VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI - 590 018

A PROJECT REPORT

on

"INTRACRANIAL HEMORRHAGE DETECTION"

Submitted by

Shetty Yajnesh Chandrashekar	4SF19CS148
Nagraj S Rao	4SF19CS100
Achinth Hegde	4SF19CS008
Saibaz Shaikh	4SF20CS411

In partial fulfillment of the requirements for the award of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE & ENGINEERING

Under the Guidance of

Mrs. Srividya

Assistant Professor, Department of CSE

at

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

2018 - 19

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project entitled **"INTRACRANIAL HEMORRHAGE DETECTION"** has been carried out by **Shetty Yajnesh Chandrashekar (4SF19CS148) , Nagraj S Rao (4SF19CS100), Achinth Hegde (4SF19CS008) and Saibaz Shaikh (4SF20CS411)**, the bonafide students of Sahyadri College of Engineering & Management in partial fulfillment for the award of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belagavi during the year 2022 - 23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Signature of the Guide	**Signature of the HOD**	**Signature of the Principal**
Mrs. Srividya	Dr. Nagesh H R	Dr. Rajesha S

External Viva:

Examiner's Name	Signature with Date
1.	.....
2.	.....

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

Department of Computer Science & Engineering

DECLARATION

We hereby declare that the entire work embodied in this Project Report titled **"INTRACRANIAL HEMORRHAGE DETECTION"** has been carried out by us at Sahyadri College of Engineering and Management, Mangaluru under the supervision of **Mrs. Srividya,** Asssistant professor for the award of **Bachelor of Engineering** in **Computer Science & Engineering**. This report has not been submitted to this or any other University for the award of any other degree.

Shetty Yajnesh Chandrashekar (4SF19CS148)

Nagraj S Rao (4SF19CS100)

Achinth Hegde (4SF19CS008)

Saibaz Shaikh (4SF20CS411)

Dept. of CS&E, SCEM, Mangaluru

Abstract

The proposed approach for intracranial hemorrhage detection using computed tomography (CT) scans involves a multi-step process. Firstly, the CT scan images of the intracranial undergo pre-processing to enhance their quality and eliminate noise or artifacts. This ensures that the subsequent analysis is performed on reliable and clean data. The pre-processed images are then segmented to isolate the intracranial region of interest from the background, enabling focused analysis on the relevant area.

Following segmentation, feature extraction is conducted to capture key characteristics and patterns that differentiate between hemorrhage and non-hemorrhage cases. Various features such as texture descriptors, shape descriptors, and intensity statistics are extracted from the segmented intracranial region. These features provide quantitative information that is crucial for accurate classification.

The final step involves the use of a Random Forest classifier. Random Forest is a machine learning algorithm that utilizes an ensemble of decision trees. The classifier is trained on a labeled dataset of CT scan images with known hemorrhage or non-hemorrhage cases. By learning from this dataset, the Random Forest classifier can predict the presence of hemorrhage in new, unseen CT scan images based on the extracted features.

By employing this computer-aided diagnosis system, medical professionals can benefit from efficient and accurate intracranial hemorrhage detection. The automated analysis of CT scan images, leveraging image processing techniques, feature extraction, and the Random Forest classifier, facilitates prompt diagnosis, enabling timely interventions and improved patient outcomes.

Acknowledgement

It is with great satisfaction and euphoria that we are submitting the Project Report on **"INTRACRANIAL HEMORRHAGE DETECTION"**. We have completed it as a part of the curriculum of Visvesvaraya Technological University, Belagavi for the award of Bachelor of Engineering in Computer Science & Engineering.

We are profoundly indebted to our guide, **Mrs. Srividya**, Asst Professor, Department of Computer Science & Engineering for innumerable acts of timely advice, encouragement and We sincerely express our gratitude.

We also thank **Mr. Shuhas Bhyrate**, **Ms. Parashakthi** and **Mrs. Pooja N S**,Assistant Professor Project Coordinators, Department of Computer Science & Engineering for their constant encouragement and support extended throughout.

We express our sincere gratitude to **Dr. Nagesh H R**, Head & Professor, Department of Computer Science & Engineering for his invaluable support and guidance.

We sincerely thank **Dr. Rajesha S**, Principal, Sahyadri College of Engineering & Management and **Dr. Umesh M. Bhushi**, Director Strategic and Planning, Sahyadri College of Engineering & Management,who have always been a great source of inspiration.

Finally, yet importantly, We express our heartfelt thanks to our family & friends for their wishes and encouragement throughout the work.

Shetty Yajnesh Chandrashekar (4SF19CS148)

Nagraj S Rao (4SF19CS100)

Achinth Hegde (4SF19CS008)

Saibaz Shaikh (4SF20CS411)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

intracranial hemorrhage is a critical medical condition that can lead to severe consequences, including disability or even death. Early and accurate detection of intracranial hemorrhage plays a vital role in facilitating prompt medical intervention, enhancing patient outcomes, and potentially saving lives. With advancements in medical imaging technology and the growing availability of digital medical records, computer-aided diagnosis systems have emerged as valuable tools for assisting healthcare professionals in the detection and diagnosis of intracranial hemorrhage. This research focuses on the utilization of image processing techniques, including Gray-Level Co-occurrence Matrix (GLCM), Histogram of Oriented Gradients (HOG), Gabor filters, and Local Binary Patterns (LBP), for feature extraction, combined with Random Forest for classification.

The manual analysis of intracranial images, particularly computed tomography (CT) scans, is a complex and time-consuming task that requires a high level of expertise. Radiologists meticulously examine the images to identify hemorrhagic regions, often relying on their visual interpretation and experience. However, this process is subjective, prone to human errors, and can be challenging when dealing with large volumes of medical data. The proposed computer-aided diagnosis system aims to address these challenges and provide a more efficient and accurate alternative to manual analysis.

1.1 Purpose

The purpose of this research is to develop an automated system for intracranial hemorrhage detection using image processing techniques and machine learning algorithms. By leveraging the power of GLCM, HOG, Gabor filters, and LBP, combined with Random Forest classification, the system aims to extract relevant features from CT scan images

and accurately classify them as hemorrhagic or non-hemorrhagic. The ultimate goal is to provide healthcare professionals with a reliable and time-efficient tool that can assist in the early detection of intracranial hemorrhage, leading to timely interventions and improved patient care.

1.2 Scope

The scope of this research encompasses various aspects related to the development and evaluation of the proposed computer-aided diagnosis system. This includes:

Image Pre-processing: The CT scan images undergo pre-processing techniques such as noise reduction, contrast enhancement, and normalization to improve image quality and facilitate accurate feature extraction.

Feature Extraction: The system employs GLCM to capture textural features that describe the spatial relationships between pixel intensities. HOG descriptors are used to analyze local gradients and extract shape and structural information. Gabor filters are applied to extract frequency and orientation features, while LBP descriptors capture local texture patterns within the images.

Classification: The extracted features are used as input to a Random Forest classifier, which is trained on a labeled dataset of CT scan images. The classifier learns the patterns and characteristics associated with hemorrhagic and non-hemorrhagic regions, enabling it to make accurate predictions on unseen images.

1.3 Overview

The proposed computer-aided diagnosis system for intracranial hemorrhage detection utilizes advanced image processing techniques, including GLCM (Gray Level Co-occurrence Matrix), HOG (Histogram of Oriented Gradients), Gabor filters, and LBP (Local Binary Patterns). These techniques play a crucial role in extracting relevant features from CT scan images of the intracranial.

By applying GLCM, the system captures the statistical properties of pixel intensities and their spatial relationships within the image. This allows for the analysis of textural patterns that may indicate the presence of hemorrhagic regions. HOG, on the other hand, focuses on capturing local gradients and edge orientations, providing valuable shape and structural information. Gabor filters are utilized to extract frequency and texture features,

enabling the system to detect subtle patterns related to intracranial hemorrhage. Lastly, LBP examines the local binary patterns of pixel neighborhoods, revealing characteristic patterns that can differentiate between hemorrhagic and non-hemorrhagic regions.

These image processing techniques work as a combination to create a comprehensive set of features that describe different aspects of the intracranial images. The combination of GLCM, HOG, Gabor filters, and LBP allows for a holistic representation of the hemorrhagic and non-hemorrhagic regions within the intracranial. This comprehensive feature set enhances the accuracy and discriminative power of the subsequent classification process.

