Contents

List of Figures

List of Acronyms

Acronym	Full Form
GUI	Graphical User Interface
AI	Artificial Intelligence
RSA	Rivest-Shamir-Adleman (Encryption Algorithm)
VGG-19	Visual Geometry Group Network - 19 Layers
GLCM	Gray-Level Co-occurrence Matrix
CNN	Convolutional Neural Network
IoT	Internet of Things
PoW	Proof of Work
ML	Machine Learning
CSS	Cascading Style Sheets
HTML	Hyper Text Markup Language
IDE	Integrated Development Environment
AIoT	Artificial Intelligence of Things
API	Application Programming Interface
ROI	Region of Interest
DB	Database
PNG	Portable Network Graphics
JPG	Joint Photographic Experts Group
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative

1. Introduction

1.1 Introduction

In the dynamic healthcare sector, the incorporation of technology has dramatically changed medical practice mainly in diagnostics and patient care. Because of dependence on the digital system, medical imaging has become a vital tool in the diagnosis and treatment of numerous health problems. On the other hand, while incorporating such advanced technology systems, there has been enormous difficulty with respect to the safekeeping of sensitive medical information and privacy. This report analyses the development and implementation of a system combining advanced image processing techniques with blockchain technology to effectively address such issues, bringing about a secure, accurate, and efficient solution for medical diagnostics.

Modern health care increasingly relies on medical imaging for an accurate identification, analysis, and treatment of complex medical conditions. Among such imaging modalities, brain imaging is most critical, especially when diagnosed in the brain, tumor, strokes, and other neurological disorders.Accurate detection and classification of affected regions in brain tumor images can only be achieved by implementing advanced techniques in image analysis. Basic steps include image acquisition, pre-processing, feature extraction, segmentation, and classification. Each of these phases is of prime importance to ensure better accuracy and reliability in diagnosis. In this project, each of the steps is optimized with new algorithms and models in order to improve the effectiveness of the results..

Despite the advancements in image processing and diagnostics algorithms, the most relevant problem persists—the security of the data. Most images of the medical body carry sensitive data about the patient which ought to be kept under wraps as confidential. Conventional systems might work efficiently in the processing of images. However, a majority lack the necessary security measures to prevent unauthorized access or breaches. Such vulnerability not only violates patient privacy but also weakens the trust of patients in such digital healthcare systems. This significant concern has urged adding an extra layer of security through blockchain technology.

Blockchains represent the most decentralized and immutable technology to date. It has been seen that its core characteristics—transparency, traceability, and tamper resistance—are especially suited to protecting sensitive medical data. Here, blockchain will be used in encrypting and securing medical image data, while ensuring access only to users authorized to have it. Another mechanism to ensure data security uses asymmetric encryption techniques like RSA, which enforces access control at the decryption level and ensuring that the information in its decrypted form is accessible only to the desired recipients. In the application presented here, the combination of blockchain and encryption as much as possible decreases the risk factor and increases the secure environment of managing medical images without breaches of unauthorized access to the data.

The methodology presented for this project is strong and unique because it allows for the maximization of accuracy in diagnosis while also providing data security. The process will begin by procuring a dataset of brain tumor images, which will then undergo a set of pre-processing operations that enhance the quality of the images for further analysis..

These processing steps involve image resizing to a standard scale, transformation to grayscale for ease of processing, and bilaterally filtering to effectively remove noise while maintaining the critical features. On pre-processing the key feature extraction is performed by statistical and texture-based techniques, which include Mean Standard Deviation as well as the Gray-Level Co-occurrence Matrix (GLCM), providing an underlying basis for the following analytical operations and classification processes.

The segmentation phase is very important for isolating the areas of interest within the medical images. It has effectively been done by using thresholding techniques wherein the tumor areas have been separated for proper analysis and classification. Further, the segmented images are divided into both the training and testing datasets for developing as well as evaluating the classification models. Transfer learning algorithms, for example, VGG-19 and Inception, are implemented to classify images with the presence of the tumor. Such algorithms are highly rated based on high accuracy efficiency and utilization of pre-trained weights and architectures for predicting reliable results.

To assess the performance of the system, several metrics are analyzed, including accuracy, error rate, and confusion matrices. Validation graphs are generated to evaluate the system's consistency and reliability across various datasets. These assessments provide valuable insights into the effectiveness of the implemented methodologies and identify areas for potential improvement.

For judging the performance of the system, metrics like accuracy, error rate, and confusion matrices are analyzed. Some validation graphs are generated to figure out how consistent and reliable the system is to appropriate datasets. These analyses give a good idea of how well the methodologies implemented work and where improvements are possible.
One of the important components of the system is the blockchain part, which aims to further enhance the image processing workflow by making sure that data security is met across its lifecycle. Every transaction, be it storing, retrieving, or transmitting medical images, is embedded as a block into the blockchain. Each block contains metadata including timestamp, sender information, receiver information, and the usual hash key to ensure all data accuracy. Proof-of-work mechanism makes the blockchain extremely difficult for malevolent actors to alter the data or hack as it involves a significant strengthening of security.

As part of the features to make the web application user-friendly, the web application will consist of an intuitive user interface accessible to users. Since this web application will be developed in Python and Streamlit, users will be allowed to upload their medical images and start a prediction process to see the results. The interface also has user registration and login options to make access strictly available to a limited number of people. By enabling an amalgam of sophisticated functionalities along with an intuitive interface, it should help bridge technical overcomplexity with practical usability in an endeavor to improve accessibility for a wide and diverse audience.

Beyond its technical achievements, this project may make a significant impact in the health care area. Introducing artificial intelligence to blockchain technology gives the system an edge in two necessary needs: high reliability in diagnosis and the protection of patients' data. These twin goals guarantee that the system is both effective and ethical-a need dictated by the increasing trend for secure and reliable healthcare solutions. In addition, the project truly showcases medical imaging innovations and sets a precedent for future generations of data security. Future challenges can indeed be better tackled with interdisciplinary approaches.

1.2 Overview

In today's technology-driven healthcare landscape, advancements in medical imaging have significantly enhanced diagnostic accuracy. However, the growing reliance on digital solutions has introduced challenges related to security, privacy, and efficiency. This project addresses these concerns through an integrated approach combining advanced image processing techniques and blockchain technology to create a robust and secure system for analyzing medical reports, particularly brain tumor imaging.

The primary goals of the project are:
Accurate Medical Diagnosis: Develop an efficient image processing pipeline to classify medical images and identify affected regions with high precision.
Enhanced Security: Employ blockchain technology to ensure the integrity, privacy, and security of sensitive patient data.
Ease of Access: Provide a user-friendly interface for seamless interaction between users and the system.

Key Components:

1. Medical Image Processing

The system processes medical images (brain tumor datasets) using a comprehensive pipeline:

- Pre-Processing: Improves image quality by resizing, converting to grayscale, and applying bilateral filtering.
- Feature Extraction: Key characteristics are extracted using statistical and texture-based methods such as the Gray-Level Co-occurrence Matrix (GLCM).
- Segmentation: Affected regions are identified through thresholding techniques.
- Classification: Advanced deep learning algorithms like VGG-19 and Inception are employed to classify images as affected or unaffected.

2. Blockchain Integration

Blockchain technology ensures data security and privacy through:

- Decentralized storage of patient data to prevent unauthorized access.
- Asymmetric encryption (RSA) for secure transmission and storage of sensitive information.
- Immutable records of data transactions using hash keys and Proof-of-Work (PoW).

3. Performance Evaluation

The system’s effectiveness is measured using key metrics like accuracy, error rates, and confusion matrices. Validation graphs further illustrate the reliability and consistency of the predictions.

4. User-Friendly Web Application

A web-based interface, developed using Python and the Streamlit framework, allows users to:

- Upload medical images for analysis.
- View classification results and suggested remedies.
- Access features like secure user authentication and seamless navigation.

