**A**

**Major Project Report**

**on**

# FUZZY ENHANCED KIDNEY TUMOUR DETECTION: INTEGRATING MACHINE LEARING OPERATIONS FOR A FUSION OF TWIN TRANFERABLE NETWORK AND WEIGHTED ENSEMBLING ML CLASSIFIER

*Submitted to CMREC (UGC Autonomous)*

*In Partial Fulfillment of the requirements for the Award of Degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**(Artificial Intelligence and Machine Learning)**

Submitted By


**M SRUJAN - 228R1A6638**

**S POOJA - 228R1A6653**

**T MADHULIKA - 228R1A6659**

**B ARUN - 238R5A6603**

Under the Esteemed guidance of

**Mr. G. Venkateswarlu**

Assistant Professor, Department of CSE (AI & ML)

**Department of Computer Science & Engineering (AI & ML)**

**CMR ENGINEERING COLLEGE**

**(UGC AUTONOMOUS)**

(Accredited by NAAC & NBA, Approved by AICTE, NEW DELHI, Affiliated to JNTU Hyderabad)

(Kandlakoya, Medchal Road, Medchal-Malkajgiri Dist,. Hyderabad-501 401)


**(2025-2026)**

**CMR ENGINEERING COLLEGE**

**UGC AUTONOMOUS**

*(Accredited by NAAC & NBA, Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad, Kandlakoya, Medchal Road, Hyderabad-501 401)*

**Department of Computer Science & Engineering**

**(Artificial Intelligence and Machine Learning)**

## CERTIFICATE

This is to certify that the Major Project entitled **"FUZZY ENHANCED KIDNEY TUMOUR DETECTION: INTEGRATING MACHINE LEARING OPERATIONS FOR A FUSION OF TWIN TRANFERABLE NETWORK AND WEIGHTED ENSEMBLING ML CLASSIFIER"**is a bonafide work carried out by

| | | |
|---|---|---|
| **M SRUJAN** | - | **228R1A6638** |
| **S POOJA** | - | **228R1A6653** |
| **T MADHULIKA** | - | **228R1A6659** |
| **B ARUN** | - | **238R5A6603** |

in partial fulfillment of the requirement for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING(AI&ML)from CMR Engineering College, under our guidance and supervision.

The results presented in this major project have been verified and are found to be satisfactory.The results embodied in this major project have not been submitted to any other university for the award of any other degree or diploma.

| **Internal Guide** | **Major Project Cordinator** | **Head of the Department** |
|---|---|---|
| Mr. G. Venkateswarlu | Mr. G. Venkateswarlu | Dr. Madhavi Pingili |
| Assistant Professor | Assistant Professor | Professor & HOD |
| CSE-(AI&ML) | CSE (AI & ML) | CSE (AI & ML) |

**External Examiner:**________________________

# DECLARATION

This is to certify that the work reported in the present Major project entitled **"FUZZY ENHANCED KIDNEY TUMOUR DETECTION:INTEGRATING MACHINE LEARING OPERATIONS FOR A FUSION OF TWIN TRANFERABLE NETWORK AND WEIGHTED ENSEMBLING ML CLASSIFIER"** is a record of bonafide work done by us in the Department of Computer Science and Engineering (AI&ML), CMR Engineering College. The reports are based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this Major project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.


**M SRUJAN - 228R1A6638**

**S POOJA - 228R1A6653**

**T MADHULIKA - 228R1A6659**

**B ARUN - 238R5A6603**

# ACKNOWLEDGMENT

We are extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr. Madhavi Pingili**, Professor & HOD, **Department of CSE (AI & ML), CMR Engineering College** for their constant support.

We are extremely thankful to **Mr. G. Venkateswarlu,** Assistant Professor, Internal Guide, Department of CSE (AI & ML), for his constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We thank **Mr. G. Venkateswarlu,** Major Project Coordinator for his constant support in carrying out the project activities and reviews.

We express our thanks to all staff members and friends for all the help and co-ordination extended in bringing out this project successfully in time.

Finally, We are very much thankful to our parents who guided us in every step.

**M SRUJAN - 228R1A6638**

**S POOJA - 228R1A6653**

**T MADHULIKA - 228R1A6659**

**B ARUN - 238R5A6603**

# TABLE OF CONTENTS

# ABSTRACT

Accurate Kidney tumor detection using Computed Tomography (CT) imaging plays a crucial role in early diagnosis and treatment planning. The existing system employs fuzzy-based enhancement with pre-trained deep convolutional neural networks (ResNet101 and DenseNet121) for feature extraction, followed by a weighted-average ensemble of classical classifiers for tumor classification. Although this approach achieves promising performance, it suffers from limitations such as manual tuning of fuzzy membership functions, reliance on relatively older backbone networks, and less effective ensemble weighting strategies, which reduce generalization and robustness under noisy conditions. To address these challenges, this work proposes an improved kidney tumor detection framework that retains the same pipeline but incorporates more accurate algorithms. The fuzzy enhancement stage is optimized through automated parameter tuning to adaptively enhance CT images. For feature extraction, ResNet101 and DenseNet121 are replaced with EfficientNetV2-B0 and ResNeSt-50, enabling the extraction of more discriminative and noise-resilient features. Feature fusion is refined using a projection layer with batch normalization and dropout to reduce overfitting. Finally, the classification stage leverages a stacked ensemble of LightGBM, XGBoost, and CatBoost, with a logistic regression meta-classifier ensuring robust predictions and probability calibration. Experimental evaluation across multiple datasets (including noisy test conditions) demonstrates that the proposed system achieves higher accuracy, precision, recall, and F1-score compared to the existing model. Furthermore, it exhibits improved robustness, better calibrated outputs, and reduced overfitting,making it more suitable.

