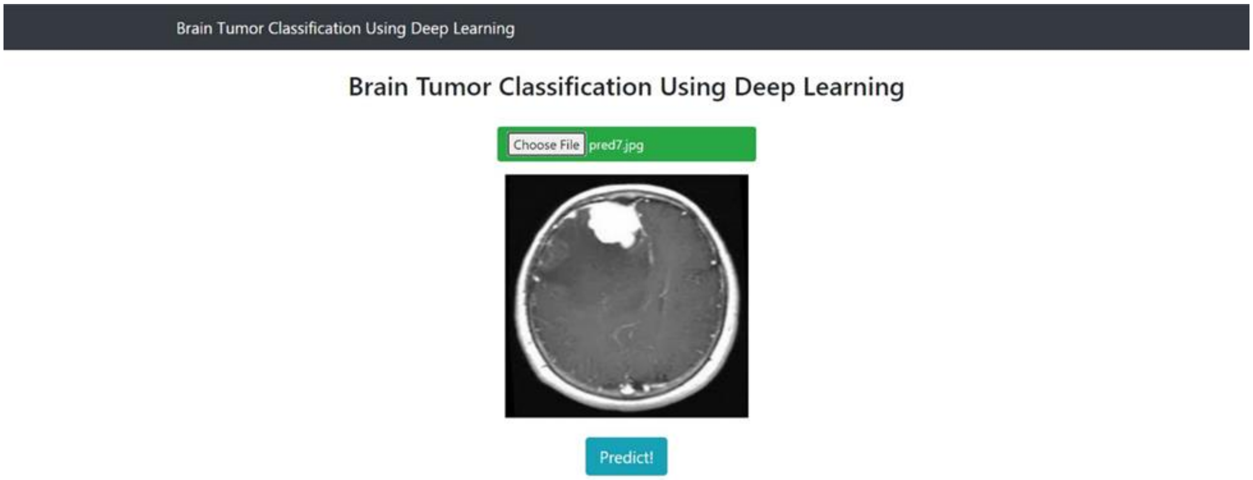


Fig 3.7 When Image is Uploaded

.

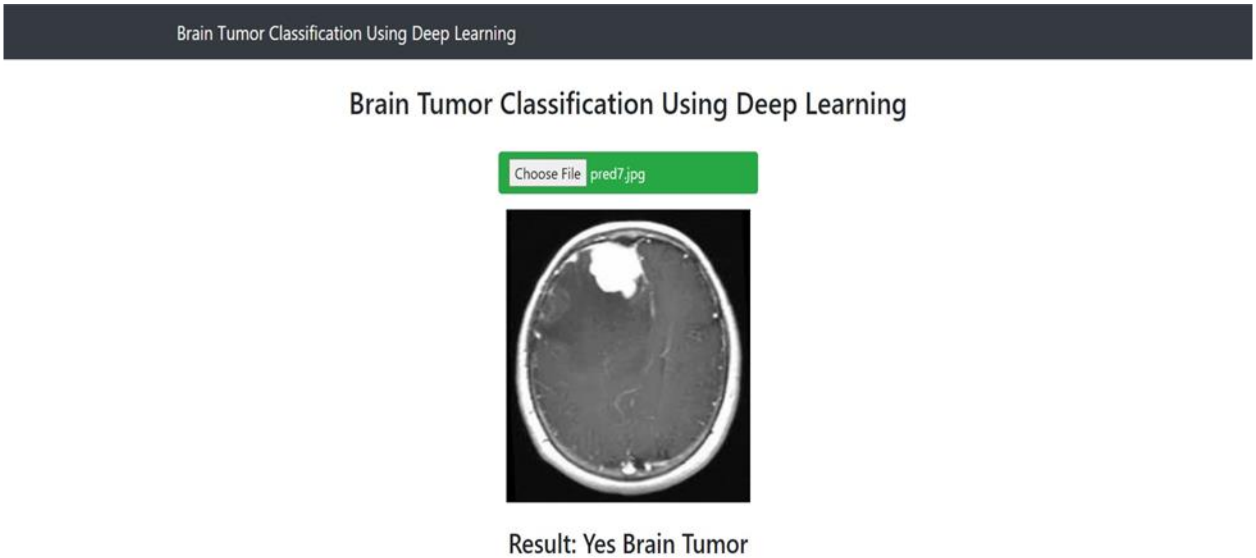
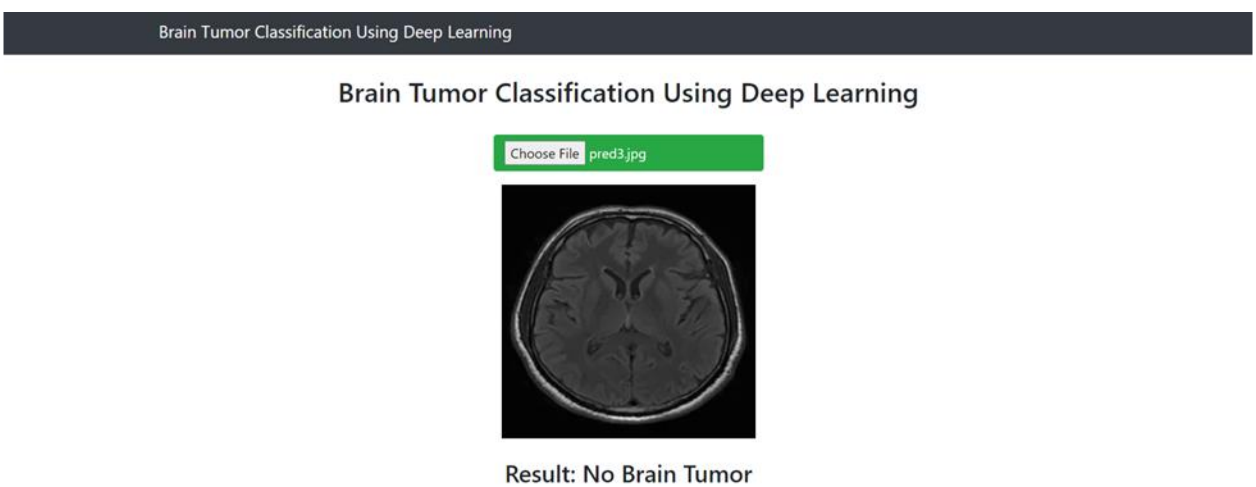