The classification stage utilizes the Random Forest algorithm, a powerful ensemble learning method. Random Forest constructs a multitude of decision trees and combines their predictions to make the final classification decision. This algorithm is well-suited for medical diagnosis tasks due to its ability to handle high-dimensional feature spaces and effectively address class imbalances. Random Forest can learn complex relationships between the extracted features and the presence of hemorrhage, resulting in accurate classification of new, unseen CT scan images.

Chapter 2

Literature Survey

This chapter gives the details of the various works which were carried out for scalable data sharing in Intracranial Hemorrhage detection.

In [1] the researchers have proposed an innovative approach that combines morphology and threshold segmentation techniques to automatically detect and locate bleeding points in MRI images. This method takes advantage of the unique characteristics and information provided by different sequences of MRI. The algorithm can effectively identify and extract the regions of interest related to bleeding. Using morphology, which deals with the spatial arrangement and shape of objects, the method applies various operations such as erosion, dilation, and opening to enhance the visibility and distinctness of the bleeding regions in the image. This helps in refining the boundaries of the bleeding points and preparing them for further analysis. Threshold segmentation, on the other hand, involves setting specific intensity thresholds to separate the bleeding regions from the surrounding tissues. By applying an appropriate thresholding technique, the algorithm can accurately segment the bleeding points based on their intensity values in the MRI images. This step allows for a more precise calculation of the location and size of the bleeding point. By combining these two techniques, the proposed method achieves a synergistic effect, leveraging the strengths of both morphology and threshold segmentation. This enables the algorithm to effectively detect and localize bleeding points in the MRI images, providing valuable information for medical professionals.

In [2], The researchers have come up with a brilliant approach to address the challenge of limited availability of extensive datasets in critical cases. They proposed a method that harnesses the abstraction capabilities of deep learning for intracranial hemorrhage classification. This method recognizes that in some situations, it may not be possible to gather a large dataset of intracranial hemorrhage images for training a neural network

model. To overcome this limitation, the researchers employed two key techniques: image augmentation and dataset imbalancing. Image augmentation involves applying various transformations and modifications to the existing dataset to generate additional diverse images. This technique helps expand the dataset and increase its variability, enhancing the model's ability to learn and generalize from a limited set of images.Dataset imbalancing refers to addressing any class imbalance issues in the available dataset. In the context of intracranial hemorrhage classification, it is common for the number of hemorrhage cases to be significantly smaller than the non-hemorrhage cases. The researchers adopted strategies to balance the dataset, ensuring that the neural network model receives adequate training on both hemorrhage and non-hemorrhage samples. With the augmented and balanced dataset, the researchers designed a unique architecture using a convolutional neural network (CNN) model. CNNs are well-suited for image-based tasks, as they can effectively capture and learn intricate patterns and features from the input images. The researchers tailored the CNN model specifically for intracranial hemorrhage classification, taking advantage of its ability to automatically extract relevant features from the images.

In[3]The researchers have come up with an enhanced U-Net neural network architecture specifically designed for the auxiliary diagnosis of intracerebral hemorrhage. Their goal was to automate the segmentation of hemorrhage regions on CT images, which is crucial for accurate diagnosis and treatment planning. By utilizing the advanced capabilities of deep learning, the improved U-Net model achieves automatic segmentation of hemorrhage areas in CT images. This development has the potential to significantly enhance the efficiency and accuracy of diagnosing intracerebral hemorrhage, ultimately improving patient outcomes. The proposed method represents a promising advancement in the field of medical image analysis and diagnosis.

In [4] The researchers have proposed a method for normalizing 3D volumetric scans, specifically focusing on head CT scans from the CQ500 dataset. The purpose of this method is to enhance the contrast around the abnormal region of interest in the scan, thereby aiding the performance of convolutional neural networks (CNNs) in detecting and analyzing abnormalities. The method utilizes the intensity profile of the training samples to establish a normalization strategy. By analyzing the intensity distribution of the training data, the method adjusts the intensities of the volumetric scans to create a higher contrast specifically around the abnormal region. This normalization process aims to highlight the abnormality and improve the discriminative power of the CNN in identifying and segmenting such regions accurately. By employing this normalization technique, the pro-

posed method enhances the overall performance of CNNs in analyzing head CT scans for abnormalities. It allows the neural network to better distinguish abnormal regions from the surrounding tissues, leading to more reliable and accurate diagnostic results. This development has the potential to significantly improve the efficiency and effectiveness of medical image analysis, assisting healthcare professionals in making informed decisions and providing better patient care.

R. S. Sandhu has explained about the Cryptographic key assignment schemes in [10], which aim to minimize the expense in storing and managing secret keys for general cryptographic use. Utilizing a tree structure, a key for a given branch can be used to derive the keys of its descendent nodes. Just granting the parent key implicitly grants all the keys of its descendent nodes. The author proposed a method to generate a tree hierarchy of symmetric keys by using repeated evaluations of pseudorandom function/block-cipher on a fixed secret. The concept can be generalized from a tree to graph. More advanced cryptographic key assignment schemes support access policy that can be modelled by an acyclic graph or a cyclic graph.

In [5]The auother reviewed various research papers were reviewed that employed deep learning algorithms such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for hemorrhage detection. These approaches utilized annotated datasets to train the models and achieved promising results in terms of accuracy and performance. Additionally, the survey highlighted the use of a windowing technique inspired by the practical approach of radiologists. This technique involved applying different window settings to the images to enhance the visibility of hemorrhage regions. By adjusting the window level and width, the models were able to capture subtle differences in the intensity values associated with hemorrhage. The experimental results from the surveyed papers demonstrated the effectiveness of deep learning approaches in automatically detecting and classifying acute intracranial hemorrhage. These findings contribute to the ongoing research efforts in developing reliable and efficient computer-aided diagnosis systems for hemorrhage detection, providing valuable insights for future studies and advancements in this field.

Chapter 3

Problem Definition

intracranial hemorrhage is a critical form of intracranial stroke that requires prompt and accurate detection for effective medical intervention. Inaccuracies in detecting hemorrhage can lead to serious consequences for the patient's health and well-being.

To address this challenge, our system is designed to provide a reliable and precise method for detecting intracranial hemorrhage. By leveraging advanced image processing techniques and machine learning algorithms, our system aims to improve the accuracy of hemorrhage detection, thereby minimizing the risks associated with misdiagnosis or delayed treatment.

The system utilizes state-of-the-art technologies to analyze medical images, such as computed tomography (CT) scans, with a high degree of accuracy. By processing these images and identifying specific patterns and characteristics indicative of hemorrhage, the system can provide healthcare professionals with more reliable and precise results.

The benefits of our system include enhanced efficiency and reduced human error in detecting intracranial hemorrhage. By automating the detection process, our system minimizes the reliance on manual interpretation, which can be subjective and prone to inconsistencies. This, in turn, enables healthcare professionals to make more informed decisions and provide timely and appropriate treatment to patients.

Overall, our system aims to improve the accuracy and reliability of intracranial hemorrhage detection, ultimately leading to better patient outcomes and reducing the potential risks associated with inaccurate diagnoses. By leveraging advanced technology and cutting-edge algorithms, we strive to provide healthcare professionals with a valuable tool that enhances their ability to detect and manage intracranial hemorrhage effectively.

Chapter 4

Software Requirements Specification

4.1 Introduction

Software Requirement Specification totally defines how the projected software behaves without unfolding how the software will perform it. This document describes the requirements for a software system that can detect intracranial hemorrhages (intracranial hemorrhage detection) from medical images. The system will be developed using Python, Django, Keras, Visual Studio Code, HTML, JavaScript, MySQL, and Microsoft Windows XP/Windows 7.

4.2 Purpose

The purpose of this system is to provide a rapid and accurate way to detect intracranial hemorrhage detection in medical images. This will help doctors to diagnose and treat intracranial hemorrhage detection early, which can improve patient outcomes.

4.3 User Characteristics

The target users of this system are doctors and other healthcare professionals who need to detect intracranial hemorrhage detection in medical images. The system will be easy to use and will require no prior experience with programming or machine learning.