Technological Stack

- Programming Language: Python
- Frameworks and Tools: Streamlit, Anaconda Navigator (Spyder IDE)
- Frontend Technologies: HTML, CSS
- Security Mechanisms: Blockchain, RSA Encryption.

Technologies and tools used in this project form a core pillar of the achievement of the objectives of secure, efficient, and user-friendly medical image processing. With careful selection for each element of the technological stack, the requirements were targeted to be achieved by having optimal performance and scalability.

Among the used frameworks and tools, the list includes Streamlit and Anaconda Navigator-Spyder IDE. Streamlit would make the interface of the user interface very easy to develop a simple yet powerful framework to be able to come up with an interactive and intuitive web application; one can readily upload medical images, present diagnosis results, and access the recommended remedies. The dynamic visualization ability will be extremely helpful in plotting graphs, confusion matrices, among others, and performance metrics to enhance the user experience. Development Environment: Anaconda Navigator is an interface for the Spyder IDE where the Python code is created and executed. Overall, the environment is linear, and it caters to dependencies, libraries, and computational power, ideal for developing and debugging highly complex functionalities such as image processing and blockchain.

Frontend technologies depend on HTML and CSS. HTML is used for writing up web applications; it provides an underlying architecture that positions content in such a manner that all functionalities such as upload and display of results function. CSS adds aesthetic value and usability value to an application through responsive design as well as personalization through styling. Collectively, they add functional and aesthetic value to the web application, targeting medical professionals and researchers broadly.

The major components of the project are security mechanisms. RSA encryption has been used to maintain data privacy and integrity. It is proposed to use blockchain technology to ensure the secure storage and transfer of all sensitive medical data. Every interaction with the system, which may range from uploading a medical image to retrieving diagnostic results, is treated as a block on the blockchain. The blocks hold the metadata, which consists of the timestamps, sender information, receiver details, and hash keys that ensure data immutability and traceability. Being decentralized, blockchain makes it impossible to have a single point of failure; hence, they are highly resistant to cyberattacks.

RSA encryption is one of the asymmetric encryption techniques adopted broadly, and it goes well with blockchain to enhance security. Thus, only the intended person can decode the sensitive data thus keeping the patients' confidentiality intact. The channels for communication while uploading images or retrieving diagnostic results are secured so that during unauthorized access through RSA encryption data transmission cannot be done. The RSA encryption, combined with the blockade, shall form a solid security framework in tandem with stringent requirements for privacy in handling medical data.

All in all, such integration of the technologies and tools in a system will ensure technical competency, friendliness, security, and adaptability to real-world medical diagnostic needs. Proper realization of goals in terms of enhancing diagnostic accuracy and protection of sensitive data that this rich stack of technologies ensures is implemented without any hindrance or inefficiency.

1.3 Objectives, Methodology, and Scope

Objectives

1. Accurate Medical Image Diagnosis:
   - Develop an efficient system to analyze medical images, specifically brain tumor datasets, for accurate classification into affected or unaffected categories.
   - Identify and segment affected regions in the images for detailed analysis.
2. Enhanced Security and Privacy:
   - Employ blockchain technology to ensure secure, tamper-proof storage and transmission of sensitive medical data.
   - Use RSA encryption to protect data privacy during interactions between users and the system.
3. User-Friendly Accessibility:
   - Provide a web-based platform with an intuitive GUI to enable users to upload images, view results, and access remedy suggestions seamlessly.
4. Performance Evaluation:
   - Evaluate the system using metrics such as accuracy, error rate, and confusion matrices to ensure reliability and effectiveness.

Methodology

1. Data Collection and Pre-Processing:
   - Utilize a brain tumor dataset in .png or .jpg format as input.
   - Pre-process images by resizing, converting to grayscale, and applying bilateral filtering to enhance image quality.
2. Feature Extraction and Segmentation:
   - Extract critical features using statistical methods like mean standard deviation and Gray-Level Co-occurrence Matrix (GLCM).
   - Segment affected regions using thresholding techniques to isolate tumor regions for detailed analysis.
3. Classification:
   - Employ transfer learning models, such as VGG-19 and Inception, to classify images as affected or unaffected.
   - Train and test the models using split datasets for improved accuracy and generalization.
4. Blockchain Implementation:
   - Design and deploy a blockchain framework to secure data transactions.
   - Encrypt medical images and related data using RSA encryption and store the information in blocks with hash keys, timestamps, and metadata.
5. Web Application Development:
   - Develop a user-friendly interface using Python and the Streamlit framework, complemented by HTML and CSS for visual design.
   - Integrate features like file uploads, prediction displays, and secure user authentication.
6. Performance Analysis:
   - Measure the system's accuracy, error rate, and reliability through validation graphs, confusion matrices, and comparison charts.

Scope

1. Medical Diagnostics:
    - Revolutionizes the way medical images, particularly brain tumor scans, are analyzed and classified.
    - Offers actionable insights to medical professionals, aiding in timely and accurate diagnosis.
2. Data Security and Privacy:
    - Implements blockchain technology to set a new standard for safeguarding sensitive medical information.
    - Addresses privacy concerns in digital healthcare systems, building trust among patients and healthcare providers.
3. Technological Integration:
    - Combines AI-driven image processing with blockchain for a secure and efficient healthcare solution.
    - Demonstrates the potential of interdisciplinary approaches in solving real-world problems.
4. Future Applications:
    - Serves as a prototype for integrating blockchain and AI in other areas of healthcare, such as radiology, pathology, and patient record management.
    - Offers scalability for implementing similar systems in different medical imaging scenarios.

2. Methodology

3.1 Blockchain Integration and Encryption Methodology

The integration of blockchain in the project has enhanced security and privacy because it uses SHA (Secure Hash Algorithm) and RSA (Rivest-Shamir-Adleman) encryption together to ensure that medical data within the blockchain is robust in terms of integrity, confidentiality, and immutability.

SHA (Secure Hash Algorithm):

SHA produces a unique hash value for each block in the blockchain. A hash is an alphanumeric string of fixed length coming out from data in the block, like details of medical images, metadata including sender, receiver, timestamp, and the hash of the previous block. It's considered the digital fingerprint of a block and assigned with all the main properties of SHA.

- Deterministic: The same input always leads to the same output.
- Irreversible: It cannot be reversed to deduce the input from the output.
- Collision-resistant: Any two different inputs cannot produce the same hash.

Whenever medical images and their data are processed, a hash is created using SHA, which is represented as SHA-256. This hash ensures the following:

- Data Integrity: Any manipulation of data changes the hash. Manipulation is thus detected right away.
- Chaining Blocks: Each block points to the hash of the previous block so that it is a secure chain. This linkage does not permit retroactive changes without breaking the entire chain.
- For instance, if a block has encrypted medical image data, SHA hashes ensure that, upon being appended to the chain, the block content cannot be modified. Whenever the attacker tries to alter a block, the hash becomes mismatched and, thus, invalidates the chain.

RSA (Rivest-Shamir-Adleman):

RSA is an asymmetric encryption algorithm that applied to the blockchain in securing data transmission. It provides a pair of keys: a public key, which has to be made public, is used to encrypt data, and a private key that is kept secret and will be the one decrypting the data. In this application, RSA secures communication and data storage about medical concerns.

- Message Encryption: Upon uploading the medical image, the data of the same is encrypted with the public key of the recipient while adding it to the blockchain. Thus, the decryption of that information would be only possible when accessed through the respective private key by the intended recipient.
- Authentication: RSA also authenticates the sender by offering digital signatures. It will give a private key to the sender to sign the encrypted data, and then the recipient will use the sender's public key to verify the authenticity of the data.

RSA not only provides intrinsic security but may also be used as extra encryption of data in the blockchain. In case a hacker were to gain access to the blockchain, he/she would not be able to decrypt it without having the corresponding private key.