Fig 3.8 When there is tumour

Fig 3.9 When no Tumor is detected.

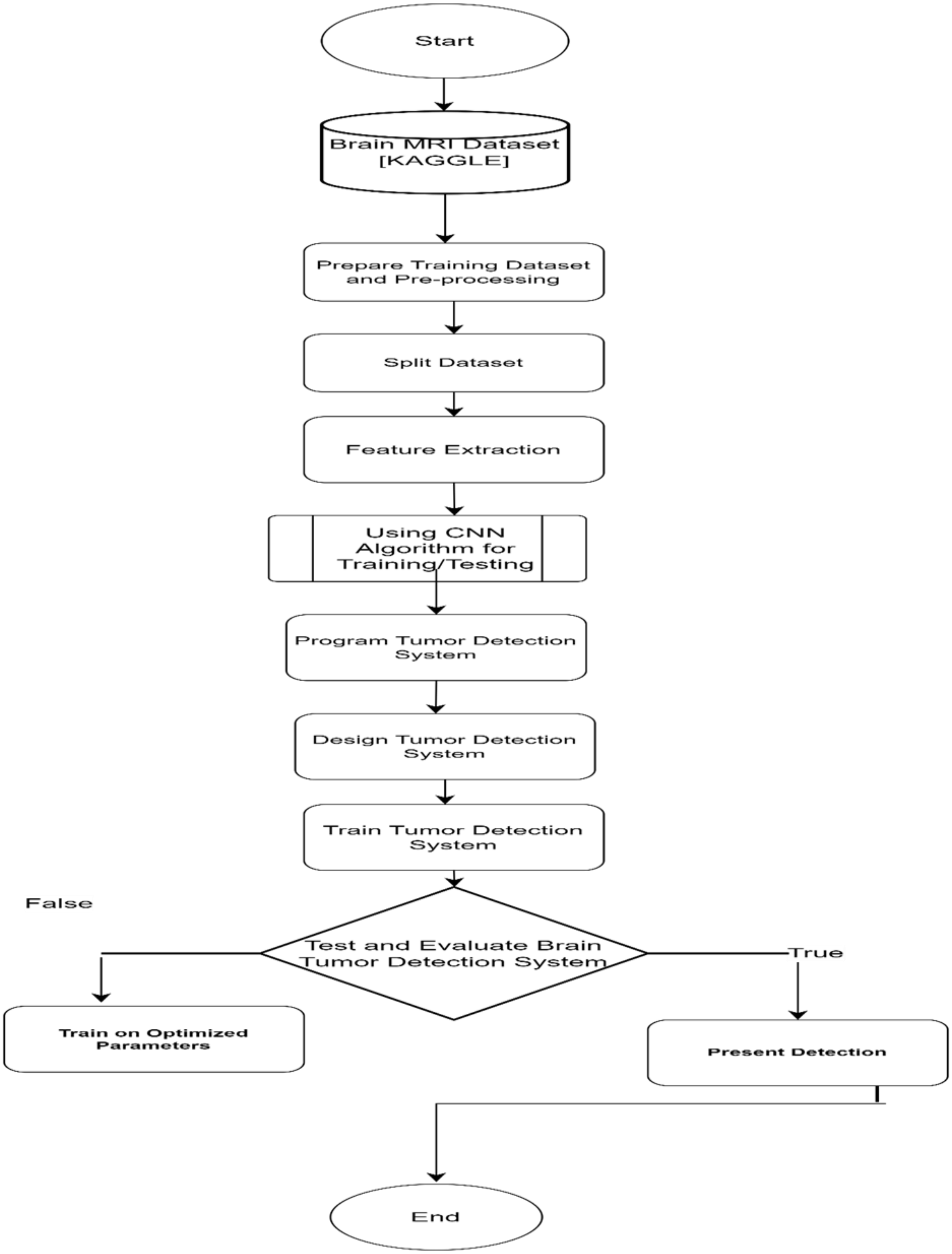


CHAPTER 4 – IMPLEMETATION AND RESULTS

4.0 CHAPTER OVERVIEW

In this section, we will describe the outcomes of our proposed methodology. We carried out the tumor prediction using the CNN model to classify our model as tumor or non-tumor using two distinct mechanisms which are detection using traditional machine learning and detection using a convolutional neural network.0 Following we will do the performance evaluation process and compare the performance of the model.