4.4 Interfaces

4.4.1 Hardware Interfaces

- Processor : Intel Core i5 1.6GHz
- RAM : 8GB
- Hard Disk : 500GB
- Input Device : Standard keyboard and Mouse
- Output Device : Monitor

4.4.2 Software Interfaces

- Database : Sqlyog Enterprise.
- Programming Language : Python, HTML.
- Server :NETBEANS
- Framework: Django
- Libraries: NumPy, Pandas, Matplotlib, Keras.
- IDE: Visual Studio 4.0 , Jupyter Notebook

4.5 Functional Requirements

- **Loading Images:**

  Load medical images from a variety of formats.The system should be able to load medical images from a variety of formats, including: TIFF, JPEG, PNG, DICOM, NIFTI, MINC

  The system should also be able to handle images that are stored in a variety of locations, such as:Local file system, Network file system, Cloud storage The system should also be able to handle images that are of different sizes and resolutions.

- **Feature extraction:**

Extract features from the images. The system should be able to extract features from the images in order to train a machine learning model to classify them. Some of the features that the system could extract include:

Intensity: The brightness of each pixel in the image.

Color: The hue, saturation, and value of each pixel in the image.

Texture: The spatial arrangement of pixels in the image.

Shape: The geometric properties of objects in the image.

Location: The position of objects in the image.

- **Classification:**

  Train a machine learning model to classify the images as either containing an ICH or not containing an intracranial hemorrhage detection.

- **Prediction:**

  Predict whether a new image contains an intracranial hemorrhage detection.

- **Interface Development:**

  Provide a user interface that allows doctors to interact with the system.

4.6 Non-Functional Requirements

The system must meet the following non-functional requirements:

- **Accuracy:**

  The system must be able to accurately detect intracranial hemorrhage detection.

- **Speed:**

  The system must be able to process images quickly.

- **Scalability:**

  The system must be able to handle large volumes of data.

- **Security:**

  The system must be secure and protect patient data.

Chapter 5

System Design

5.1 Architecture Design

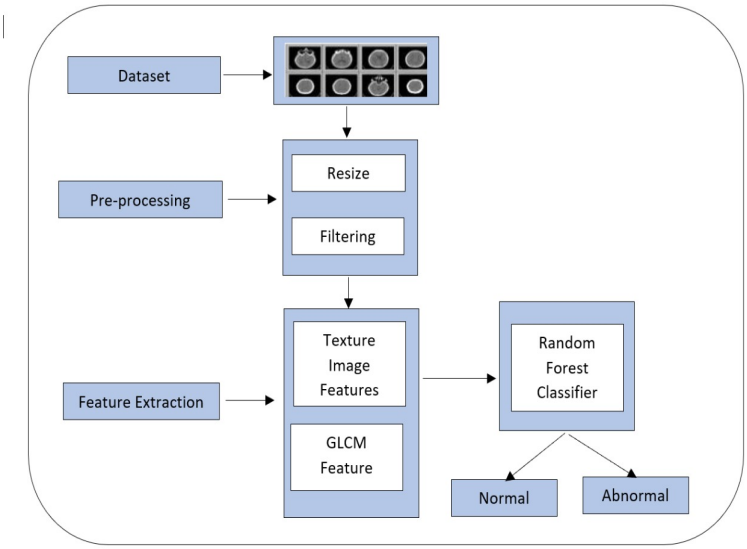

Figure 5.1: System Architecture Diagram

After pre-processing the collected dataset, which involves techniques such as noise reduction and image enhancement, feature extraction algorithms are applied to the images. These algorithms analyze the images and extract a set of features that capture important characteristics related to intracranial hemorrhage. These features can include information about texture, shape, intensity, and other relevant properties.

Once the feature extraction step is completed, the resulting set of features serves as input for the classification process. Random Forest, is employed to classify the extracted features into categories, namely hemorrhage or non-hemorrhage. These classification algorithms learn from labeled data to recognize patterns and make predictions on new, unseen images.

The combination of pre-processing, feature extraction, and classification allows for the detection and classification of intracranial hemorrhage in the collected dataset. It enables healthcare professionals to analyze the images effectively and make informed decisions based on the classification results.

5.2 Decomposition Description

The decomposition description outlines the functions and responsibilities of the admin and user entities in the system. The admin entity collects and preprocesses the dataset, trains the model, saves the model, and creates the feature extraction algorithm. The user entity handles user-related functionalities such as registration, login, image upload, and result display. This decomposition provides a clear understanding of the system's structure and the distribution of tasks among different entities.

User Entity:

User Registration Function: The user entity allows users to register in the system. This function involves capturing user registration details, validating the input, and storing the user's data in the system. Upon successful registration, a success message is returned to the user.

User Login Function: The user entity enables users to log in to the system. This function verifies the user's credentials, checks the database for authentication, and grants access to system features upon successful login.

Upload Image Function: The user entity provides the functionality for users to upload an image for hemorrhage detection. This function allows users to select an image file, send it to the system, and initiate the classification process.

Display Result Function: After the image is classified, the user entity is responsible for displaying the classification result to the user. This function involves retrieving the result from the system and presenting it to the user in a user-friendly format.

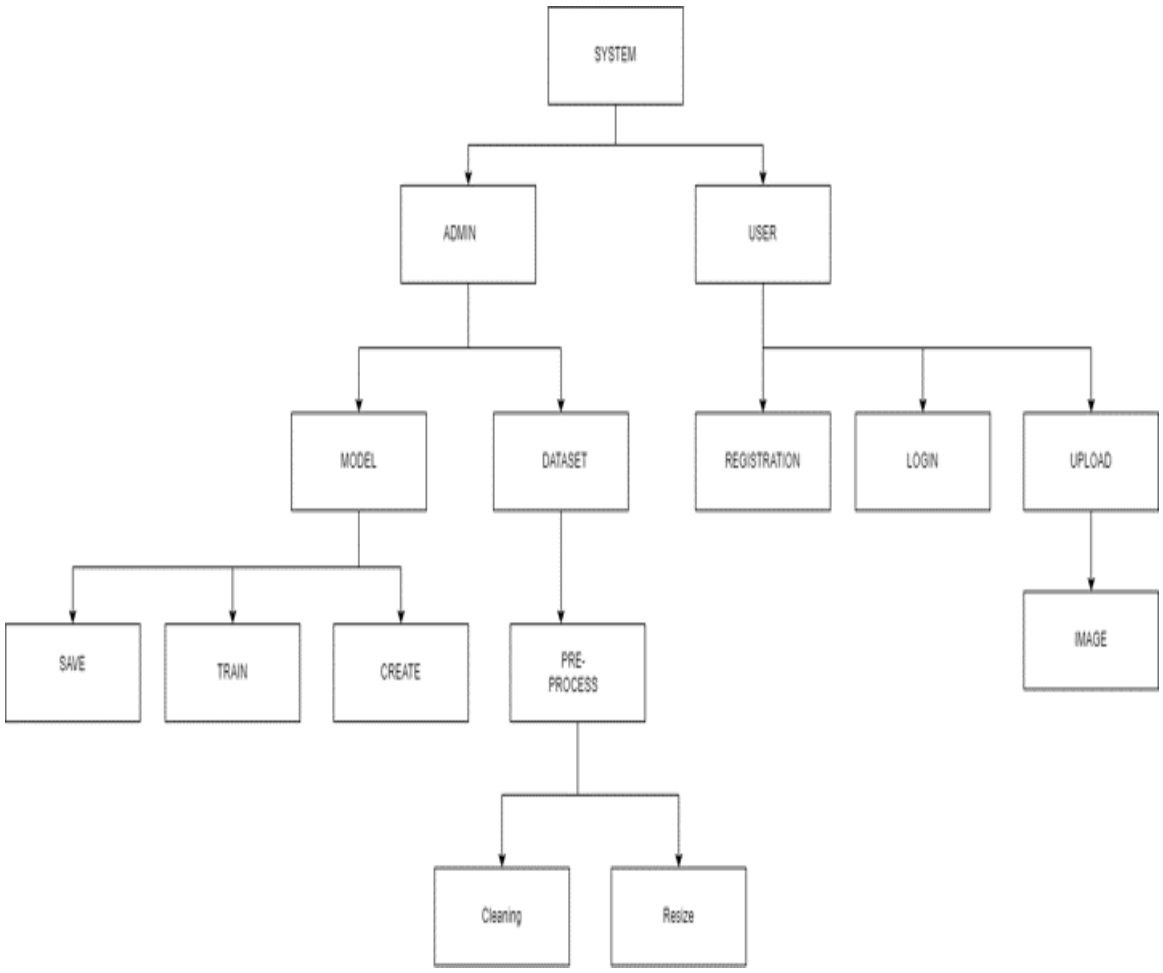

Figure 5.2: Decomposition Description diagram

5.3 Data Flow Design

The dataflow diagram illustrates the flow of data and processes involved in the detection and classification of hemorrhagic and non-hemorrhagic data.