SHA and RSA provide an integrated framework for secure and complete safety measures: Data confidentiality RSA encrypts sensitive medical information, available only to authorized parties. Data Integrity SHA verifies any alteration through hash values. Immutable Storage: SHA hashes, along with the structure of blockchain, create an immutable ledger. RSA Ensures the data exchanged between nodes is tamper-proof. RSA has a critical role to play in ensuring the medical records were properly protected and kept confidential. It surely addresses critical issues regarding data security and privacy. SHA makes integrity checks on the data of the blockchain, while RSA does confidentiality checks. In other words, together, they ensure safe storage and sharing of medical data with robust infrastructure.

3.2 Image Processing Techniques

Image processing techniques that lie in this project form a robust pipeline of analysis on medical images that focuses specifically on brain tumor datasets. Such techniques, ranging from pre-processing to feature extraction, segmentation, and preparation for machine learning themselves improve the accuracy of classification and prediction tasks. These methods are discussed in detail as follows:

1. Pre-Processing Techniques

It involves improving the quality and consistency of raw input images. Medical images are often noisy, with inconsistent dimension and unnecessary complexity, making analysis cumbersome. It resolves this by cleaning the images through:

- Resizing: Medical images are resized to a standard dimension, ensuring uniformity across the dataset. This step is crucial as neural networks and machine learning models require inputs of fixed sizes. Resizing maintains the image's spatial integrity while making it compatible with the model architecture.
- Grayscale Conversion: the reduction of the complexity of images by removing color information, which is often irrelevant to the medical image, while preserving the intensity details for further processing.
- Bilateral Filtering: It smoothens images by removing noise while preserving the edges. Bilateral filtering does not blur the boundaries between regions like standard smoothing filters do because these boundaries represent the edge details in medical images, which can signify the boundary of anomalies such as tumors.

2. Feature Extraction

Feature extraction entails extracting significant features in the pre-processed input images, which are then fed to the classification algorithms. This project uses two main techniques for feature extraction:

- Mean Standard Deviation: This statistical measure gives the information regarding the intensity distribution of the image. It calculates the mean and variation in pixel intensities, thus capturing the overall texture and brightness levels of the image. The feature is particularly useful in indicating abnormal regions because tumors often follow specific patterns of intensity that are different from normal tissues.

- Gray-Level Co-occurrence Matrix (GLCM): GLCM is a widely used method for texture analysis.

The spatial relationship of pixel intensities in an image is assessed, which generates a matrix that depicts how often pairs of pixel intensities (gray levels) appear at distances and angles. Some features produced by GLCM include contrast, correlation, energy, and homogeneity, which are very useful in distinguishing normal regions from affected regions in medical images.

3. Segmentation

Segmentation is an important step in any method used in medical image analysis, and it is aimed at segmentation of a tumor or affected tissue. In this exercise, thresholding is utilized as one of the simple and most efficient segmentation techniques.

- Thresholding Method: The method splits the image into the foreground, that is, the region of interest, and background, based on a pixel intensity threshold. Qualifying pixels lie above this threshold; the remaining ones are assigned to the background. In detecting brain tumors, thresholding results in regions that are isolated by differences in the intensity value from normal brain tissue. Segmentation simplifies a complex image, highlighting the region of concern and discarding the rest.

4. Image Segmentation

The processed images are further divided into subsets for the development of machine learning:

- Training Images: These are the images used for training a classification model. They constitute most of the dataset and are used in the process of training so that the model can ascertain patterns and relations within the data.
- Testing Images: This subset tests the performance of a model to unseen data to ensure that it is generalizing beyond the training set.

Image splitting will ensure that the model is well trained and validated on appropriately different and non-overlapping data, to avoid overfitting.

This Image Processing pipeline directly works with blockchain and classification technologies. Extracted features of the image along with regions that are further segmented form the base for classification algorithms using transfer learning like VGG-19 and Inception. At the same time, it safely stores and shares information and data by processing images by using blockchain technology; it ensures intimacy and authenticity of medical information. It also ensures that higher accuracy and efficiency in medical image analysis are achieved while, at the same time, trying to reach the widest possible goal to support security and privacy when handling sensitive information. By integrating basic image processing techniques with advanced computational approaches, we provide improved medical imaging.

3.3 Stages of Input and Classifications

Input Stage

The methodology begins with the input phase, where a medical image dataset is the source input to the other following processes. These datasets are commonly sourced from reputable medical repositories and are highly contributed by real-world examples of affected versus unaffected brain regions. On this note, the input phase emphasizes obtaining raw image data for analysis. Steps begin with loading the dataset, ensuring compatibility, and checking its quality. This is a critical stage as all subsequent stages would be incorrect only if the input data were correct. In addition, metadata including patient identifiers and timestamps are captured so that every image can trace back to and securely link up with the source. With this standardized input and of very good quality, the system is then prepared for strong processing.

Classification

Classification then occurs following the pre-processing, image segmentation and separation into training and testing sets. Advanced machine learning techniques coupled with precision transfer learning are used in this project with models such as VGG-19 and Inception. Then, the pre-trained models on such large sets for images are further fine-tuned specifically to solve the task at hand: whether a medical image indicates an affected or unaffected region.

- Feature Use: The features extracted include texture metrics derived from GLCM except the intensity statistics and used as inputs to the models.
- Deep Learning Models: These models consist of deep convolutional layers equal to 19. VGG-19 maximizes function representation. Inception Models: Optimized Architecture for Accuracy in Multi-scale Convolutions.
- Training process: Models learn normal versus abnormal patterns and come up with experience in medical imaging complexities as part of the training process.
- Testing and Validation: These learned models are now tested over the unseen data. This will check whether they are generalising well or not. The testing phase does not just predict the affected areas but also provides the framework for robust assessment of the performance.

Prediction and Recommendation

After the classification, it gives predictions regarding the state of the images under analysis. If the input image is classified as "affected," it marks the area under impact and creates a visual mark to better understand it. This stage also comprises recommendations by suggesting action by result according to the output prediction. The model outputs a binary result—affected or unaffected. The affected status is further supplemented with confidence scores, showing how certain the model feels about its classification. It suggests possible next steps, which may be contacting a specialist, more diagnostic procedures, or reviewing specific parts of the brain. These recommendations are premade, based on actual best practices in medicine. Then the outcome is represented in an understandable fashion so there could be clear visuals including heat maps and textual explanations. That makes the system accessible to medical professionals and other interacting users.

Performance Estimation

The performance estimation stage evaluates the effectiveness of the model, checks its reliability for applications in healthcare, and computes accuracy, error rate, and confusion matrix to measure the ability of the system to predict:

- Accuracy Measurement: The ratio of correctly classified images to the total images provides an overarching view of the model's performance. High accuracy reflects the model's capability to discern anomalies accurately.
- Error Rate Analysis: This measure outlines cases of wrongly classifying outputs; it hence suggests there is room for improvement.
- Validation Graphs: These graphs show the training and validation accuracy and loss with the number of epochs of the model, such as overfitting or underfitting.
- Comparative Analysis: If more than one model is implemented, such as VGG-19 and Inception, their performances are compared to select the best-performing approach for deployment.

3.4 Web application development

It will include the user-friendly web interface, integrating all functionalities, those which the end-users such as the medical professional or researchers interact most with.