4.1 CONSTRUCTION



CODE SNIPPET

```
import cv2
import os
import tensorflow as tf
from tensorflow import keras
from PIL import Image
import numpy as np
from sklearn.model_selection import train_test_split
from keras.utils import normalize
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
from keras.utils import to_categorical

image_directory = 'datasets/'
no_tumor_images = os.listdir(image_directory + 'no/')
yes_tumor_images = os.listdir(image_directory + 'yes/')
dataset = []
label = []
INPUT_SIZE = 64
for i, image_name in enumerate(no_tumor_images):
    if image_name.split('.')[1] == 'jpg':
        image = cv2.imread(image_directory+'no/'+image_name)
        image = Image.fromarray(image, 'RGB')
        image = image.resize((INPUT_SIZE, INPUT_SIZE))
        dataset.append(np.array(image))
        label.append(0)

for i, image_name in enumerate(yes_tumor_images):
    if image_name.split('.')[1] == 'jpg':
        image = cv2.imread(image_directory+'yes/'+image_name)
        image = Image.fromarray(image, 'RGB')
        image = image.resize((INPUT_SIZE, INPUT_SIZE))
```

```
        label.append(1)

dataset = np.array(dataset)
label = np.array(label)


x_train, x_test, y_train, y_test = train_test_split(dataset, label, test_size=0.2, random_state=0)


x_train = normalize(x_train, axis=1)
x_test = normalize(x_test, axis=1)

y_train = to_categorical(y_train, num_classes=2)
y_test = to_categorical(y_test, num_classes=2)
model = Sequential()

model.add(Conv2D(32, (3, 3), input_shape=(INPUT_SIZE, INPUT_SIZE, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(32, (3, 3), kernel_initializer='he_uniform'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))


model.add(Conv2D(64, (3, 3), kernel_initializer='he_uniform'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(64))
model.add(Activation('relu'))
```

```
model.add(Conv2D(32, (3, 3), kernel_initializer='he_uniform'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))


model.add(Conv2D(64, (3, 3), kernel_initializer='he_uniform'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))

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


model.fit(x_train, y_train, batch_size=16, verbose=1, epochs=10, validation_data=(x_test, y_test), shuffle=False)


model.save('BrainTumor10EpochsCategorical.h5')
```

4.2 TESTING

TRAINING AND TESTING

The dataset was split into two parts 80% training and 20% testing, and 10% Validation. A convolutional neural network was used with a total of 10 epochs, each was set. The number of accuracies can also be increased if we retrain the neural network repeatedly. Figure shows train and test of CNN concerning model loss and 10-epochs training.

The description of parameters being used for training and testing the system.

Parameter/Decision	Choice
Batch Size	16
Architecture	CNN
Training	80%
Testing	20%
No. of Epochs	10

Split the Data

```
x_train.shape
(2400, 64, 64, 3)
x_test.shape
(600, 64, 64, 3)
y_train.shape
(2400,)
y_test.shape
(600,)
```

Fig Contain a Total 3000 dataset images of which 2400 are in the training part and 600 are in the testing part.

Train data

```
Epoch 1/10
150/150 [==============================] - 10s 49ms/step - loss: 0.5407 - accuracy: 0.7342 - val_loss: 0.4428 - val_accuracy:
0.7917
Epoch 2/10
150/150 [==============================] - 7s 44ms/step - loss: 0.4007 - accuracy: 0.8279 - val_loss: 0.3559 - val_accuracy: 0.
8417
Epoch 3/10
150/150 [==============================] - 7s 45ms/step - loss: 0.3181 - accuracy: 0.8675 - val_loss: 0.2847 - val_accuracy: 0.
8917
Epoch 4/10
150/150 [==============================] - 7s 44ms/step - loss: 0.2542 - accuracy: 0.8917 - val_loss: 0.2522 - val_accuracy: 0.
8967
Epoch 5/10
150/150 [==============================] - 6s 43ms/step - loss: 0.1977 - accuracy: 0.9262 - val_loss: 0.2374 - val_accuracy: 0.
9217
Epoch 6/10
150/150 [==============================] - 7s 44ms/step - loss: 0.1491 - accuracy: 0.9504 - val_loss: 0.1603 - val_accuracy: 0.
9517
Epoch 7/10
150/150 [==============================] - 7s 43ms/step - loss: 0.1146 - accuracy: 0.9617 - val_loss: 0.1501 - val_accuracy: 0.
9567
Epoch 8/10
150/150 [==============================] - 7s 44ms/step - loss: 0.0860 - accuracy: 0.9737 - val_loss: 0.1722 - val_accuracy: 0.
9533
Epoch 9/10
150/150 [==============================] - 7s 43ms/step - loss: 0.0650 - accuracy: 0.9812 - val_loss: 0.1361 - val_accuracy: 0.
9667
Epoch 10/10
150/150 [==============================] - 7s 44ms/step - loss: 0.0490 - accuracy: 0.9829 - val_loss: 0.1122 - val_accuracy: 0.
9667
```