**Keyword**: Kidney Tumor Detection, Fuzzy Enhancement, Pre-trained Deep Convolutional Neural Networks, EfficientNetV2-B0, ResNeSt-50, Feature Fusion, Gradient Boosted Ensemble,LightGBM, XGBoost, CatBoost, Medical Image Classification.

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER-1

# INTRODUCTION

# 1. INTRODUCTION

## 1.1 Introduction

In The kidneys play a vital role in eliminating waste products and toxins from the bloodstream **[1][6]**.Tumors, or cancers, arise due to abnormal cell growth, impacting individuals in different ways and producing varied symptoms. Therefore, early detection of kidney tumors is essential to lower the risk of disease progression and improve patient survival **[2]**. Nearly one-third of kidney tumor cases are diagnosed only after the cancer has spread to other parts of the body, while many remain asymptomatic and are often identified incidentally during examinations for unrelated health conditions.

On radiographic scans, kidney tumors can sometimes be misinterpreted as cysts, abdominal masses, or sources of abdominal pain, even though the symptoms may not originate from the kidneys themselves **[3][5]**.Some of the mild signs or complications linked to kidney tumors include fatigue, low hemoglobin levels, abdominal pain, vomiting, hyperglycemia, and hematuria. Approximately 30% of patients develop anemia as a result of these tumors **[2][4]**. In most cases, solid masses or growths inside the kidneys tend to be malignant. Detecting such tumors early is crucial for selecting the most suitable treatment approach, as recovery outcomes largely depend on timely diagnosis.

One of the standard diagnostic tools used to identify kidney tumors is a computed tomography (CT) scan of the abdomen and pelvis. Through careful analysis of CT image characteristics, medical professionals can confirm the presence of tumors. Given their life-threatening nature, kidney tumors continue to drive the development of advanced methods for better diagnosis and treatment **[5][7]**.Kidney tumors present in different forms, and CT images may not always provide sufficient visible characteristics for accurate identification.

The manual interpretation of complex CT scans is not only time-consuming but also susceptible to human error, which can potentially lead to fatal consequences.

thereby improving accuracy. By enabling advanced data processing and enhancing reliability, deep learning contributes substantially to medical research **[6]**. Its growing adoption in healthcare is attributed to its capability of efficiently analyzing large-scale datasets, including clinical outcomes and research data, with high precision. Since the early detection of kidney tumors is essential for effective treatment, both deep learning and machine learning approaches play a pivotal role in accurately identifying tumors, assisting medical professionals in making timely treatment decisions.

An ensemble of ML classifiers, optimized through a weighted average strategy, delivers superior accuracy compared to individual classifiers. The rationale behind employing ensemble techniques in both DL-based feature extraction and ML-based classification is rooted in their ability to enhance predictive performance**[7]**. Moreover, ensemble methods foster model diversity, which improves classification efficiency and robustness, particularly when inter-model variance is high **[8]**.

## 1.2 Project Objectives

The main objective of this project is to develop an efficient and accurate system for detecting kidney tumors from CT images using advanced machine learning and deep learning techniques. The system aims to enhance image quality through optimized fuzzy enhancement methods, enabling better visualization of tumor regions. It also focuses on extracting highly discriminative and noise-resilient features by utilizing modern deep convolutional neural networks such as EfficientNetV2-B0 and ResNeSt-50. Furthermore, the project intends to improve classification performance by employing a robust ensemble learning approach, combining multiple machine learning models to achieve higher accuracy, precision, and reliability.

- To improve CT image quality using optimized fuzzy enhancement, ensuring better visibility of tumor regions.

- To extract discriminative and noise-resilient features by replacing older CNN backbones (ResNet101, DenseNet121) with EfficientNetV2-B0 and ResNeSt-50.

## 1.3 Purpose of the Project

The purpose of this project is to design and develop an efficient and reliable computer-aided diagnostic system for detecting kidney tumors from CT images. The project aims to overcome the limitations of existing methods, such as manual tuning of fuzzy parameters, less effective feature extraction techniques, and traditional ensemble classifiers that may not provide optimal accuracy. By incorporating optimized fuzzy enhancement, advanced deep learning models, and modern ensemble learning techniques, the system seeks to improve detection accuracy and robustness.