- Images upload and prediction: The mobile application would permit the patient to upload medical images directly within the application. Therefore, a back-end operation of initial pre-processing, feature extraction, and then classification based upon the uploaded images.
- Real-Time Feedback: Though the application does promise real-time predictions, details will include the type of region that is affected and break up the affected area into segments along with their confidence levels.
- Technological Framework: A web application developed with Streamlit framework, a powerful Python-based app for interactive data applications Front-end: Code for proper, accessible design using HTML and CSS.
- Performance Reporting: Users will find accessed performance metrics such as validation graphs and comparative analysis after analysis. It's very helpful for the researcher to evaluate this model for reliability.
- Integration with Blockchain: The web application will perfectly interact with the blockchain in such a way that each of the uploaded images and corresponding metadata-such as its user information, prediction results-will be securely recorded on the blockchain. Indeed, integration with blockchain will serve to enhance transparency and to tamper-proof something, making users comfortable.
- Development Environment: This is developed in Python and is compatible with many IDEs, such as Spyder, and offers the possibility of effective development through it.

3. Model Architecture

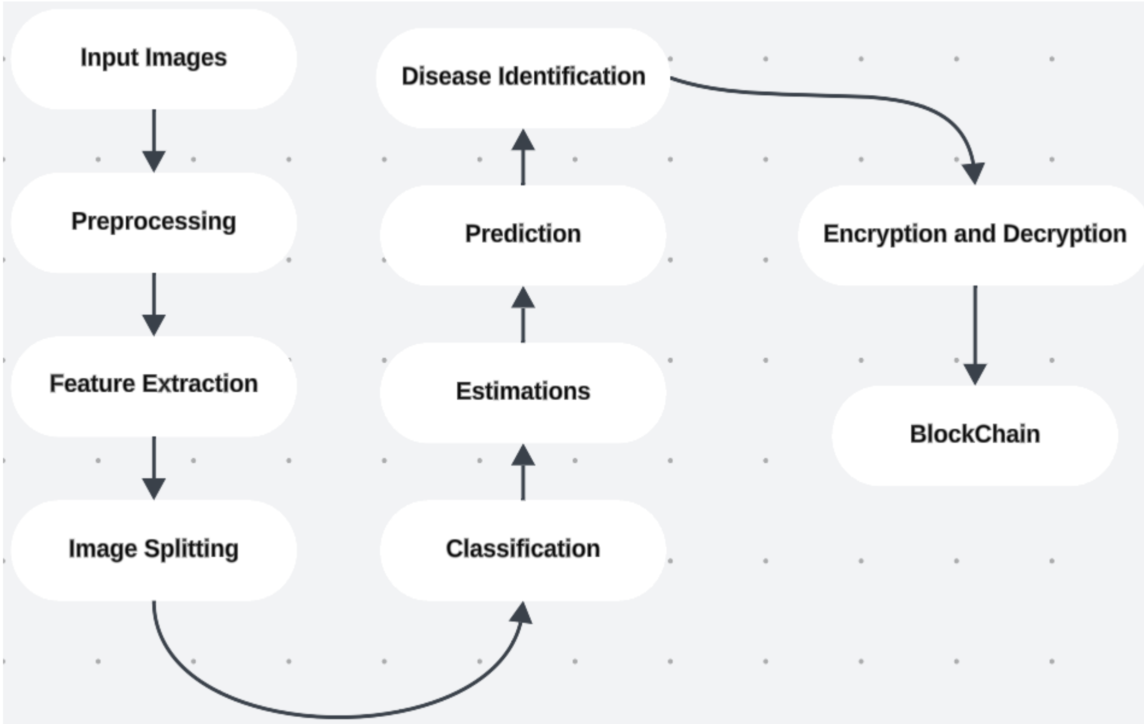


Figure 1Model Architecture

The workflow starts from data input and progresses through classification, prediction, performance evaluation, and a web-based user interface. It starts by ingesting various medical images, especially datasets of brain tumors in the formats .png or .jpg. Such images are then pre-processed; the goal is to standardize their dimensions and quality for compatibility with the following machine learning models. Some additional metadata such as time stamps and user identifiers is added to the data for its complete traceability and security throughout the process. This makes the data well-structured, and analysis and visualization can be done effectively for better medical decisions.

4.1 Pre-processing stage

The pre-processing stage of this project is important because it is used to prepare raw medical images for detailed analysis. Medical images seem to suffer from inconsistencies, such as resolution differences, noise, or other irrelevant details, which could feed into machine learning models inappropriately. To conquer these challenges, the project does employ necessary pre-processing techniques such as image resizing and grayscale conversion, therefore standardizing and simplifying the dataset being used.

The pre-processing stage thus converts raw medical images in the clean format from a dirty format in the precise manner, therefore providing good foundation ground for reliable feature extraction and classification. Image resizing and grayscale conversion are simple but very effective processes for data enhancement both in regard to quality and consistency and ensuring reliable subsequent stages.

1. Resizing Images

The first pre-processing step is to resize the images of the same dimension. Most medical image datasets face this problem of size variation; it is going to make a problem as such input dimensions are incoherent when fed towards the machine learning models. Uniformity can be ensured across the dataset so that the output looks good for neural networks, which are sensitive to the shapes of the inputs. This option will resize all images to a certain resolution, though it will not distort the aspect ratio. It would be 224x224 pixels, for example. Interpolation techniques, such as bilinear or bicubic interpolation, are used.

Purpose: Resizing retains spatial relationships and respects the integrity of important features such as an edge position inside tumor or anomaly by decreasing the complexity of computation. Standardization of dimension to resized images makes them perfectly aligned with the requirement of deep-learning models like VGG-19 and Inception, among others.

2. Gray Scale Conversion

Medical image data are mainly assumed to contain color information, most of which is irrelevant to the analysis. For example, the intensity-driven rather than the color-driven brain tumor images transform the data into a more simplified version to relieve redundancy. This transforms an image from three-channel RGB into a one-channel format in which pixel intensities in the shade vary from 0 for black to up to 255 for white. It saves a large amount of size in data and computational loads with not losing one single particle of information to the analysis.

These images emphasize texture, contrast, and intensity patterns, as these are prominent features to ascertain anomalous objects such as tumors. This simplification also reduces noises and less important details that make feature extraction easier.

Advantages of Pre-Processing:

- Standardization: Resizing ensures images are all standard sizes, which will now prove to be useful in processing them in a batch within the neural network.
- Simplification: This greyscale conversion method simplifies the input data without losing their important features that can be beneficial for handling it with greater accuracy.
- Two techniques in concert remove noisy features from the images; overall, this would increase their signal-to-noise ratio.
- Compatibility Improved: An input pre-processed image will be apt, enabling the advanced classification model used to perform optimally during the training and testing stages.

4.2 Feature Extraction

where critical information is extracted from medical images for further processing. The techniques applied are as follows:

1. Mean, Median, and Mode (Statistical Features)

These statistical measures summarize the distributions of pixel intensities in an image:

- Means: Mean value pixel intensity. This measures overall brightness or contrast for an image-this can help in distinguishing normal from abnormal regions.
- Median: It gives the middle pixel intensity and therefore it allows to resist much noise or erroneous outlier pixels.
- Mode: The maximum pixel intensity, used in case of strong features or patterns, for peak detection.

These measures provide the basis for identifying regions of interest, since most abnormalities will present as deviations in these features relative to normal tissue.

2. Gray-Level Co-Occurrence Matrix (GLCM) Features

An application of GLCM is the extraction of texture information from the image. It is describing how often pairs of pixels values-gray levels-occur in a specified spatial relationship. Frequently extracted features by using GLCM consist of

- Contrast: Measures the variation in the intensities of pixels, highlights edges or anomalies.
- Correlation: Represents the linear association between pixel pairs, indicative of uniformity or texture.
- Energy: Summarizes the textural uniformity, with higher values indicating smoother textures. Homogeneity: Assesses the similarity of pixel values, detecting consistent patterns in the image.

After pre-processing with resize, grayscale conversion, and bilateral filtering, they are extracted as features to represent the unique patterns from the medical images. Global features in statistics of the mean, median, and mode are accompanied by detailed textures and structural properties in GLCM features of the affected and unaffected regions. The features extracted above are then used in segmentation to isolate the affected regions and in classification algorithms, namely VGG-19 and Inception, to predict whether the input image is supposed to signify a medical condition.