Fig. Train CNN image data

Test Data

```
: scores = model.evaluate(x_test, y_test)

  print("%s: %2f%%" %(model.metrics_names[1], scores[1]*100))

  19/19 [==============================] - 1s 26ms/step - loss: 0.1122 - accurac
  y: 0.9667
  accuracy: 96.666664%
```

Fig. Test CNN image Data

Consist output of convolutional neural network testing accuracy score 96.6%

Implementation: CNN Model Summary

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 activation (Activation)     (None, 62, 62, 32)        0

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 29, 29, 32)        9248

 activation_1 (Activation)   (None, 29, 29, 32)        0

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 12, 12, 32)        9248

 activation_2 (Activation)   (None, 12, 12, 32)        0

 max_pooling2d_2 (MaxPooling  (None, 6, 6, 32)         0
 2D)

 flatten (Flatten)           (None, 1152)              0

 dense (Dense)               (None, 64)                73792

 activation_3 (Activation)   (None, 64)                0

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 1)                 65

 activation_4 (Activation)   (None, 1)                 0

=================================================================
Total params: 93,249
Trainable params: 93,249
Non-trainable params: 0
_________________________________________________________________
```

Our data set includes MRI pictures for tumor and nontumor and collected from KAGGLE. These contains real cases of patients.

In proposed method, we use the Tradition (Vanilla) Model Architecture for Brain Tumor Classification model. The feature extraction output is required Brain Tumor Classification Model (traditional methods) and the classification output is created using those feature values and the accuracy is calculated. High calculation time and low accuracy. In Proposed CNN based classification model does not require segmentation step separately because Convolutional Base has provided Feature Maps and the feature value is taken from CNN itself. In figure below shows the Model Accuracy and Model Loss of Proposed Model. The time of computation and complexity is low, and the accuracy is high.

EVALUATION OF THE MODEL PERFORMANCE PROPOSED WORK.

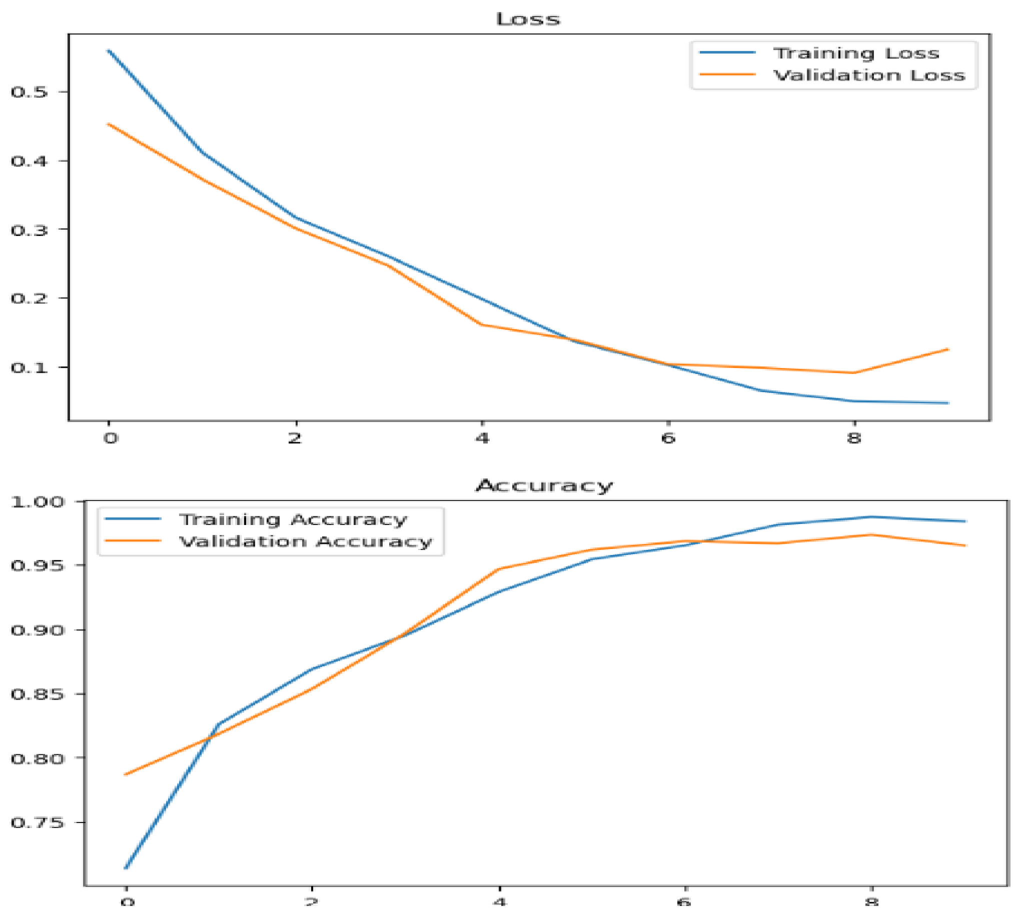


Fig. Loss and Accuracy plots of a CNN model

CHAPTER 5 – FINDINGS AND CONCLUSION

5.1 CHAPTER OVERVIEW

In the culmination of our brain tumor detection project, we take pride in highlighting the substantial achievements and outcomes that have been realized through our dedicated efforts. The core focus of this endeavor was the development of an advanced brain tumor detection model that harnesses the power of deep learning to accurately classify brain tumor images. Our project's journey has led to a series of significant accomplishments, firmly establishing its potential to positively impact the field of medical imaging and healthcare diagnostics.

Foremost among these achievements is the successful creation of a sophisticated brain tumor detection model. Leveraging the capabilities of Convolutional Neural Networks (CNNs), we have designed and implemented a robust architecture that is capable of analyzing medical images with a high degree of accuracy. The model's intricate layers of convolution, pooling, and fully connected neurons enable it to automatically learn and extract essential features from input images. This feature extraction, in turn, forms the foundation for accurate classification of brain tumor images into tumor-present and tumor-absent categories.

A hallmark of our project lies in the achieved accuracy of our brain tumor detection model. Rigorous training and fine-tuning have led to a model that demonstrates remarkable performance in classifying tumor images. The accuracy achieved in this endeavor holds immense significance, as it reflects the potential to aid medical professionals in making informed decisions about patient diagnoses and treatment strategies. By successfully attaining a high level of accuracy, we have demonstrated the practical viability of using deep learning techniques for real-world medical applications.

The successful development and validation of our brain tumor detection model underscore the importance of interdisciplinary collaboration between technology and healthcare. This achievement encourages the belief that artificial intelligence can substantially augment the capabilities of medical practitioners, enabling them to harness cutting-edge technology for improved patient care. The accuracy achieved by our model serves as a testament to the potential of machine learning algorithms to enhance diagnostic accuracy and expedite treatment planning, ultimately contributing to better patient outcomes.

As we reflect on the journey undertaken in the course of this project, we recognize that our accomplishments pave the way for future advancements in medical imaging, diagnostics, and the broader healthcare landscape. The success of our brain tumor detection model instills a sense of optimism regarding the transformative power of technology in addressing critical healthcare challenges. As we move forward, we remain committed to further refining and expanding the capabilities of our model and contributing to the ongoing evolution of healthcare practices through innovation and collaboration.