The ultimate goal is to assist radiologists and medical professionals in early and accurate diagnosis, reduce the chances of human error, and support better clinical decision-making, thereby enhancing patient care and treatment outcomes.Another important purpose of the project is to overcome the limitations of existing systems. Traditional approaches rely on manually tuned fuzzy parameters, older deep learning architectures, and simple ensemble techniques, which may lead to reduced accuracy and poor generalization when dealing with noisy or diverse datasets.

## 1.4 Problem Statement

The early detection of kidney tumors is a critical challenge in medical imaging, as delayed diagnosis can lead to severe health complications and reduced survival rates. Computed Tomography (CT) scans are widely used for identifying kidney abnormalities; however, accurately distinguishing tumor regions from normal tissues remains difficult due to variations in image quality, noise, and similarity with other structures. Traditional diagnostic methods rely heavily on manual interpretation by radiologists, which can be time-consuming, error-prone, and inconsistent, especially when dealing with large volumes of medical data.

Existing automated systems for kidney tumor detection utilize fuzzy enhancement techniques combined with deep learning models such as ResNet101 and DenseNet121 for feature extraction, followed by ensemble machine learning classifiers.

This challenge becomes more critical in large-scale scenarios with diverse mobility patterns such as pedestrian and vehicular movement, where traffic behavior Furthermore, the direct fusion of high-dimensional features without proper optimization increases the risk of overfitting, particularly when working with limited or imbalanced datasets. The lack of well-calibrated prediction probabilities also affects the reliability of these systems in real-world clinical environments. As a result, there is a need for a more advanced and adaptive framework that can improve feature representation, enhance classification performance, and ensure robustness under varying imaging conditions.

## 1.5 Existing Systems and their Limitations

The existing approaches for kidney tumor detection include fuzzy-based image enhancement techniques, traditional machine learning models, deep learning architectures, and ensemble-based classification methods. Although these methods have improved detection accuracy, they still face several limitations when applied to real-world medical imaging scenarios involving noisy and complex CT scan data.

**a. Statistical Model Limitations:**

Traditional image processing and statistical techniques are used for tumor detection; however, they fail to capture complex patterns and subtle variations in medical images. These methods rely on predefined rules and assumptions, which are not suitable for diverse and non-linear tumor characteristics.

**b. Machine Learning Limitations:**

Machine learning approaches such as SVM, Random Forest, and KNN improve classification performance but require manual feature extraction and selection. This process is time-consuming and highly dependent on domain expertise. Additionally, these models struggle to generalize well across different datasets and are less effective in handling noisy medical images.

**c. Deep Learning Limitations:**

Deep learning models like ResNet101 and DenseNet121 are widely used for feature extraction; however, they are relatively older architectures and may not achieve optimal performance compared to modern networks.

**d. Ensemble Model Limitations:**

Existing systems use weighted-average ensemble methods combining multiple classifiers. However, these methods depend on manually assigned weights, which may not optimally utilize the strengths of individual models. This reduces the overall effectiveness and adaptability of the system in different conditions.

**e. Fuzzy Enhancement Limitations:**

Fuzzy-based image enhancement improves contrast in CT images, but it relies on manually defined membership functions. These parameters may not adapt well to variations in image quality and noise levels across datasets. This limits the system's ability to consistently enhance tumor regions and affects overall detection accuracy.

## 1.6 Proposed System

The proposed system aims to enhance kidney tumor detection by improving each stage of the existing pipeline using advanced machine learning and deep learning techniques. The system follows a structured approach that includes fuzzy-based image enhancement, feature extraction using modern deep convolutional neural networks, feature fusion, and ensemble-based classification. Unlike traditional methods, the proposed system introduces optimized algorithms to achieve higher accuracy, robustness, and reliability.

Initially, the input CT images undergo fuzzy-based enhancement, where the membership function parameters are automatically optimized instead of being manually tuned. This step improves image contrast and highlights tumor regions more effectively, making it easier for the model to identify important features. In the feature extraction stage, older architectures such as ResNet101 and DenseNet121 are replaced with more advanced models like EfficientNetV2-B0 and ResNeSt-50. These modern networks are capable of extracting more discriminative and noise-resistant features, thereby improving the overall performance of the system.