4.3 Image Splitting

The feature-extracted medical images split into training and testing images. It is an important process in preparing the data for building proper decision-making pattern development and the effective classification model.

- Training Set: The training images prepare the classification model by training it, such as using transfer learning algorithms like VGG-19 or Inception. The model learns patterns, features, and relationships during this stage about the data that distinguish the affected regions from unaffected ones.

- Testing Set: The testing images are left for testing the performance of the model learned.

These images ensure the generalization of the predictions of the model to unseen data, evaluating metrics such as accuracy, error rate, and confusion matrix.

4.4 Classification

Architectures are essentially defined at the classification stage using deep learning models such as VGG-19 and Inception. The pre-trained big datasets, they are further fine-tuned to extract the difference between affected brain regions and healthy ones. The extracted features utilize intensity statistics and texture metrics from preprocessed images to ensure accurate predictions by the classification module.

- VGG-19: Since VGG-19 maintains very deep convolutional layers, it actually captures very fine-grained spatial information in images and is quite appropriate for medical imaging activities.
- Inception: It adapted multiscale operations on convolutions that achieved optimization in computing efficiency and accuracy, thus boasting robust performances on different medical image datasets.

4.5 Estimations and Prediction

Estimation (Accuracy and Error Rate)

Estimation deals with the analysis of performance metrics of the system after the classification and prediction stages:

- Accuracy: These metric measure correct predictions against the total number of predictions. High accuracy further indicates the possibility of the system's ability to classify medical images as affected or not.
- Error Rate: This represents the fraction of incorrect predictions. A lower error rate further suggests better model reliability.

These estimations have been analyzed using:

- Confusion Matrix: It offers detailed true positives, true negatives, false positives, and false negatives.
- Validation Graphs: Plotting error and accuracy convergence in the iterations to ensure that the system generalizes.

Prediction

In the system, predictions are outputs obtained from a trained classification model:

- Prediction Process:
- The pre-processed and segmented medical image would be classified using transfer learning algorithms.
- The output tells whether a part of the medical image is affected, e.g., affected with a tumor, or not.
- Utility:

- Diagnostic aid: To identify affected areas in medical images, which forms the basis of any medical decision
- Suggestions: The system can suggest some steps or remedies on the basis of the result.

4.6 Block Chain

Blockchain Characteristics

- Basic Blockchain Structure:
  - There are blocks of messages and associated metadata, including sender, receiver, and timestamp.
  - Blocks are linked one to another through cryptographic hashes; therefore, data integrity is maintained.
- Public/Private Key System:
  - Users publish and obtain a public-private key pair.
  - Public keys can either be on the blockchain or even in the database, making it safe to share data.
- Message Encryption
  - Asymmetric encryption, such as RSA, encrypts medical images and sensitive information.
  - Only the person who intends to decrypt the data is privileged by their private key.
- Data Transfer Using Blockchain:

Once there is a transmission of a message or image, a new block will contain:

  - The encrypted data.
  - Metadata contents such as sender, receiver, and timestamp.
  - Hash keys link it to previous blocks.
  - This is traced back and shows signs that it cannot be amended.
- Message Recovery:
  - From the blockchain, recipients can query for messages sent to them.
  - Decrypt messages using their private keys.
- Secure Access
  - The blockchain uses cryptographic hashes, timestamps, and PoW as mechanisms for secure data protection.
  - Decentralized storage tends to limit potential breaches or unauthorized access by a hacker.

Implementation of the System Workflow

- Medical Image Processing:
  - Inputs: These include images like X-rays or tumors in the brain scans. The images are processed using machine learning models.
  - The results from the model, with the corresponding messages, are put on the blockchain after encryption
- User Interface through Web Application
  - Doctor User and Patient User can login, upload images and view prediction for example if it contains abnormalities.

- The blockchain ensures that the interactions and data exchanges remain private and secure.

4. Results

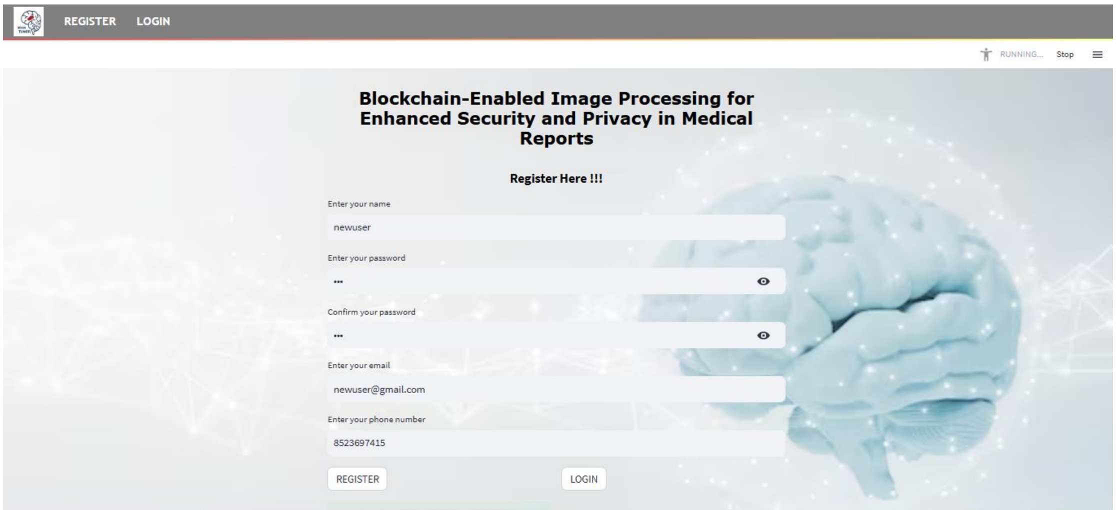
Overview of the Web Application

- A user interface through which doctors and patients can securely upload and retrieve medical reports.
- Application of Blockchain to demonstrate privacy and security.
- Use of predictive analysis in uploaded medical images like the detection of a brain tumor

Features Implemented

- User Authentication:

Registration page:



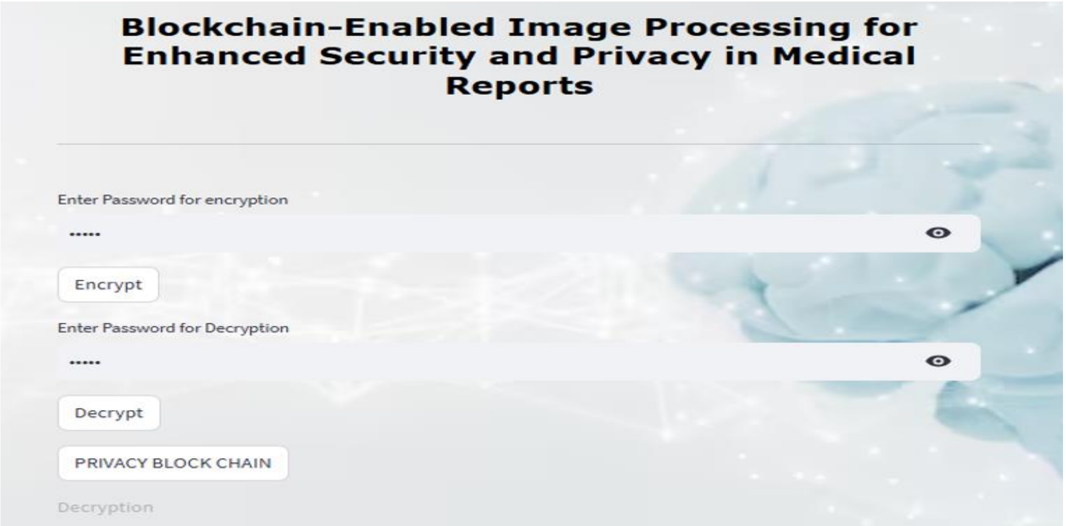
Login page:

Image Upload and Processing

Real-time prediction of affected or not affected images.