5.3.1 Hemorrhagic and Non-Hemorrhagic Data Split:

Input: Hemorrhagic and non-hemorrhagic data

Process: The data is split into two categories: hemorrhagic and non-hemorrhagic. This categorization helps in distinguishing between the two types of images.

Output: Hemorrhagic data and non-hemorrhagic data

5.3.2 Resize:

Input: Hemorrhagic and non-hemorrhagic data

Process: The images are resized to a standard size. Resizing ensures that all images have the same dimensions, which is essential for further processing and feature extraction.

Output: Resized hemorrhagic and non-hemorrhagic data

5.3.3 Normalize:

Input: Resized hemorrhagic and non-hemorrhagic data

Process: The pixel values of the images are normalized. Normalization helps in bringing the pixel values within a specific range, making them suitable for subsequent processing steps.

Output: Normalized hemorrhagic and non-hemorrhagic data

5.3.4 Threshold:

Input: Normalized hemorrhagic and non-hemorrhagic data

Process: A thresholding technique is applied to the images. This process converts the grayscale images into binary images by classifying pixels as hemorrhagic or non-hemorrhagic based on a specified threshold value.

Output: Thresholded hemorrhagic and non-hemorrhagic data

5.3.5 Contouring:

Input: Thresholded hemorrhagic and non-hemorrhagic data

Process: The contours of the hemorrhagic and non-hemorrhagic regions are extracted from the thresholded images. Contouring helps in identifying the boundaries of the regions of interest.

Output: Contoured hemorrhagic and non-hemorrhagic data

5.3.6 Cropping:

Input: Contoured hemorrhagic and non-hemorrhagic data

Process: The regions of interest containing hemorrhagic and non-hemorrhagic areas are cropped from the original images based on the extracted contours. Cropping isolates the relevant areas for feature extraction and classification.

Output: Cropped hemorrhagic and non-hemorrhagic data

5.3.7 Feature Extraction:

Input: Cropped hemorrhagic and non-hemorrhagic data Process: Feature extraction algorithms such as GLCM, HOG, Gabor, LBP are applied to the cropped images. These algorithms compute descriptive features from the images, capturing important characteristics related to hemorrhage. Output: Extracted features from hemorrhagic and non-hemorrhagic data

5.3.8 Classification:

Input: Extracted features from hemorrhagic and non-hemorrhagic data

Process: The extracted features are used as input to a classification algorithm, such as Random Forest. The classifier analyzes the features and determines whether the image represents a hemorrhagic or non-hemorrhagic condition.

Output: Classification result (hemorrhagic or non-hemorrhagic)

5.3.9 Output:

Input: Classification result

Process: The classification result is displayed or communicated to the user or system for further processing or visualization.

Output: Classification output

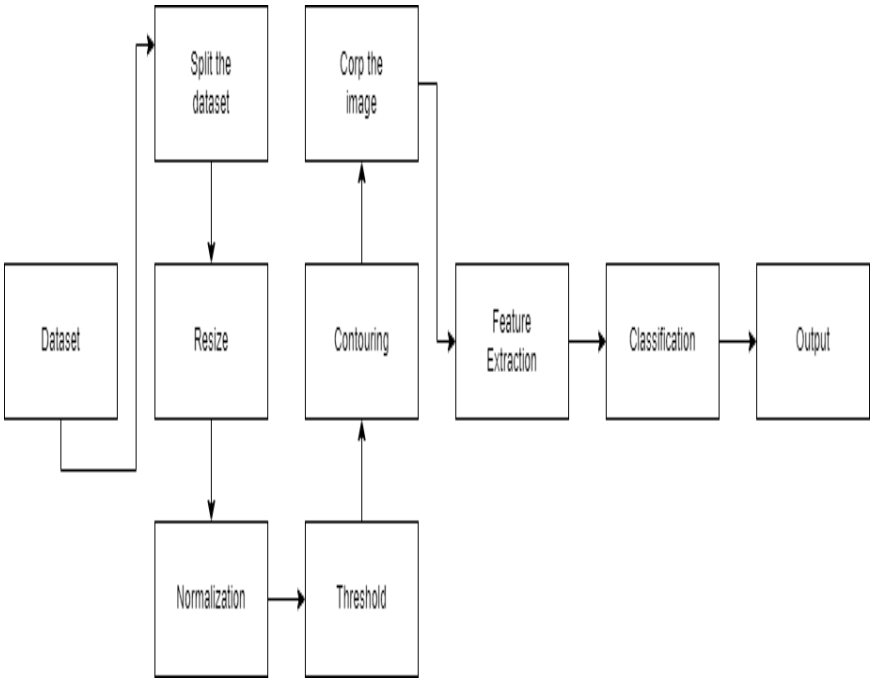

Figure 5.3: Data Flow Design

5.4 Sequence Diagram

User Registration Sequence:

The user sends a registration request to the system. The system receives the registration request and validates the user's input. The system checks the database to ensure that the user does not already exist. The system saves the user's registration data in the database. The system generates a success message indicating that the user is successfully registered. The success message is sent back to the user.

User Login Sequence:

The user sends a login request to the system. The system receives the login request and validates the user's credentials. The system checks the database to verify the user's information. If the user's credentials are valid, the system generates a success message indicating that the user is logged in. The success message is sent back to the user.

Image Classification Sequence:

The user sends an image upload request to the system. The system receives the image upload request and verifies the user's authentication. The user's credentials are checked to ensure they are logged in and authorized to upload images. The system accepts the uploaded image and performs the necessary preprocessing steps. The system applies the trained model to classify the image for hemorrhage detection. The result of the classification is obtained. The system generates a response message containing the classification result. The response message is sent back to the user.

In this sequence diagram, the user interacts with the system by sending requests for registration, login, and image upload. The system processes these requests, validates user input and credentials, performs database operations for user registration and login, and executes the image classification process. The system communicates with the database to store and retrieve user data. The sequence diagram illustrates the flow of interactions between the user, system, and database, providing an overview of the steps involved in user registration, login, and image classification processes.

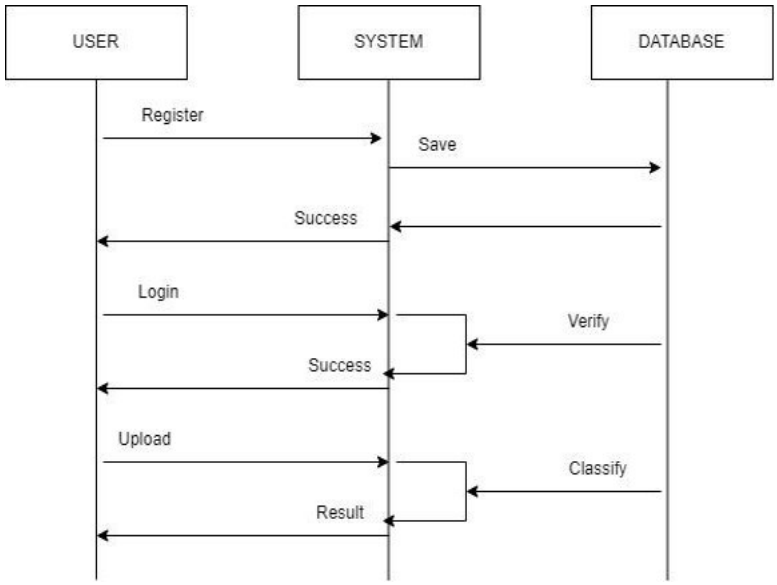

Figure 5.4: Sequence diagram for the proposed system

5.5 Use case Diagram

A use case diagram is a visual representation of the interactions between actors (users or external systems) and the system itself. In this case, we have two main actors: the admin and the user. The admin performs tasks related to data preprocessing, model creation, feature extraction, and saving the model. The user, on the other hand, interacts with the system by inputting an image to be classified and obtaining the result. Here's a description of the use case diagram:

Admin Use Cases:

Preprocess Data: The admin is responsible for preprocessing the collected dataset. This involves tasks such as data cleaning, normalization, image enhancement, and any other necessary steps to prepare the data for further processing.

Model Creation: The admin creates a model for hemorrhage detection. This includes selecting an appropriate machine learning algorithm, training the model using the prepro-

cessed data, and evaluating its performance.