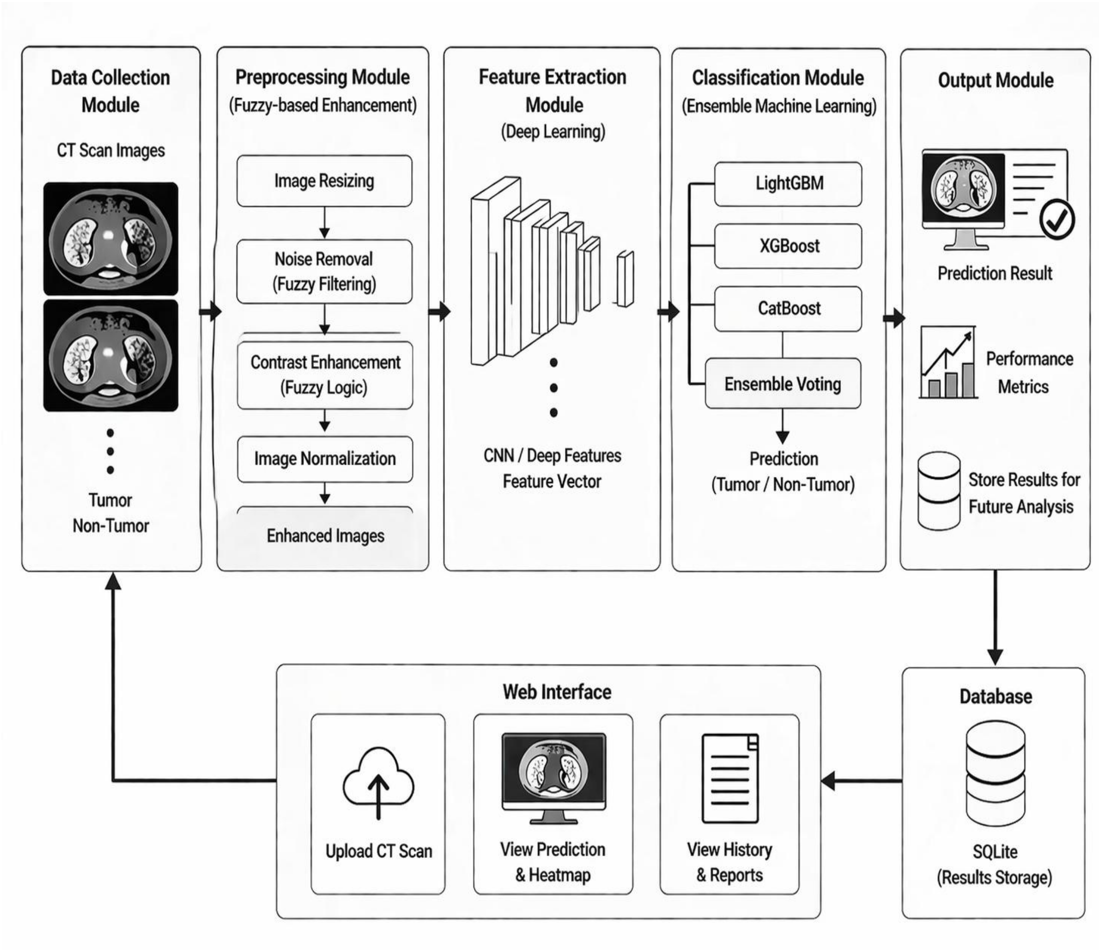


Fig 1.6 Block Diagram of the Proposed System

**Advantages of the Proposed System**

- Captures complex spatial dependencies between LTE cells
- High accuracy in tumor detection
- Improved image quality using fuzzy enhancement
- Advanced CNN models for better feature extraction
- Robust ensemble classification approach
- Reduced overfitting and better generalization
- Effective performance on noisy data
- Faster and efficient processing

## 1.7 Input and Output Design

### a. Input Design

Input design is a critical component of the kidney tumor detection system, focusing on how medical data such as CT scan images is collected, entered, and prepared for processing. It acts as a bridge between the user (doctor or technician) and the system, ensuring that all required data is captured accurately and efficiently. A well-designed input system reduces errors, improves data quality, and enhances overall system performance.

The input design process involves identifying the type of data required, such as medical images, patient details, and diagnostic parameters. The system provides user-friendly interfaces for uploading CT scan images and entering relevant information. It also includes preprocessing steps like normalization, resizing, and noise reduction to ensure that the data is in a suitable format for analysis. Validation techniques are applied to check image formats, file sizes, and completeness of input data.

**Key Considerations:**

- Identification of required input data (CT images, patient details)
- Proper arrangement and preprocessing of image data
- User-friendly interfaces for image upload and data entry

**Objectives:**

- Convert raw medical data into system-compatible formats
- Design intuitive and efficient input interfaces for users
- Ensure data accuracy through validation and preprocessing

### b. Output Design

Output Output design focuses on how the processed results of the kidney tumor detection system are presented to users in a clear and meaningful way. It plays a vital role in helping doctors and medical professionals interpret the results and make accurate clinical decisions. A good output design ensures that the information is precise, well-structured, and delivered at the right time.

The system generates outputs such as tumor detection results (tumor or non-tumor), highlighted tumor regions in images, and performance metrics. These outputs are displayed through user-friendly interfaces, including visual representations like annotated images or reports. The output format is designed to be clear and easy to understand, with important information highlighted for quick interpretation. The system may also provide confirmation messages and feedback to ensure transparency and usability. Outputs can be viewed on-screen, saved as reports, or stored for future reference, supporting efficient medical analysis and decision-making.