Blockchain Security:

Encrypt images and metadata of uploaded pictures.



Blockchain nodes for validating and storing secure transactions.

	Time	Data	Hash
0	2024-11-20 19:44:13	IDENTIFIED GLIOMA	a20200a94c75010576e2d6a83e6fa69271901a9d805894b28bd91
1	2024-11-20 19:44:13	IDENTIFIED GLIOMA	a20200a94c75010576e2d6a83e6fa69271901a9d805894b28bd91
2	2024-11-20 19:44:13	IDENTIFIED GLIOMA	a20200a94c75010576e2d6a83e6fa69271901a9d805894b28bd91

Image Processing Techniques

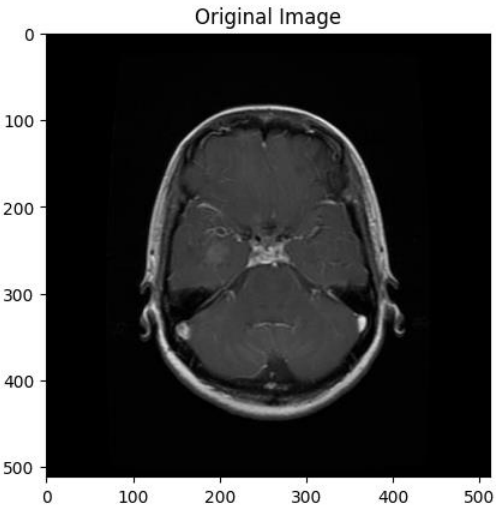


Figure 2

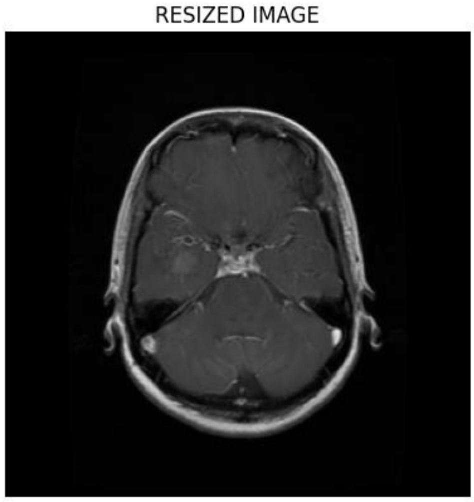


Figure 3

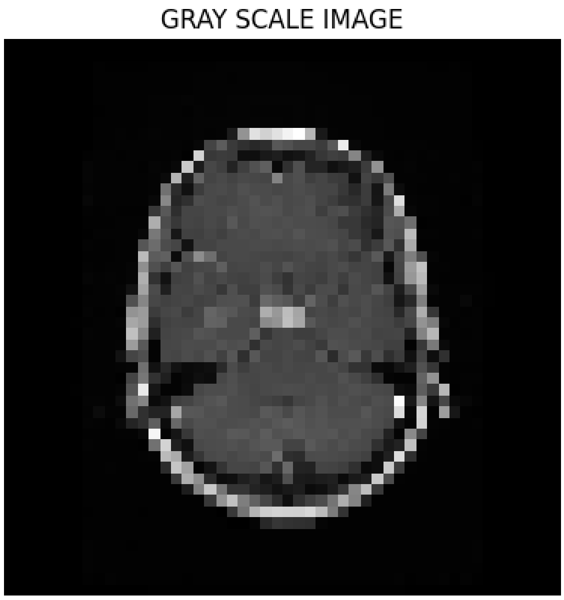


Figure 4

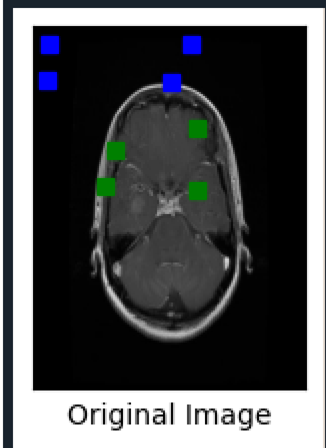
Figure 2 original medical image; in this case, an MRI scan is shown as originally captured from the imaging device. It retains native dimensions, resolution, and color format. *Figure 3* the same image resized so that its dimensions are standardized at sizes, such as 224×224 or 512×512 pixels. The scaling, therefore, allows uniformity for tasks like training machines and image analysis, keeping the essentials of the visual details of the original image intact. Finally, *Figure 4* displays the greyscale version of the image whereby all color information is removed, and the picture is represented in shades of gray. This conversion economizes the computation complexity without losing any structure details that are important for analysis or further processing.

Feature Extractions of the image

```
 ---------------------------------------------
  MEAN, VARIANCE, MEDIAN
 ---------------------------------------------

 1. Mean Value     = 22.3496

 2. Median Value   = 2.0

 3. Variance Value = 1347.39377984

 ---------------------------------------------
```

The statistical parameters of the medical image in the header section that have been computed include an average pixel intensity value given to 22.3496, a median value represented by 2.0, meaning it is the middle intensity level when the pixel values are arranged in ascending order; the variance value to 1347.39377984 that represent the spread of pixel intensities indicates variability in the image.

```
 ---------------------------------------------
  GRAY LEVEL CO-OCCURENCE MATRIX
 ---------------------------------------------
```




```
GLCM FEATURES =

[0.0, 0.0, 1.619047619047619, 6.4852607709750565]
```

Figure 5 GLCM

Figure(d) Gray Level Co-Occurrence Matrix analysis of the original image. The feature values derived from the GLCM help determine the spatial relationship of pixel intensity in an image. The feature values of the accompanying GLCM [0.0, 0.0, 1.619047619047619, 6.4852607709750565] help to quantify characteristics including contrast and homogeneity, energy, or even correlation, which could be useful for such medical image classification and diagnosis work.

```
-------------------------------------
       IMAGE SPLITTING
-------------------------------------

Total no of data        : 1311
Total no of test data   : 1048
Total no of train data  : 263
```

Dataset Preparation of how the image split into training subsets and testing subsets. There are 1311 total data points; 263 were reserved for training, while 1048 were kept for testing. This ensures a fair balance in reaching stable results toward model training or actual testing in machine learning tasks.

Estimations and Prediction

```
PERFORMANCE
-----------------------------------

1. Accuracy    = 98.49379205703735 %

2. Error Rate = 1.5062079429626465

3. Precision   = 85.71428571428571 %

4. Recall      = 92.3076923076923

5. F1-score    = 88.8888888888889
33/33 ━━━━━━━━━━━━━━━━━━━━ 9s 267ms/step
Classification Report
              precision    recall  f1-score   support

      glioma       1.00      1.00      1.00       231
  meningioma       1.00      1.00      1.00       249
     notumor       1.00      1.00      1.00       327
   pituitary       1.00      1.00      1.00       241

    accuracy                           1.00      1048
   macro avg       1.00      1.00      1.00      1048
weighted avg       1.00      1.00      1.00      1048
```