Feature Extraction: The admin applies feature extraction algorithms (such as GLCM, HOG, Gabor, LBP) to the images in the dataset. This step aims to extract relevant features that can be used by the model for accurate classification.

Save Model: After the model is trained and ready, the admin saves it for future use. This involves storing the model parameters, weights, or any other necessary information to be able to load and utilize the model during the classification process.

User Use Cases:

Input Image: The user interacts with the system by providing an image to be classified. This can be done through an upload feature, where the user selects an image file from their device.

Classify Image: Once the user uploads the image, the system utilizes the trained model to classify the image and determine whether hemorrhage is present or not. This involves applying the feature extraction algorithms to the uploaded image and passing the extracted features to the model for prediction.

Obtain Result: After the classification process, the system provides the user with the result of the image classification. The result indicates whether hemorrhage is detected or not. This information can be displayed on the result page or conveyed through a notification or message to the user.

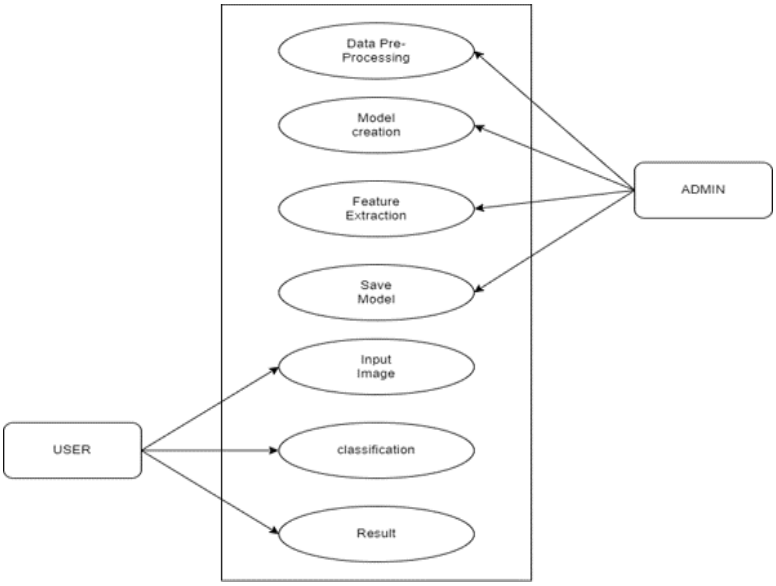

Figure 5.5: Use case diagram for customer

Chapter 6

Implementation

6.0.1 Dataset Collection:

Collect a dataset of intracranial images containing hemorrhagic and non-hemorrhagic cases. This dataset should be labeled to indicate the presence or absence of hemorrhage.

6.0.2 Preprocessing:

Load the intracranial images from the dataset. Apply necessary preprocessing techniques such as resizing, normalization, and noise removal to ensure consistency and enhance the quality of the images.

6.0.3 Feature Extraction:

Utilize multiple feature extraction techniques, including GLCM (Gray-Level Co-occurrence Matrix), HOG (Histogram of Oriented Gradients), LBP (Local Binary Patterns), and Gabor filters. For GLCM feature extraction, compute statistical measures such as dissimilarity and correlation from the co-occurrence matrix. For HOG feature extraction, calculate the gradients and generate histograms of oriented gradients. For LBP feature extraction, encode the local texture patterns using binary patterns. For Gabor feature extraction, apply Gabor filters at different orientations and scales to capture texture information.

6.0.4 Feature Fusion:

Combine the extracted features from GLCM, HOG, LBP, and Gabor into a single feature vector. Use appropriate techniques like concatenation or averaging to merge the features.

6.0.5 Training:

Split the dataset into training and testing sets. Use the training set to train a Random Forest classifier. The Random Forest classifier is a popular machine learning algorithm suitable for classification tasks.

6.0.6 Model Evaluation:

Evaluate the trained model's performance using the testing set. Calculate metrics such as accuracy, precision, recall, and F1 score to assess the model's effectiveness in detecting intracranial hemorrhage.

6.0.7 Model Saving:

Save the trained Random Forest model to disk for future use. User Input and Image Processing:

Allow the user to input a intracranial image for testing the model's detection capability. Preprocess the user's input image using the same techniques applied to the dataset images, including resizing, normalization, and noise removal.

6.0.8 Feature Extraction on User Image:

Apply the same feature extraction techniques (GLCM, HOG, LBP, Gabor) used during training to extract features from the user's preprocessed image.

6.0.9 Feature Fusion on User Image:

Combine the extracted features from the user's image into a single feature vector.

6.0.10 Classification of User Image:

Use the trained Random Forest model to classify the user's image based on the extracted features. Obtain the predicted class, indicating whether hemorrhage is present or not.

6.0.11 Display the Result:

Present the classification result to the user, indicating whether the uploaded image depicts intracranial hemorrhage or not.

6.0.12 Feature Extraction Algorithm

```
def extract_features(image):
    # HOG feature extraction
    fd_hog, hog_image = hog(image, orientations=8, pixels_per_cell=(16, 16), cells_per_block=(1, 1), visualize=True)

    # GLCM feature extraction
    glcm = greycomatrix(image, [5], [0], 256, symmetric=True, normed=True)
    fd_glcm = np.hstack([greycoprops(glcm, 'dissimilarity')[0], greycoprops(glcm, 'correlation')[0]])

    # LBP feature extraction
    fd_lbp = local_binary_pattern(image, 8 * 3, 3)

    # Gabor filter feature extraction
    kernels = []
    for theta in range(4):
        theta = theta / 4. * np.pi
        for sigma in (1, 3):
            kernel = np.real(gabor_kernel(0.5 / sigma, theta=theta))
            kernels.append(kernel)
    feats = []
    for kernel in kernels:
        filtered = cv2.filter2D(image, cv2.CV_8UC3, kernel)
        stats = filtered.flatten()
        feats.extend(stats)
    fd_gabor = np.asarray(feats)

    return np.hstack([fd_hog, fd_glcm, fd_lbp.ravel(), fd_gabor])
```

Figure 6.1: Feature Extraction Algorithm
GLCM (Gray-Level Co-occurrence Matrix), HOG (Histogram of Oriented Gradients), Gabor filters, and LBP (Local Binary Patterns) to extract relevant features from the images .