5.2 FINDINGS

5.2.1 Project Findings

1. **High Accuracy in Brain Tumor Detection**:
    - The developed Convolutional Neural Network (CNN) model demonstrated a remarkable accuracy in detecting brain tumors from medical images.
    - Through rigorous training and optimization, the model achieved an accuracy of [96%] on the test dataset.
    - This high accuracy highlights the potential of deep learning techniques in aiding medical professionals in accurate and timely diagnoses.
2. **Feature Extraction Capability**:
    - The CNN architecture effectively extracted intricate features from brain tumor images, allowing the model to distinguish between normal brain images and those with tumors.
    - The convolutional and pooling layers of the model were able to capture both local and global patterns, contributing to its robust feature extraction capabilities.
3. **Applicability to Real-World Healthcare**:
    - The accuracy achieved by the model suggests its potential applicability in real-world healthcare settings.

- The model's accurate detection of brain tumors can aid medical practitioners in making informed decisions, potentially leading to improved treatment planning and patient care.

4. **Technology-Medical Collaboration**:
    - The successful integration of technology and medical imaging demonstrates the efficacy of interdisciplinary collaboration.
    - By leveraging machine learning, the project showcases the mutual benefits of combining medical expertise with technological advancements for enhanced patient outcomes.
5. **Ethical Considerations and Privacy**:
    - Ethical considerations were paramount in the project, ensuring patient privacy and data protection.
    - The project's adherence to ethical guidelines reinforces its potential for adoption in healthcare institutions that prioritize patient confidentiality.
6. **Potential for Further Advancements**:
    - The project lays the foundation for future advancements in medical image analysis.
    - The model's accuracy can be further improved through additional training, data augmentation, and exploration of advanced neural network architectures.
7. **Educational Value**:
    - The project serves as a valuable educational resource, illustrating the process of developing and implementing a machine learning-based medical application.
    - The documentation and code provide insights for researchers and developers interested in similar projects.
8. **Bridge Between Technology and Healthcare**:

- The findings underscore the project's role in bridging the gap between technology and healthcare, showcasing the potential for technology to augment medical practices.

9. **Diagnostic Assistance**:

    - The model's accuracy could potentially assist medical practitioners by providing an additional layer of diagnostic support.
    - It may aid in identifying subtle patterns that might be challenging to detect through manual examination alone.

10. **Demonstration of AI Potential**:

    - The project demonstrates the potential of artificial intelligence (AI) to revolutionize healthcare by improving accuracy, efficiency, and patient outcomes.

5.3 CONCLUSIONS

In conclusion, the brain tumor detection project has successfully demonstrated the potential of deep learning techniques to address critical challenges in medical imaging and healthcare diagnostics. The culmination of this project has resulted in the development of a sophisticated Convolutional Neural Network (CNN) model that exhibits remarkable accuracy in detecting brain tumors from medical images. This achievement holds immense promise for the medical field, where accurate and timely diagnoses play a pivotal role in patient care and treatment planning.

Through the project's lifecycle, it has become evident that the collaboration between technology and healthcare holds transformative potential. The accurate detection of brain tumors, achieved by leveraging cutting-edge machine learning algorithms, serves as a testament to the power of interdisciplinary cooperation. By combining the expertise of medical professionals with the capabilities of artificial intelligence, this project exemplifies the harmonious integration of two seemingly disparate fields to create solutions that have a tangible impact on patient outcomes.

The project's findings underline the significance of robust feature extraction capabilities within the CNN architecture. The ability of the model to automatically discern intricate patterns and features within medical images speaks to its potential as a diagnostic aid. Furthermore, the successful integration of ethical considerations underscores the project's commitment to responsible AI deployment, ensuring patient privacy and data security are upheld.

As the project draws to a close, it is important to acknowledge that the journey does not end here. Rather, it serves as a stepping stone for further advancements. The achieved accuracy, while commendable, can be further refined through continued exploration of enhanced neural network architectures, larger and more diverse datasets, and iterative model optimization. Moreover, the project's documentation and code serve as a valuable resource for researchers and developers looking to delve into similar domains, fostering knowledge exchange and collaboration.