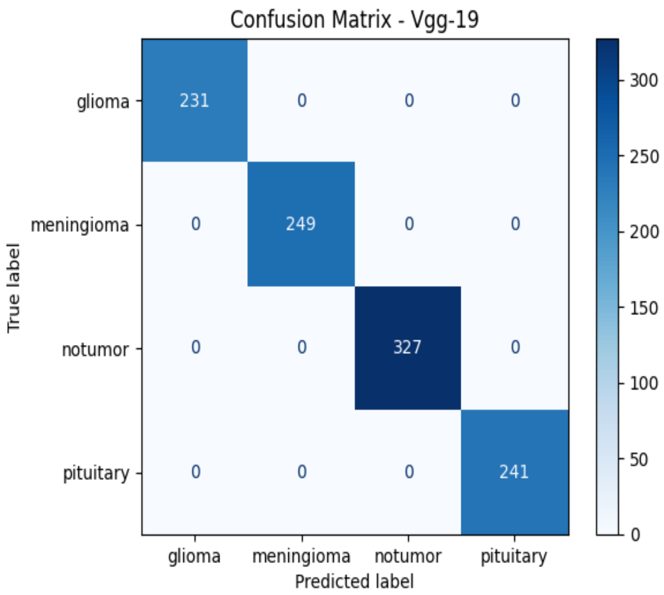


Figure 6 Accuracy of VGG-19

The performance metrics and confusion matrix indicate the correctness of the classification model achieved by VGG-19. Having an accuracy of 98.49% with an error rate of just 1.51%, the model has strong predictive skills. The classification report indicates a perfect precision, recall, and F1-score of 1.00 for all four classes: glioma, meningioma, notumor, and pituitary tumor. Moreover, the confusion matrix shows that there are no misclassifications of any sample, as all samples are correctly predicted for their respective classes, for example, 231 glioma samples predicted as glioma, 249 meningioma samples as meningioma, etc. The total dataset considered here is of 1,048 samples for all classes. The results are superb; such perfection achieved raises concerns about overfitting or problems in the dataset, and it therefore needs further validation to ensure the robustness and generalizability of the model.

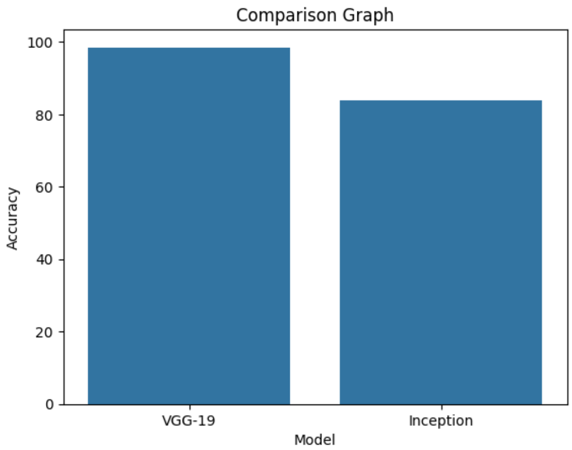


Figure 7

The comparison graph high- lights how well two deep learning models, in this case the VGG-19 and Inception, work in delivering accuracy on the same dataset. Here, VGG-19 shows better results than the latter in terms of accuracy- being close to 100%, whereas the Inception model comes about with about 85-90% of accuracy. Therefore, VGG-19 is much more likely to enhance correctness in this particular task of classifying data. However, high accuracy for VGG-19 may be overfitting or dataset-specific advantage and should be further checked to confirm the generalization capability of a model. Inception, although less accurate might be a better candidate for some contexts in terms of robustness or efficiency.

5. Conclusion and Future Work

Conclusion

The project successfully integrates the blockchain technology advanced image processing methods to achieve the highly accurate classification of images of the brain tumor. Utilizing transfer learning models VGG-19 and Inception, it wins high performance in the question of whether or not medical images are affected. Integration of the blockchain assures that sensitive medical information is being managed securely and immutably; thus the critical concern in healthcare was answered: privacy issues. Performance metrics are given in considerable detail to affirm the strength and reliability of the system, including confusion matrices and comparison graphs. Additionally, the user-friendly GUI has improved access, making it practical and convenient in use among medical professionals.

Future Work

- Generalization Across Modalities: Extend the dataset to include other medical imaging modalities such as CT and MRI to test whether the model generalizes.
- Real-Time Deployment: Develop a real-time diagnostic system that is integrated into the management systems of hospitals to be applied immediately in clinics.
- Improved Security: Enhance the scalability of blockchain, for example, through the use of advanced consensus mechanisms such as PoS or delegated PoS that scale down energy consumption.
- Multi-Model Fusion: aggregation of different outputs from various deep learning models with higher accuracy and reliability in predictions
- Explainable AI (XAI): inclusion of interpretable AI techniques so that the predictions will become explainable to the medical practitioner about the decision-making process of the system.
- Cloud and Edge Deployment: cloud or edge computing platforms for greater accessibility and scalability.
- Validation with Clinical Experts: Engage the health care staff to validate and fine-tune the system based on real-world issues and practical feedback.

6. Appendix

Image Processing Techniques

- Input Data

```
filename = askopenfilename()
img = mpimg.imread(filename)
plt.imshow(img)
plt.title("Original Image")
plt.show()
```

- Grayscale Image

```
try:
    gray11 = cv2.cvtColor(img_resize_orig, cv2.COLOR_BGR2GRAY)

except:
    gray11 = img_resize_orig

fig = plt.figure()
plt.title('GRAY SCALE IMAGE')
plt.imshow(gray11,cmap="gray")
plt.axis ('off')
plt.show()
```

- Resize Image

```
resized_image = cv2.resize(img, dsize: (300,300))
img_resize_orig = cv2.resize(img,((50, 50)))

fig = plt.figure()
plt.title('RESIZED IMAGE')
plt.imshow(resized_image)
plt.axis ('off')
plt.show()


#==== GRAYSCALE IMAGE ====

try:
    gray11 = cv2.cvtColor(img_resize_orig, cv2.COLOR_BGR2GRAY)

except:
    gray11 = img_resize_orig

fig = plt.figure()
plt.title('GRAY SCALE IMAGE')
plt.imshow(gray11,cmap="gray")
plt.axis ('off')
plt.show()
```

- Feature Extraction

```
mean_val = np.mean(gray11)
median_val = np.median(gray11)
var_val = np.var(gray11)
Test_features = [mean_val,median_val,var_val]


print()
print("---------------------------------------------")
print(" MEAN, VARIANCE, MEDIAN ")
print("---------------------------------------------")
print()
print("1. Mean Value     =", mean_val)
print()
print("2. Median Value   =", median_val)
print()
print("3. Variance Value =", var_val)

 # === GLCM ===


print()
print("---------------------------------------------")
print(" GRAY LEVEL CO-OCCURENCE MATRIX ")
print("---------------------------------------------")
print()

PATCH_SIZE = 21
```

```
image = img[:,:,0]
image = cv2.resize(image, dsize: (768,1024))

grass_locations = [(280, 454), (342, 223), (444, 192), (455, 455)]
grass_patches = []
for loc in grass_locations:
    grass_patches.append(image[loc[0]:loc[0] + PATCH_SIZE,
                               loc[1]:loc[1] + PATCH_SIZE])

# select some patches from sky areas of the image
sky_locations = [(38, 34), (139, 28), (37, 437), (145, 379)]
sky_patches = []
for loc in sky_locations:
    sky_patches.append(image[loc[0]:loc[0] + PATCH_SIZE,
                             loc[1]:loc[1] + PATCH_SIZE])

# compute some GLCM properties each patch
xs = []
ys = []
for patch in (grass_patches + sky_patches):
    glcm = graycomatrix(image.astype(int), distances=[4], angles=[0], levels=256,symmetric=True)
    xs.append(graycoprops(glcm, 'dissimilarity')[0, 0])
    ys.append(graycoprops(glcm, 'correlation')[0, 0])
```

```python
fig = plt.figure(figsize=(8, 8))

# display original image with locations of patches
ax = fig.add_subplot(3, 2, 1)
ax.imshow(image, cmap=plt.cm.gray,
          vmin=0, vmax=255)
for (y, x) in grass_locations:
    ax.plot(x + PATCH_SIZE / 2, y + PATCH_SIZE / 3, 'gs')
for (y, x) in sky_locations:
    ax.plot(x + PATCH_SIZE / 2, y + PATCH_SIZE / 2, 'bs')
ax.set_xlabel('Original Image')
ax.set_xticks([])
ax.set_yticks([])
ax.axis('image')
plt.show()

# for each patch, plot (dissimilarity, correlation)
ax = fig.add_subplot(3, 2, 2)
ax.plot(xs[:len(grass_patches)], ys[:len(grass_patches)], 'go',
        label='Region 1')
ax.plot(xs[len(grass_patches):], ys[len(grass_patches):], 'bo',
        label='Region 2')
ax.set_xlabel('GLCM Dissimilarity')
ax.set_ylabel('GLCM Correlation')
ax.legend()
plt.show()
```

```python
sky_patches0 = np.mean(sky_patches[0])
sky_patches1 = np.mean(sky_patches[1])
sky_patches2 = np.mean(sky_patches[2])
sky_patches3 = np.mean(sky_patches[3])

Glcm_fea = [sky_patches0,sky_patches1,sky_patches2,sky_patches3]
Tesfea1 = []
Tesfea1.append(Glcm_fea[0])
Tesfea1.append(Glcm_fea[1])
Tesfea1.append(Glcm_fea[2])
Tesfea1.append(Glcm_fea[3])


print()
print("GLCM FEATURES =")
print()
print(Glcm_fea)
```