**Key Considerations:**

- Structured and organized presentation of results
- Selection of appropriate output formats (images, reports, dashboards)
- Clear and readable visualization of detection results

**Objectives:**

- Convert raw medical data into system-compatible formats
- Design intuitive and efficient input interfaces for users
- Ensure data accuracy through validation and preprocessing

# CHAPTER-2

# LITERATURE SURVEY

# 2 LITERATURE SURVEY

1. **Alzu'bi, et al., "Kidney Tumor Detection Using Convolutional Neural Network Architectures,"** ***Journal of Medical Imaging and Health Informatics*****, 2025.** The study compares three CNN models, namely a 6-layer CNN, ResNet50, and VGG16, using 8,400 CT images from 120 patients. It highlights that the 6-layer CNN achieved the highest accuracy of 97%, outperforming deeper architectures.
2. **M. Obaid, "A Framework for Kidney Tumor Detection Using Image Processing and Machine Learning Techniques,"** ***International Journal of Advanced Computer Science and Applications*****, 2024.** The study proposes a tumor detection framework where CT images are denoised using a median filter and segmented using K-means clustering. PCA is used for feature extraction, and classification is performed using a Probabilistic Neural Network, achieving 96.8% accuracy.
3. **Z. Zhou, et al., "Deep Learning-Based Classification of Kidney Tumors Using Transfer Learning,"** ***IEEE Access*****, 2024.** The study explores transfer learning using the InceptionV3 model on CT scans from 192 patients. It applies five-fold cross-validation and ROC analysis, achieving 97% accuracy.
4. **N. Schieda, et al., "Application of Deep Learning for Kidney Tumor Detection Using CT Imaging,"** ***European Radiology*****, 2024.** The study evaluates deep learning techniques on CT images from 177 patients and highlights improved diagnostic accuracy in tumor detection.
5. **D. Hain et al., "Anemia Management in Chronic Kidney Disease,"** ***Kidney Medicine*****, 2023.** The study provides insights into anemia management in CKD patients and highlights its importance in improving patient.
6. **H. Liu et al**., **"Kidney Tumor Detection Using Shape and Texture Features,"** ***Computerized Medical Imaging and Graphics*****, 2023**. The study analyzes shape and texture features using Random Forest and AdaBoost classifiers. It shows that combining features improves performance with AUC values ranging from 0.68 to 0.75
7. **Cleveland Clinic**, **"Kidney Tumor,"** ***Cleveland Clinic Health Library*****, 2022.** The study explains the types, causes, diagnosis, and treatment options for kidney tumors. It serves as a reliable online health resource providing fundamental medical knowledge

8. **S. Mahmoud et al., “Machine Learning for Kidney Cancer Diagnosis Using Imaging and Clinical Data,” *Cancers*, 2022.** The study focuses on integrating CT imaging with clinical metadata to improve classification accuracy and support surgical decision-making.
9. **M. Rana and M. Bhushan, “Artificial Intelligence in Medical Imaging: A Review,” *Multimedia Tools and Applications*, 2022.** The study reviews AI applications in medical imaging, emphasizing early and accurate diagnosis of diseases. It highlights the effectiveness of machine learning and deep learning methods in detection and diagnosis.
10. **A. Eskandari et al., “An Autonomous Fault Diagnosis Framework Using Ensemble Learning,” *International Journal of Electrical Power & Energy Systems*, 2022.** The study develops an autonomous fault diagnosis framework and shows that ensemble learning combined with genetic algorithms improves robustness and accuracy.
11. **A. Sarkar et al.**, **“Efficient Intrusion Detection Using Ensemble Learning Techniques,” *International Journal of Information Technology*, 2021**. The study achieves efficient intrusion detection with improved accuracy by applying ensemble learning along with hyperparameter optimization techniques.
12. **S. Litjens et al.**, **“A Survey on Deep Learning in Medical Image Analysis,” *Medical Image Analysis*, 2021.** The study provides a comprehensive overview of deep learning techniques in medical imaging, highlighting their effectiveness in disease detection and diagnosis. It emphasizes the role of CNNs in improving accuracy and automation in medical image analysis.

**Literature Review Summary**

The reviewed studies Table 2.0, collectively highlight the continuous evolution, focusing on improving accuracy, efficiency, and adaptability.

| **Title** | **Key Findings** | **Method / Approach** |
|---|---|---|
| URMC, "Overview of Kidney Disorders" | Provides fundamental medical knowledge on kidney disorders, symptoms, and treatment approaches. | Online medical encyclopedia |
| S. Mahmud et al., Cancers | Achieved effective classification and decision support for surgery planning using combined imaging and metadata. | ML on CT scans + clinical metadata |
| UrologyHealth.org, "Renal Mass and Localized Renal Tumors" | Offers clinical information about renal masses, diagnosis, and treatment options. | Online medical source |
| D. Hain et al., Kidney Med. | Provides insights on anemia management in chronic kidney disease (CKD) patients. | Narrative review |

| Title | Key Findings | Method / Approach |
|---|---|---|
| Cleveland Clinic, "Kidney Tumor" | Explains types, causes, diagnosis, and treatment options for kidney tumors. | Online health source |
| M. Rana & M. Bhushan, Multimedia Tools Appl. | Reviewed AI applications in medical imaging for early and accurate diagnosis of diseases. | ML & DL methods for diagnosis/detection |
| A. Eskandari et al., Int. J. Electr. Power Energy Syst. | Developed an autonomous fault diagnosis framework; ensemble improves robustness and accuracy. | Weighted ensemble learning + genetic algorithm |
| A. Sarkar et al., Int. J. Inf. Technol. | Achieved efficient intrusion detection with improved accuracy using optimized ML ensembles. | Ensemble learning + hyperparameter optimization |
| S. Yang et al., Pattern Recognit. | Demonstrated how diverse augmentation strategies significantly improve model performance. | Empirical analysis of similarity & diversity |
| UrologyHealth.org, "Renal Mass and Localized Renal Tumors" | Offers clinical information about renal masses, diagnosis, and treatment options. | Online medical source |