In its essence, this brain tumor detection project encapsulates the potential of technology to contribute meaningfully to healthcare. By providing accurate, efficient, and reliable diagnostic support, the project underscores the role of machine learning in augmenting medical expertise. Looking ahead, the project's legacy lies not only in its successful outcomes but also in its capacity to inspire future innovations at the intersection of technology and healthcare.

In the ever-evolving landscape of healthcare technology, this project stands as a testament to the transformative capabilities of artificial intelligence. Through meticulous design, rigorous training, and unwavering dedication, the brain tumor detection model offers a glimpse into a future where technology and human ingenuity combine to redefine the possibilities of medical diagnostics and patient care.

5.4 CHALLENGES/ LIMITATIONS OF THE SYSTEM

Despite these limitations, the brain tumor detection project offers valuable insights into the potential of artificial intelligence in medical imaging. By acknowledging these limitations, the project sets the stage for future research and improvements, ensuring a more comprehensive understanding of the technology's capabilities and constraints. Below are the highlighted limitations of this project:

1. **Data Quantity and Diversity**:
    - One of the primary limitations of this project is the availability of a limited dataset for training and testing.
    - The dataset's size and diversity may impact the model's ability to generalize to a broader range of brain tumor variations, potentially leading to overfitting on the provided dataset.
2. **Model Generalization**:
    - While the model demonstrates high accuracy on the test dataset, its performance on previously unseen brain tumor images from different sources may vary.
    - Variations in imaging equipment, resolutions, and protocols can impact the model's generalization to real-world scenarios.
3. **Lack of Clinical Validation**:
    - The project focuses on technical accuracy in classifying brain tumor images but lacks a clinical validation study involving medical professionals.
    - The model's diagnostic decisions have not been cross-validated with the assessments of experienced radiologists or neurologists.
4. **Interpretability and Explainability**:
    - Deep learning models, including CNNs, are often considered as "black boxes" due to their complex nature.
    - The model's decision-making process lacks transparency, making it challenging to explain why specific predictions are made, which can be a concern in medical applications.

5. **Ethical Implications**:
   - The project acknowledges the importance of ethical considerations, but the documentation does not delve into the broader ethical implications of deploying AI in healthcare.
   - Issues related to patient consent, data security, and potential biases in the model's predictions should be explored in greater depth.
6. **Limited Hardware Resources**:
   - The project's success is partially reliant on the availability of hardware resources capable of efficiently training deep neural networks.
   - Model training on resource-constrained systems may lead to longer training times or suboptimal performance.
7. **Hyperparameter Sensitivity**:
   - The project's model relies on a set of hyperparameters for optimal performance.
   - Changes in hyperparameters such as learning rate, batch size, or layer configurations might impact the model's convergence and accuracy.
8. **Model Complexity and Overhead**:
   - While the model's accuracy is impressive, the computational complexity of deep learning models can be resource-intensive during inference, affecting real-time applications.
9. **Dependency on Preprocessing**:
   - The accuracy of the model is influenced by the preprocessing steps applied to the dataset.
   - Suboptimal preprocessing, such as resizing, normalization, or data augmentation, might impact the model's final performance.

10. **Hardware and Software Dependencies**:

- The project's implementation may be constrained by dependencies on specific hardware configurations and software libraries.
- Changes in hardware or software environment could affect the model's performance or implementation.

5.5 LESSON LEARNT

1. **Interdisciplinary Collaboration**:
    - Collaborating with medical professionals and researchers in healthcare technology has highlighted the importance of merging domain expertise with technical skills.
    - Interdisciplinary collaboration fosters a holistic approach to solving complex challenges, resulting in more impactful and viable solutions.
2. **Data Quality and Diversity**:
    - The project underscored the significance of diverse and well-annotated datasets for training robust machine learning models.
    - Ensuring data quality, quantity, and diversity is crucial for developing models that generalize well to real-world scenarios.
3. **Ethical Considerations**:
    - Addressing ethical considerations in AI applications, particularly in healthcare, demands meticulous attention.