6.0.13 Classification

```
def main():
    # Load images from folders and extract features from them.
    images_hemorrhage = load_images_from_folder(r"C:\Users\Yajnesh\OneDrive\Desktop\newdata\php")
    images_no_hemorrhage = load_images_from_folder(r"C:\Users\Yajnesh\OneDrive\Desktop\newdata\pnp")

    features_hemorrhage = []
    for image in images_hemorrhage:
        features_hemorrhage.append(extract_features(image))

    features_no_hemorrhage = []
    for image in images_no_hemorrhage:
        features_no_hemorrhage.append(extract_features(image))

    # Create labels for the data.
    labels_hemorrhage = np.ones(len(features_hemorrhage))
    labels_no_hemorrhage = np.zeros(len(features_no_hemorrhage))

    # Combine data and shuffle it.
    X = np.vstack([features_hemorrhage, features_no_hemorrhage])
    y = np.hstack([labels_hemorrhage, labels_no_hemorrhage])

    X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3)

    # Train Random Forest classifier.
    rf_clf = RandomForestClassifier(max_depth=None, min_samples_leaf=2, min_samples_split=5, n_estimators=150).fit(X_train,y_train)


    #predict using rf
    y_pred_rf = rf_clf.predict(X_test)


    print("Accuracy of Random Forest classifier:", accuracy_score(y_test,y_pred_rf))

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

Figure 6.2: Classification Algorithm
Random Forest Algorithm to classify the images.

Chapter 7

System Testing

Testing is a process of executing a program with the explicit intention of finding error. It is a process used to identify correctness, completeness and quality of developed computer software. There are many approaches to software testing, but effective testing of complex product is essential a process of investigating. Testing helps in verifying and validating if he software working as it is intended to work. This involves using static and dynamic methodologies to test the application. There are two methods for test case design.

White Box testing:

White box testing strategy deals with the internal logic and structure of the code. It is also called as glass, structural, open and clear box testing. The test that are written based on the white box testing strategy incorporate coverage of the code written, branches, statements and internal logic of the code etc. In order to implement white box testing the tester has to deal with the code and hence it is required possess knowledge of the coding and logic i.e. Internal working of the code.

Advantages:

• As the knowledge of the internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively

• It helps in optimizing the code

• It helps in removing the extra line of code, which introduce defect in the code

Disadvantage

• As the knowledge of code and internal structure is a prerequisite, a skilled tester is needed to carry out this type of testing, and this, in turn, increase the cost of the software

• It is nearly impossible into every bit of code to find out the hidden errors, which may create problems, resulting in failure of application

Black Box testing Black box testing takes the internal perspective of the test object

to derived test cases. These tests can be functional or non-functional though usually functional. The test designer selects valid and invalid inputs and determines the correct input. There is no knowledge of the test object's internal structure. This method of test design is applicable to all levels of software testing: unit, internal, functional and system and acceptance

Advantages:

- Black box test are reproducible
- The environment in which the program is running is also tested
- The invested effort can be used multiple times

Disadvantages:

- The results are often over estimated
- Not all properties of the software can be tested
- The reason for failure is not found

Testing Methodologies

- Unit Testing
- Integration Testing
- System Testing

Unit Testing

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage producers, are tested to determine ifthey are fit to use. Intuitively, one can view a unit as the smallest testable part of an application, In procedural programming a unit could be an entire module but is more commonly an individual function or procedure. In object oriented programming a unit is often an entire interface, such as class, but could be an individual method. Unit tests are created by programmers or occasionally by white box testers during the development process.

Integration Testing

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These design items, i.e. assemblages (or group of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test that all components with in assemblages interact correctively, for example across producers call of

procedures activation, and this is done after testing individual modules, i.e. unit testing.

System Testing