Table 2.0 Literature Review Summary

# CHAPTER-3

# SOFTWARE REQUIREMENT ANALYSIS

# 3 SOFTWARE REQUIREMENT ANALYSIS

Software Requirement Analysis (SRA) is a crucial phase in the development of the proposed kidney tumor detection system, as it defines the overall functionality, performance, and constraints of the system. This phase involves identifying and analyzing user needs, system requirements, and operational conditions to ensure the successful design and implementation of the application. It includes understanding the problem statement, defining functional requirements such as image input, preprocessing, feature extraction, and tumor classification, and specifying non-functional requirements like performance, accuracy, scalability, and reliability.

## 3.1 Modules and Their Functionalities

### 3.1.1 Data Preprocessing Module

Data preprocessing is a crucial step in the proposed kidney tumor detection system, as it prepares raw medical images for effective analysis and model training. In this stage, CT scan images are cleaned and standardized to ensure consistency across the dataset. Techniques such as resizing, normalization, and format conversion are applied to make the images suitable for processing by machine learning models.

### 3.1.2 Graph Construction Module

The Graph Construction Graph construction is an important step in systems that utilize graph-based approaches for data representation and analysis. In this stage, the input data is transformed into a graph structure where individual elements are represented as nodes and the relationships or similarities between them are represented as edges.

### 3.1.3 Model Training

The Model training is a critical phase in the proposed kidney tumor detection system, where the machine learning and deep learning models learn patterns from the preprocessed data. In this stage, the prepared CT images are fed into the models to enable them to identify distinguishing features between tumor and non-tumor regions.

### 3.1.4 Visualization Module

Visualization plays an important role in the proposed kidney tumor detection system by presenting the results in a clear and interpretable manner. It involves displaying medical images, highlighted tumor regions.

### 3.1.5 Model Evaluation & Testing

Model evaluation and testing are essential steps in the proposed kidney tumor detection system to assess the performance and reliability of the trained models. In this phase, the model is tested using unseen data to evaluate how well it generalizes beyond the training.

### 3.1.6 Prediction & Deployment

Prediction and deployment represent the final stage of the proposed kidney tumor detection system, where the trained model is used to make real-time predictions and is integrated into a practical environment.

### 3.1.7 Storage and Serialization Module

This module handles loading of the trained model and preprocessing tools using the pickle library. It enables efficient reuse of trained components without retraining, improving system performance.

## 3.2. Functional Requirements

### a. Data Preprocessing

Performance is a critical non-functional requirement of the proposed kidney tumor detection system, as it determines how efficiently and quickly the system processes medical images and delivers results. The system is expected to provide fast and accurate predictions with minimal latency, ensuring timely diagnosis in clinical settings.

### b. Scalability

Scalability is an important non-functional requirement of the proposed kidney tumor detection system, as it determines the system's ability to handle increasing amounts of data and users efficiently. The system should be capable of processing large volumes of medical images without degradation in performance.

**c. Reliability**

Reliability is a crucial non-functional requirement of the proposed kidney tumor detection system, as it ensures consistent and dependable performance under various conditions. The system should provide accurate and stable predictions for the same input data, minimizing errors and variations in result.

**d. Usability**

After training Usability is an important non-functional requirement of the proposed kidney tumor detection system, as it ensures that the system is easy to use and accessible for healthcare professionals improvements.

**e. Visualization**

Visualization plays a key role in understanding model performance and data structure. Training curves, accuracy graphs, and loss curves are plotted to analyze the learning process of the model. Furthermore, graph structures and LTE cell tower connections are visualized to better understand spatial relationships within the network.

**f. Prediction & Deployment**

The system is designed to run across multiple platforms, including Windows, Linux, and cloud environments such as Google Colab and AWS. This ensures flexibility and ease of deployment in different computing environments.

## 3.3. Non-Functional Requirements

**Performance**

The system is designed to support fast execution of graph operations to ensure efficient processing. It also leverages GPU acceleration to speed up model training and handle computationally intensive tasks effectively.

**Scalability**

Scalability is an important non-functional requirement of the proposed kidney tumor detection system, as it determines the system's ability to handle increasing amounts of data and users efficiently.

**Reliability**

Reliability is a crucial non-functional requirement of the proposed kidney tumor detection system, as it ensures consistent and dependable performance under various conditions. The system should provide accurate and stable predictions for the same input data, minimizing errors and variations in results.

**Usability**

Usability is an important non-functional requirement of the proposed kidney tumor detection system, as it ensures that the system is easy to use and accessible for healthcare professionals interaction.