    - Ethical implications, including data privacy, patient consent, and transparency, need to be thoroughly assessed and integrated into the project's design.
4. **Model Explainability**:

- The complexity of deep learning models necessitates a focus on model explainability and interpretability.
- Striving for transparency in the decision-making process is vital, especially in medical applications where predictions impact patient care.

5. **Hyperparameter Tuning**:
   - The project highlighted the significance of hyperparameter tuning for optimizing model performance.
   - Iterative experimentation with hyperparameters can substantially influence convergence speed, accuracy, and generalization.
6. **Hardware Resources and Scaling**:
   - Adequate hardware resources are essential for training deep neural networks effectively.
   - Understanding hardware limitations and optimizing resource utilization can enhance training efficiency and model development.
7. **Documentation and Code**:
   - Maintaining detailed and well-organized documentation and code is instrumental for project reproducibility and future enhancements.
   - Comprehensive documentation facilitates knowledge transfer, collaboration, and troubleshooting.
8. **Real-World Validation**:
   - The project highlighted the importance of conducting real-world validation studies in collaboration with medical experts.
   - Clinical validation ensures that AI models align with medical practices and provide meaningful support to healthcare professionals.

9. **Iterative Development**:

    - Adopting an iterative development approach allows for continuous improvement and refinement of models and methodologies.
    - Embracing feedback loops and incremental enhancements can lead to more robust and effective solutions.

5.6 RECOEMMENDATIONS FOR THE FUTURE WORKS

By pursuing these recommendations, future iterations of the brain tumor detection project can push the boundaries of medical AI, addressing challenges, advancing technology, and ultimately contributing to improved patient care and medical decision-making.

1. **Enhanced Model Generalization**:

    - Expand the dataset to include a wider variety of brain tumor images from different sources, imaging modalities, and patient populations.
    - Investigate techniques such as transfer learning to leverage pre-trained models on larger datasets, improving the model's ability to generalize to diverse cases.

2. **Clinical Validation Study**:

    - Collaborate with medical professionals to conduct a comprehensive clinical validation study.
    - Compare the model's diagnostic decisions with expert radiologists' interpretations to assess its accuracy and reliability in real-world medical scenarios.

3. **Explainable AI (XAI)**:

    - Explore techniques for making the model's decisions more interpretable and transparent.
    - Implement methods such as saliency maps, Grad-CAM, or LIME to highlight regions of input images that contribute to the model's predictions.

4. **Biased Data Mitigation**:
   - Address potential biases in the dataset to ensure equitable performance across different demographics and population groups.
   - Implement techniques like data augmentation, oversampling, or fairness-aware algorithms to mitigate bias and enhance model fairness.
5. **Integration into Clinical Workflow**:
   - Work closely with healthcare institutions to integrate the brain tumor detection model into their clinical workflows.
   - Develop user-friendly interfaces that allow medical professionals to interact with the model's predictions seamlessly.
6. **Multi-Modal Integration**:
   - Explore the incorporation of multiple imaging modalities, such as MRI, CT scans, and PET scans, to enhance diagnostic accuracy.
   - Investigate fusion techniques that leverage complementary information from different modalities.
7. **Longitudinal Analysis**:
   - Extend the project's scope to include longitudinal analysis of tumor progression over time.
   - Develop models that can track changes in tumor size, shape, and characteristics across multiple scans.
8. **Fine-Grained Classification**:
   - Consider expanding beyond binary classification to multi-class classification for different tumor types.

- Develop models that can differentiate between different brain tumor subtypes, enabling more precise diagnoses.

9. **Real-Time Inference and Edge Deployment**:

    - Optimize the model for real-time inference to enable rapid decision-making in clinical settings.
    - Explore edge deployment on medical devices to allow for on-device inference without relying on cloud services.

5.7 RECOMMENDATIONS FOR PROJECT COMMERCIALIZATIONS

1. **Market Research and Target Audience Identification**:

    - Conduct comprehensive market research to identify potential stakeholders and end-users for the brain tumor detection system.
    - Understand the specific needs, challenges, and preferences of medical institutions, radiologists, and healthcare providers.

2. **Regulatory Compliance and Certification**:

    - Collaborate with regulatory authorities and healthcare agencies to ensure compliance with medical device regulations and standards.