A system testing of software or hardware is testing conducted on a complete, integrated system to evaluate system's compliance with its specified requirements. System testing falls within the scope of black box testing, and such as, should require no knowledge of the inner design of the integrated software components that have successfully passed integration testing and also the software components itself integrated with any applicable hardware systems. The purpose of integration testing is to detect any inconsistencies between software units that are integrated together(called assemblages) or between any of the assemblages and the hardware, System is more limited type of testing, it seeks to detect defects both within the inter-assemblages and also within the system as whole.

Table 7.1: Work Flow

Sl No	Work	Duration(in Weeks)
1	Dataset Collection	2
2	Pre-processing Images	1
3	Extracting Features from set of Images	1
4	Classification based on the set of extracted features	1
5	Optimizing parameters	1
6	Testing	1

Table 7.2: Registration Test cases

TC#	Description	Expected Result	Actual Result	Status
TC-1	When user click on the Create new account	Displays registration form	Displays registration form	Pass
TC-2	user Enters Name, Mobile No, email, Address, Gender, DOB,and clicks on signup.	Successfully registered	Successfully registered	Pass
TC-3	If any fields are blank	Please enter Mobile no	Please enter Mobile no	Pass

Table 7.3: Login Test cases

TC#	Description	Expected Result	Actual Result	Status
TC-1	If user clicks on login button without entering username and password.	Please enter username and password	Please enter username and password	Pass
TC-2	Enter username and password and click enter	Login success	Login success	Pass
TC-3	If username is blank but password is entered.	Please enter username	Please enter username	Pass
TC-3	If password is blank but username is entered.	Please enter password	Please enter password	Pass
TC-3	If the username or password is incorrect.	Invalid login id or Password	Invalid login id or Password	Pass

Chapter 8

Results and Disscussion

8.1 Landing Page

The Landing page serves as the entry point for users who want to access a specific system or application. It typically has user buttons which navigates to a page which includes registration and login buttons that provide different options for users based on their registration status.

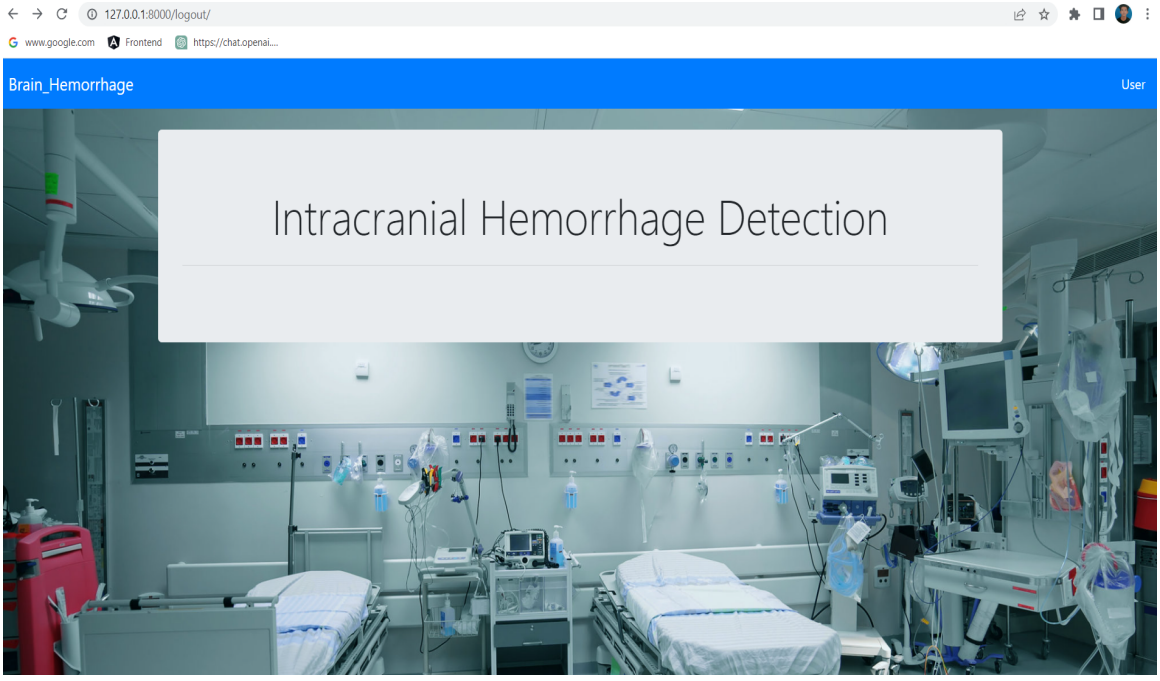

Figure 8.1: User Landing Page

8.2 Registration Page

The registration button allows new users who have not yet registered to navigate to the registration page. When clicked, it redirects the user to a separate page where they can provide their necessary information to create a new account. This page usually includes fields for entering personal details, such as name, email address, password, and any other required information.

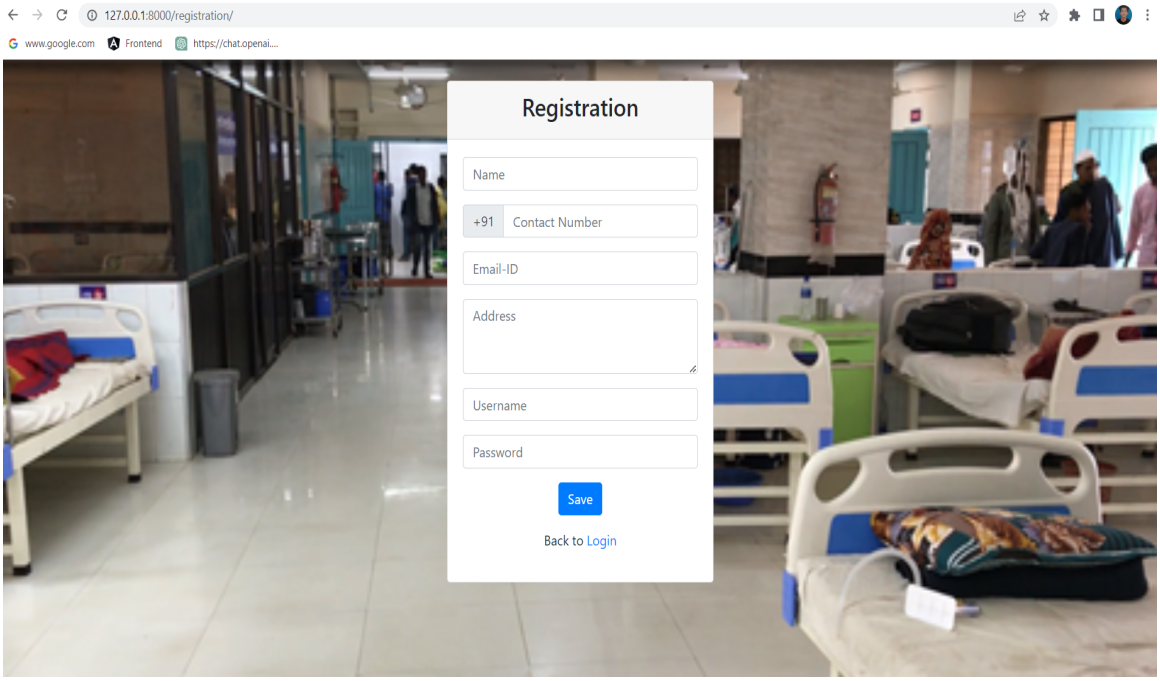

Figure 8.2: Registration Page

8.3 Login Page

The login button is designed for users who have already registered and want to access their existing account. When clicked, it redirects the user to the login page, where they can enter their credentials, such as username/email and password, to authenticate themselves and gain access to the system. The login page typically includes a "Forgot Password" option in case users have trouble remembering their login credentials.

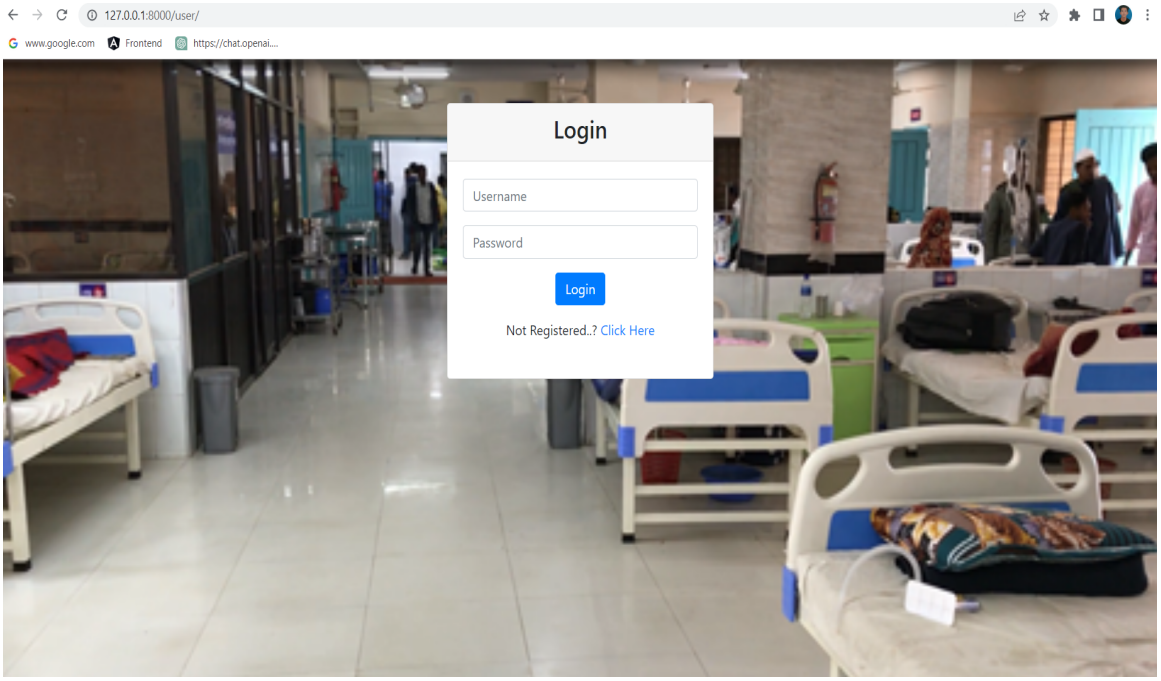

Figure 8.3: Login Page

8.4 User Dashboard

The user dashboard provides users with a convenient interface to access various functionalities and navigate within the system. It typically includes several buttons that allow users to perform different actions.

Home Button: The home button is designed to navigate users back to the home page of the system or application. When clicked, it redirects the user to the main landing page or dashboard

Upload Image Button: The upload image button allows users to navigate to the image testing page or module. By clicking this button, users can upload an image file from their device or a designated location within the system for analysis.

Logout Button: The logout button is used to log out the currently logged-in user from the system. When clicked, it terminates the user's session and redirects them to back to login page

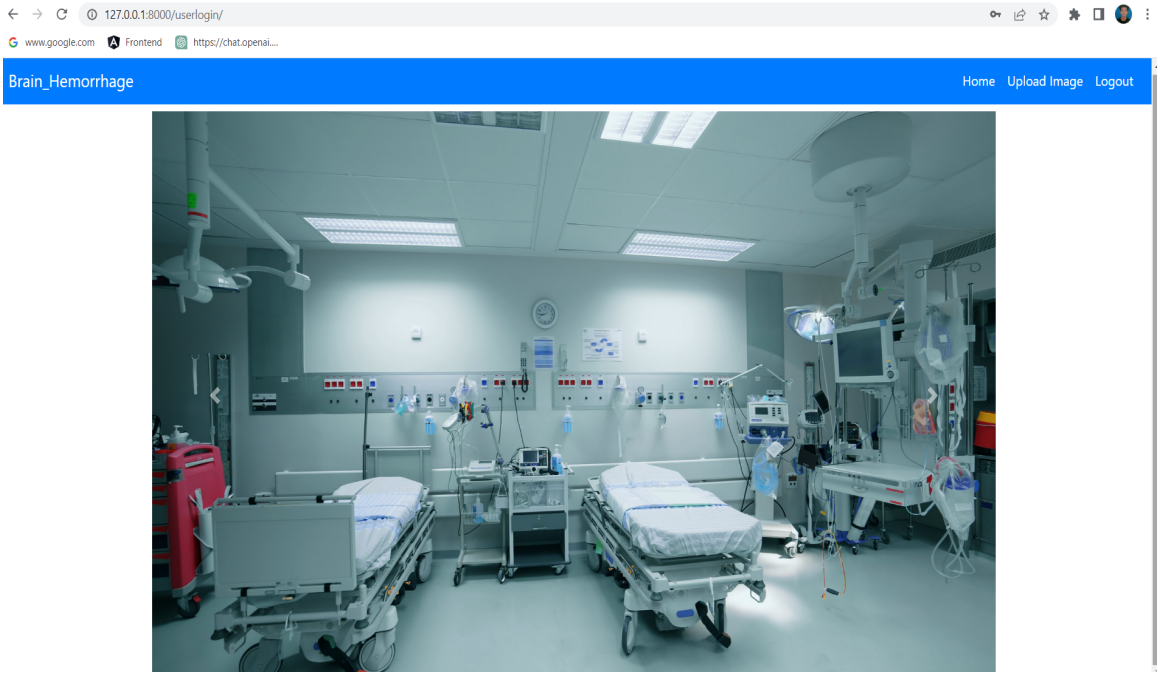

Figure 8.4: User Dashboard

8.5 Test Image Page

The image Testing page provides users with the capability to upload images and perform the hemorrhage detection process. Here's an elaboration on the functionality of the image testing page:

Image Upload: On the image testing page, users are presented with an option to upload their medical images for analysis. This feature allows users to select and upload an image file from their device or a designated location within the system. The supported image formats and any size restrictions should be communicated to the user to ensure proper image upload. Once the user uploads an image, the system performs the hemorrhage detection process on the uploaded image. The system utilizes the feature extraction algorithms such as GLCM (Gray-Level Co-occurrence Matrix), HOG (Histogram of Oriented Gradients), Gabor filters, and LBP (Local Binary Patterns) to extract relevant features from the image. These features are then fed into a classification algorithm, such as Random Forest, to determine whether hemorrhage is present or not.