**Portability**

The system is designed to run across multiple platforms, including Windows, Linux, and cloud environments such as Google Colab and AWS. This ensures flexibility and ease of deployment in different computing environments.

## 3.4. Feasibility Study

Feasibility study is an important phase in the development of the proposed kidney tumor detection system, as it evaluates whether the system can be successfully designed, implemented, and deployed in real-world conditions. It involves analyzing different aspects such as technical, economic, and operational feasibility to determine the practicality of the project. Technical feasibility ensures that the required tools, technologies, and resources are available to develop the system effectively. Economic feasibility evaluates the cost involved in development and maintenance, ensuring that the system is affordable and cost-effective.

### 3.4.1 Economic Feasibility

Economic feasibility evaluates whether the proposed kidney tumor detection system can be developed and maintained within reasonable cost limits. The system is designed using open-source tools, programming languages, and machine learning libraries, which significantly reduces development expenses. It does not require highly specialized or expensive hardware, as it can operate on standard computing systems, making it cost-effective for both academic and clinical environments. Additionally, maintenance and operational costs are minimized due to the use of widely supported technologies. Overall, the project is economically feasible as it provides an efficient solution with low investment while delivering high value in terms of improved diagnostic accuracy and healthcare support.

### 3.4.2 Technical Feasibility

Technical feasibility evaluates whether the proposed kidney tumor detection system can be developed using available technologies, tools, and resources. The system utilizes well-established programming languages such as Python and widely used machine learning and deep learning libraries like TensorFlow, Keras, and Scikit-learn, which are easily accessible and supported. The required techniques, including image preprocessing, fuzzy enhancement, feature extraction, and ensemble learning, are technically achievable with current computational capabilities. Additionally, the system can run on standard hardware with moderate specifications, making implementation practical..

### 3.4.3 Social Feasibility

Social feasibility evaluates the acceptance and impact of the proposed kidney tumor detection system among users and society. The system is designed to assist healthcare professionals in early and accurate diagnosis, which contributes to improved patient care and better treatment outcomes. Since kidney tumor detection is a critical medical need, the use of an automated and reliable system is likely to be well accepted by doctors, technicians, and healthcare organizations.

By providing accurate traffic predictions, the system enhances decision-making capabilities, enabling engineers to manage congestion, allocate resources efficiently, and improve overall network performance. Additionally, the automation of tumour prediction reduces manual workload, increasing productivity and operational efficiency. Improved network performance also leads to better service quality for end-users, resulting in higher customer satisfaction. Therefore, the system achieves high social feasibility as it is easy to adopt, beneficial, and well-received by stakeholders.

# CHAPTER-4

# SOFTWARE AND HARDWARE REQUIREMENTS

# 4 SOFTWARE AND HARDWARE REQUIREMENTS

## 4.1 SOFTWARE REQUIREMENTS

The software requirements for the proposed kidney tumor detection system define the essential tools, platforms, and technologies needed for its development and operation. The system is primarily developed using Python due to its simplicity and extensive support for machine learning and image processing tasks.

**Provided Software Requirements**

- Operating System: Windows 7 Ultimate
- Coding Language: Python

**Software Requirements for GNN-based Implementation**

- Operating System: Windows 10 / Windows 11 / Ubuntu 20.04+
- Programming Language: Python 3.8 or above

**Development Environment**

- Jupyter Notebook / VS Code / PyCharm

**Libraries & Frameworks**

- PyTorch – Deep learning model development
- PyTorch Geometric (PyG) – Graph Neural Network operations
- NumPy / Pandas – Data preprocessing
- NetworkX – Graph construction and analysis
- Matplotlib / Seaborn – Visualization

These software components collectively provide full support for data handling, graph processing, model training, and visualization required for the kidney tumor detection system.

## 4.2 HARDWARE REQUIREMENTS

The hardware requirements for the proposed kidney tumor detection system include essential components needed to ensure efficient processing and smooth system performance. A system with an Intel Core i5 or higher processor is recommended to handle computational tasks effectively.

**Hardware Specifications**

- Processor: Intel i5/i7 or AMD Ryzen 5/7
- RAM: Minimum 8 GB (16 GB recommended)
- Storage: 256 GB SSD or higher
- GPU: NVIDIA GPU with CUDA support (e.g., GTX 1650 / RTX series) for faster model training
- Monitor: Full HD Display
- Peripherals: Standard keyboard and optical mouse
- Keyboard – Standard input device
- Mouse – Two or three-button mouse