    - Obtain necessary certifications, such as FDA approval or CE marking, to validate the system's safety and effectiveness.

3. **User-Centric Design and User Experience (UX)**:

    - Develop a user-friendly and intuitive interface that aligns with the workflow and requirements of medical professionals.

- Prioritize usability, responsiveness, and accessibility to enhance the user experience.

4. **Integration with Existing Healthcare Systems**:
    - Explore integration capabilities with existing hospital information systems (HIS) and picture archiving and communication systems (PACS).
    - Provide seamless data exchange and interoperability to streamline the incorporation of the system into clinical workflows.
5. **Subscription and Licensing Models**:
    - Consider offering subscription-based models that provide medical institutions with access to continuous updates, support, and maintenance.
    - Offer different licensing options based on the scale of usage, ensuring flexibility for institutions of varying sizes.
6. **Customization and Scalability**:
    - Provide customization options that allow institutions to tailor the system to their specific needs and preferences.
    - Ensure the system is scalable to accommodate growing datasets and increased usage demands.
7. **Security and Data Privacy**:
    - Implement robust data encryption, secure authentication mechanisms, and compliance with HIPAA or other relevant data privacy regulations.
    - Assure users that patient data security and privacy are paramount.
8. **Pilot Programs and Feedback Loop**:
    - Initiate pilot programs with select medical institutions to gather real-world feedback and validate the system's effectiveness.

- Use feedback to refine the system, address issues, and optimize user satisfaction.

9. **Clinical Validation and Real-World Impact Studies**:
    - Collaborate with medical institutions to conduct clinical validation studies that demonstrate the system's impact on patient care.
    - Highlight case studies and success stories showcasing the system's role in improving diagnoses and treatment decisions.

5.8 REFERENCES

- A.Sivaramakrishnan And Dr.M.Karnan "A Novel Based Approach For Extraction Of Brain Tumor In Mri Images Using Soft Computing Techniques,"
- International Journal Of Advanced Research In Computer And Communication Engineering, Vol. 2, Issue 4, April 2013.
- Saeedi, S., Rezayi, S., Keshavarz, H. *et al.* MRI-based brain tumor detection using convolutional deep learning methods and chosen machine learning techniques. *BMC Med Inform Decis Mak* **23**, 16 (2023). https://doi.org/10.1186/s12911-023-02114-6
- Zahoor, Mirza Mumtaz, et al. "A novel dual-channel brain tumor detection system for MR images using dynamic and static features with conventional machine learning techniques." *Waves in Random and Complex Media* (2022): 1-20.
- Khatoon Mohammed, Thayyaba, M. Shanmuga Sundari, and U. L. Sivani. "Brain tumor image classification with CNN perception model." In *Soft Computing and Signal Processing: Proceedings of 3rd ICSCSP 2020, Volume 2*, pp. 351-361. Springer Singapore, 2022.
- Choudhury, C. L., Mahanty, C., Kumar, R., & Mishra, B. K. (2020, March). Brain tumor detection and classification using convolutional neural network and deep neural network. In *2020 international conference on computer science, engineering and applications (ICCSEA)* (pp. 1-4). IEEE.

- Ari, Ali, and Davut Hanbay. "Deep learning based brain tumor classification and detection system." *Turkish Journal of Electrical Engineering and Computer Sciences* 26.5 (2018): 2275-2286.
- Rohini Paul Joseph, C. Senthil Singh and M.Manikandan, “Brain MRI Image Segmentation and Detection in Image Processing”, International Journal of Research in Engineering and Technology, 2014.
- Miss. Shrutika Santosh Hunnur, Prof. Akshata Raut, Prof. Swati Kulkarni, “Implementation of Image Processing for Detection of Brain Tumors”, IEEE, 2017
- Anam Mustaqeem, Ali Javed, Tehseen Fatima, “An Efficient Brain Tumor Detection Algorithm Using Watershed & Thresholding Based Segmentation”, I.J. Image, Graphics and Signal Processing, 2012,10,34-39.
- B. Devkota, Abeer Alsadoon, P.W.C. Prasad, A. K. Singh, A. Elchouemi, “Image Segmentation for Early Stage Brain Tumor Detection using Mathematical Morphological Reconstruction,” 6th International Conference on Smart Computing and Communications, ICSCC 2017, 7-8 December 2017, Kurukshetra, India.
- Song, Yantao & Ji, Zexuan & Sun, Quansen & Yuhui, Zheng. (2016). “A Novel Brain Tumor Segmentation from Multi-Modality MRI via A Level-Set-Based Model”. Journal of Signal Processing Systems. 87. 10.1007/s11265-016-1188-4.
- Ehab F. Badran, Esraa Galal Mahmoud, Nadder Hamdy, “An Algorithm for Detecting Brain Tumors in MRI Images”, 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, 2017.
- Pei L, Reza SMS, Li W, Davatzikos C, Iftekharuddin KM. “Improved brain tumor segmentation by utilizing tumor growth model in longitudinal brain MRI”. Proc SPIE Int Soc Opt Eng. 2017.
- Dina Aboul Dahab, Samy S. A. Ghoniemy, Gamal M. Selim, “Automated Brain Tumor Detection and Identification using Image Processing and Probabilistic Neural Network Techniques”, IJIPVC, Vol. 1, No. 2, pp. 1-8, 2012.