- Image Splitting

```
import os

from sklearn.model_selection import train_test_split


data_glioma = os.listdir('Data/glioma/')
data_menign = os.listdir('Data/meningioma/')
data_non = os.listdir('Data/notumor/')
data_pit = os.listdir('Data/pituitary/')

# ------

dot1= []
labels1 = []
for img11 in data_glioma:
        # print(img)
        img_1 = mpimg.imread('Data/glioma//' + "/" + img11)
        img_1 = cv2.resize(img_1,((50, 50)))


        try:
            gray = cv2.cvtColor(img_1, cv2.COLOR_BGR2GRAY)

        except:
            gray = img_1


        dot1.append(np.array(gray))
        labels1.append(1)
```

```
for img11 in data_menign:
        # print(img)
        img_1 = mpimg.imread('Data/meningioma//' + "/" + img11)
        img_1 = cv2.resize(img_1,((50, 50)))


        try:
            gray = cv2.cvtColor(img_1, cv2.COLOR_BGR2GRAY)

        except:
            gray = img_1


        dot1.append(np.array(gray))
        labels1.append(2)

for img11 in data_non:
        # print(img)
        img_1 = mpimg.imread('Data/notumor//' + "/" + img11)
        img_1 = cv2.resize(img_1,((50, 50)))


        try:
            gray = cv2.cvtColor(img_1, cv2.COLOR_BGR2GRAY)

        except:
            gray = img_1


        dot1.append(np.array(gray))
        labels1.append(3)
```

```python
for img11 in data_pit:
        # print(img)
        img_1 = mpimg.imread('Data/pituitary//' + "/" + img11)
        img_1 = cv2.resize(img_1,((50, 50)))


        try:
            gray = cv2.cvtColor(img_1, cv2.COLOR_BGR2GRAY)

        except:
            gray = img_1


        dot1.append(np.array(gray))
        labels1.append(4)


x_train, x_test, y_train, y_test = train_test_split(dot1,labels1,test_size = 0.2, random_state = 101)

print()
print("------------------------------------")
print("       IMAGE SPLITTING              ")
print("------------------------------------")
print()


print("Total no of data        :",len(dot1))
print("Total no of test data   :",len(x_train))
print("Total no of train data  :",len(x_test))
```

Block Chain Integration

```python
pred=pred_lr
import hashlib
from datetime import datetime
previous_hash1 =[]
timestamp1 = []
data1 = []
hash1 = []
class Block:  2 usages

    def __init__(self,  data, previous_hash):
        self.timestamp =  datetime.now()
        self.data = pred
        self.previous_hash = previous_hash
        self.hash = self.calc_hash()
        self.next = None
```

```python
    def calc_hash(self):  1 usage
        hash_str = "We are going to encode this string of data!".encode('utf-8')
        return hashlib.sha256(hash_str).hexdigest()

class Blockchain:  1 usage

    def __init__(self):
        self.head = None
        self.next = None

    def add_block(self, data):  3 usages

        if self.head == None:
            self.head = Block(data, previous_hash: 0)

        else:
            current = self.head

          # loop to the last node of the linkedlist
            while current.next:
                current = current.next

            # stores the previous has for the next block
            previous_hash = current.hash
            current.next = Block(data, previous_hash)
```

```python
    def print_blockchain(self):  1 usage

        if self.head == None:
            print("The blockchain is empty")
            return

        else:
            current = self.head
            while current:
                print("Timestamp:", current.timestamp)
                print("Data:", current.data)
                print("Hash:", current.hash)
                print("Previous hash:", current.previous_hash)
                print("--------------->")
                timestamp1.append(current.timestamp)
                data1.append(current.data)
                hash1.append(current.hash)
                previous_hash1.append(current.previous_hash)

                current = current.next

bitcoin = Blockchain()

bitcoin.add_block("block 1")
bitcoin.add_block("block 2")
bitcoin.add_block("block 3")
bitcoin.print_blockchain()
```

```python
import pandas as pd
dframe = pd.DataFrame()
dframe['Time'] = timestamp1
dframe['Data'] = data1
dframe['Hash'] = hash1
dframe['Previous Hash'] = previous_hash1


st.write(dframe)
```

Access Page

```python
import streamlit as st
import sqlite3
import re

# Function to create a database connection
def create_connection(db_file):  1 usage
    conn = None
    try:
        conn = sqlite3.connect(db_file)
    except sqlite3.Error as e:
        print(e)
    return conn

# Function to create a new user
def create_user(conn, user):  1 usage
    sql = ''' INSERT INTO users(name, password, email, phone)
              VALUES(?,?,?,?) '''
    cur = conn.cursor()
    cur.execute(sql, user)
    conn.commit()
    return cur.lastrowid

# Function to check if a user already exists
def user_exists(conn, email):  1 usage
    cur = conn.cursor()
    cur.execute("SELECT * FROM users WHERE email=?", (email,))
    if cur.fetchone():
        return True
    return False
```

```python
def validate_email(email):  1 usage
    pattern = r'^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$'
    return re.match(pattern, email)

# Function to validate phone number
def validate_phone(phone):  1 usage
    pattern = r'^[6-9]\d{9}$'
    return re.match(pattern, phone)

# Main function
def main():  1 usage
    # st.title("User Registration")

    # Create a database connection
    conn = create_connection("dbs.db")

    if conn is not None:
        # Create users table if it doesn't exist
        conn.execute('''CREATE TABLE IF NOT EXISTS users
                     (id INTEGER PRIMARY KEY,
                     name TEXT NOT NULL,
                     password TEXT NOT NULL,
                     email TEXT NOT NULL UNIQUE,
                     phone TEXT NOT NULL);''')
```

```python
name = st.text_input("Enter your name")
password = st.text_input("Enter your password", type="password")
confirm_password = st.text_input("Confirm your password", type="password")
email = st.text_input("Enter your email")
phone = st.text_input("Enter your phone number")

col1, col2 = st.columns(2)

with col1:

    aa = st.button("REGISTER")

    if aa:

        if password == confirm_password:
            if not user_exists(conn, email):
                if validate_email(email) and validate_phone(phone):
                    user = (name, password, email, phone)
                    create_user(conn, user)
                    st.success("User registered successfully!")
                else:
                    st.error("Invalid email or phone number!")
            else:
                st.error("User with this email already exists!")
        else:
            st.error("Passwords do not match!")
```

```
                conn.close()
                # st.success('Successfully Registered !!!')
            # else:

                # st.write('Registeration Failed !!!')

        with col2:

            aa = st.button("LOGIN")

            if aa:
                import subprocess
                subprocess.run(['python','-m','streamlit','run','Login.py'])



if __name__ == '__main__':
    main()
```