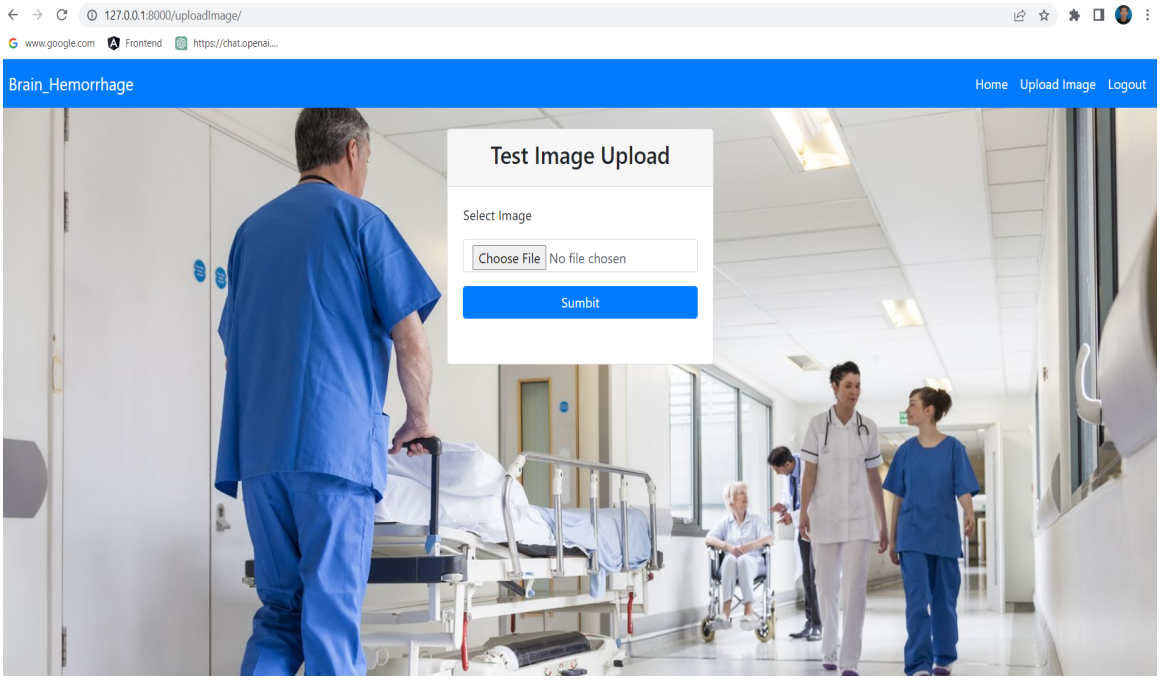

Figure 8.5: Image testing Page

8.6 Result Page

The result page provides users with the outcome of the hemorrhage detection process performed on the uploaded image. Here's an elaboration on the functionality of the result page:

Result Display: Upon completion of the hemorrhage detection process, the result page displays the outcome of the analysis for the uploaded image. The page presents a clear indication of whether hemorrhage is present or not in the image.

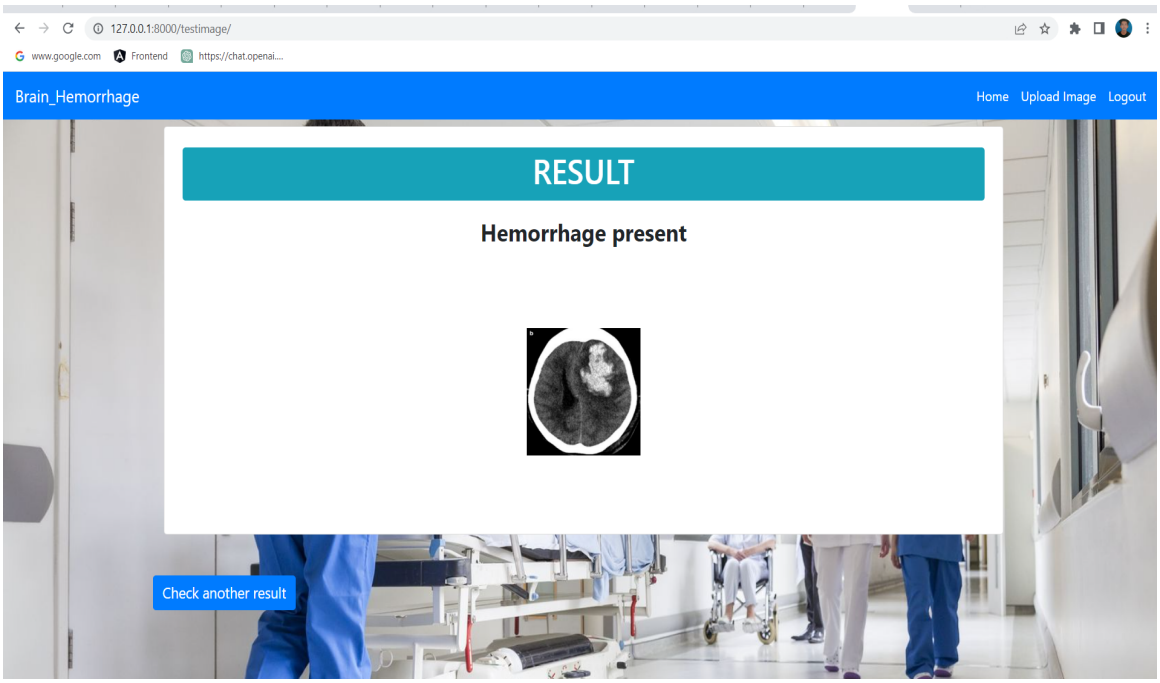

Figure 8.6: Result Page

Chapter 9

Conclusion and Future work

The application of image processing techniques, such as GLCM, HOG, Gabor filters, and LBP, combined with the Random Forest classification algorithm, holds great promise for the detection of intracranial hemorrhage. By extracting relevant features from medical images, these techniques enable the accurate and automated identification of hemorrhage regions, aiding healthcare professionals in their diagnosis and treatment decisions.

The future of intracranial hemorrhage detection using image processing techniques looks promising. Advancements in computer vision and machine learning algorithms are expected to further enhance the accuracy and efficiency of detection systems. Here are some potential future directions:

Improved Feature Extraction: Research and development efforts can focus on refining existing feature extraction algorithms or exploring new techniques that capture more specific and discriminative characteristics of intracranial hemorrhage. This could involve incorporating additional texture descriptors, shape analysis methods, or advanced feature fusion approaches.

Integration of Deep Learning: Deep learning techniques, such as convolutional neural networks (CNNs), have shown remarkable success in various medical imaging tasks. Integrating deep learning models into the detection system could potentially improve the accuracy and robustness of intracranial hemorrhage detection by learning hierarchical representations directly from the raw image data.

Expansion of Dataset: The availability of large and diverse datasets plays a crucial role in training accurate and generalizable detection models. Efforts can be made to expand existing datasets or create new ones with a wide range of hemorrhage cases, including different sizes, locations, and types. This would help to enhance the performance and generalizability of the detection system.

Real-time and Point-of-Care Applications: Streamlining the detection system for real-time and point-of-care applications can significantly impact patient outcomes. Optimizing the algorithms for speed and resource efficiency, and developing user-friendly interfaces for healthcare professionals, would enable prompt and efficient detection of intracranial hemorrhage during critical medical interventions.

Integration with Clinical Decision Support Systems: Integrating the detection system with clinical decision support systems can provide valuable insights and assistance to healthcare professionals. By combining the automated detection results with patient information and medical records, the system can offer personalized recommendations and aid in treatment planning.

References

[1] Ruan, X., Shi, T., Chen, H., Yao, H. (2020, September). Feature Extraction and Research Based on MRI Image of Cerebral Hemorrhage. In 2020 International Conference on Computer Network, Electronic and Automation (ICCNEA) (pp. 369-372). IEEE.

[2] Mushtaq, M. F., Shahroz, M., Aseere, A. M., Shah, H., Majeed, R., Shehzad, D., Samad, A. (2021). BHCNet: neural network-based intracranial hemorrhage classification using head CT scan. IEEE Access, 9, 113901-113916.

[3] Cao, G., Wang, Y., Zhu, X., Li, M., Wang, X., Chen, Y. (2020, December). Segmentation of intracerebral hemorrhage based on improved U-Net. In 2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS) (pp. 183-185). IEEE.

[4] Singh, S. P., Wang, L., Gupta, S., Gulyas, B., Padmanabhan, P. (2020). Shallow 3D CNN for detecting acute intracranial hemorrhage from medical imaging sensors. IEEE Sensors Journal, 21(13), 14290-14299.

[5] Rane, H., Warhade, K. (2021, March). A survey on deep learning for intracranial hemorrhage detection. In 2021 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 38-42). IEEE.

[6] Majeed, M. A. A., Al Okashi, O. M., Alrawi, A. T. (2023, January). Intracranial hemorrhage detection and classification from CT images based on multiple features and machine learning approaches. In 2023 15th International Conference on Developments in eSystems Engineering (DeSE) (pp. 498-503). IEEE.

[7] Yu, W., Kang, H., Sun, G., Liang, S., Li, J. (2022). Bio-Inspired Feature Selection in intracranial Disease Detection via an Improved Sparrow Search Algorithm. IEEE Transactions on Instrumentation and Measurement, 72, 1-15.