# CHAPTER-5

# SOFTWARE DESIGN

# 5 SOFTWARE DESIGN

## 5.1 SYSTEM ARCHITECTURE

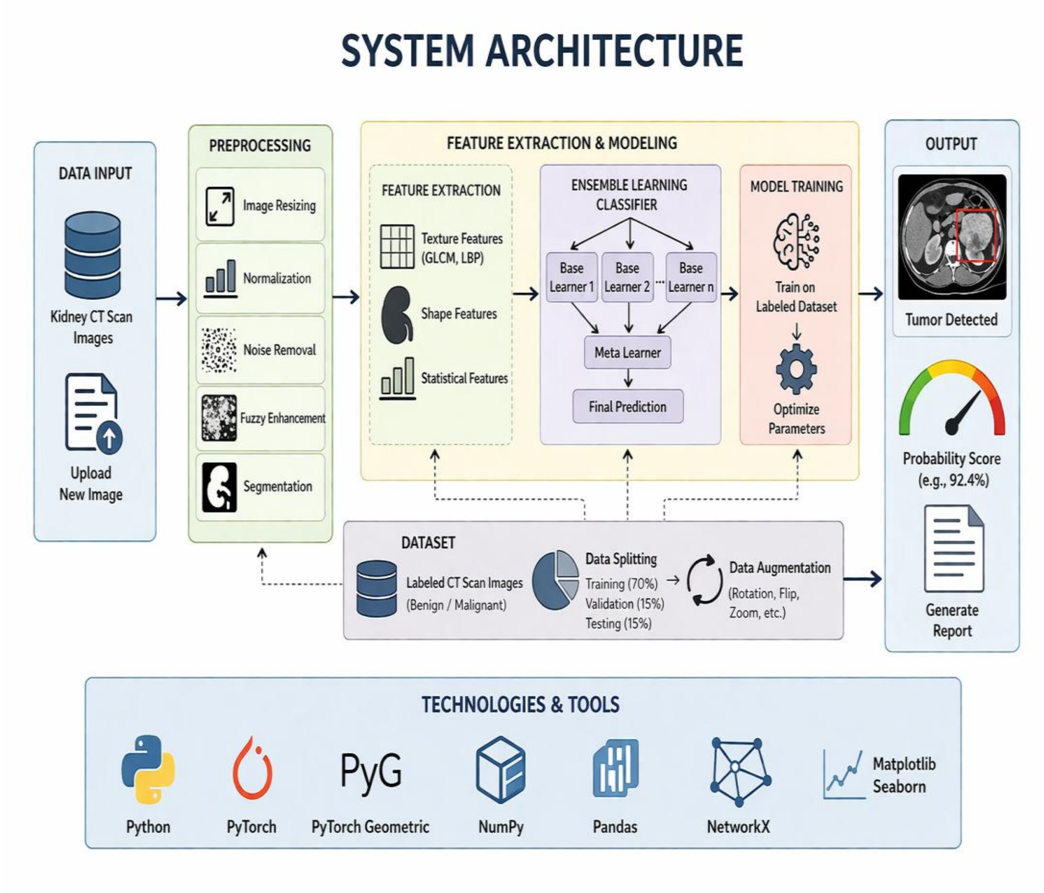


Fig 5.1 System Architecture

The Fig 5.1 The system architecture of the proposed kidney tumor detection system is designed to follow a structured and modular approach, ensuring efficient data processing and accurate prediction. The process begins with the input of CT scan images, which serve as the primary data source for analysis. These images are first passed through a preprocessing stage, where operations such as noise removal, normalization, and resizing are performed. Additionally, fuzzy-based enhancement techniques are applied to improve image contrast and highlight tumor regions, making them more distinguishable for further processing.Following preprocessing, the enhanced images are fed into the feature extraction module, which utilizes advanced deep learning models such as EfficientNetV2-B0 and ResNeSt-50.

The extracted features are then passed to the feature fusion stage, where they are combined and refined using techniques such as projection layers, batch normalization, and dropout. This step reduces redundancy, prevents overfitting, and enhances the generalization capability of the system. The fused features are then forwarded to the classification module, which employs a stacked ensemble of machine learning models including LightGBM, XGBoost, and CatBoost. A logistic regression meta-classifier is used to combine the outputs of these models and generate the final prediction.

Finally, the system produces an output indicating whether a tumor is present or not, along with visualization of the detected regions if applicable. The architecture is designed to be scalable, efficient, and adaptable to real-world clinical environments. By integrating advanced preprocessing, deep learning, and ensemble techniques, the system ensures high accuracy, reliability, and practical usability in kidney tumor detection.

## 5.2 Data Flow Diagram

The Data Flow Diagram (DFD) Fig 5.2 , are used to represent the flow of data within the proposed kidney tumor detection system in a clear and structured manner. They illustrate how data moves from input to output through various processing stages. The DFD helps in understanding the interaction between different components such as users, system processes, data storage, and outputs. It provides both a logical and visual representation of how the system processes medical images and generates predictions.

In the initial stage, the input data consists of CT scan images provided by the user or healthcare professional. These images are passed to the preprocessing module, where operations such as noise removal, normalization, resizing, and fuzzy enhancement are performed. The processed data is then forwarded to the feature extraction stage, where deep learning models analyze the images and extract meaningful features. This stage transforms raw image data into structured information suitable for classification.

Overall, the Data Flow Diagram ensures a smooth and efficient flow of data throughout the system, helping to identify how each component interacts and contributes to the final prediction. It simplifies system understanding, improves design clarity, and supports effective implementation of the kidney tumor detection system.

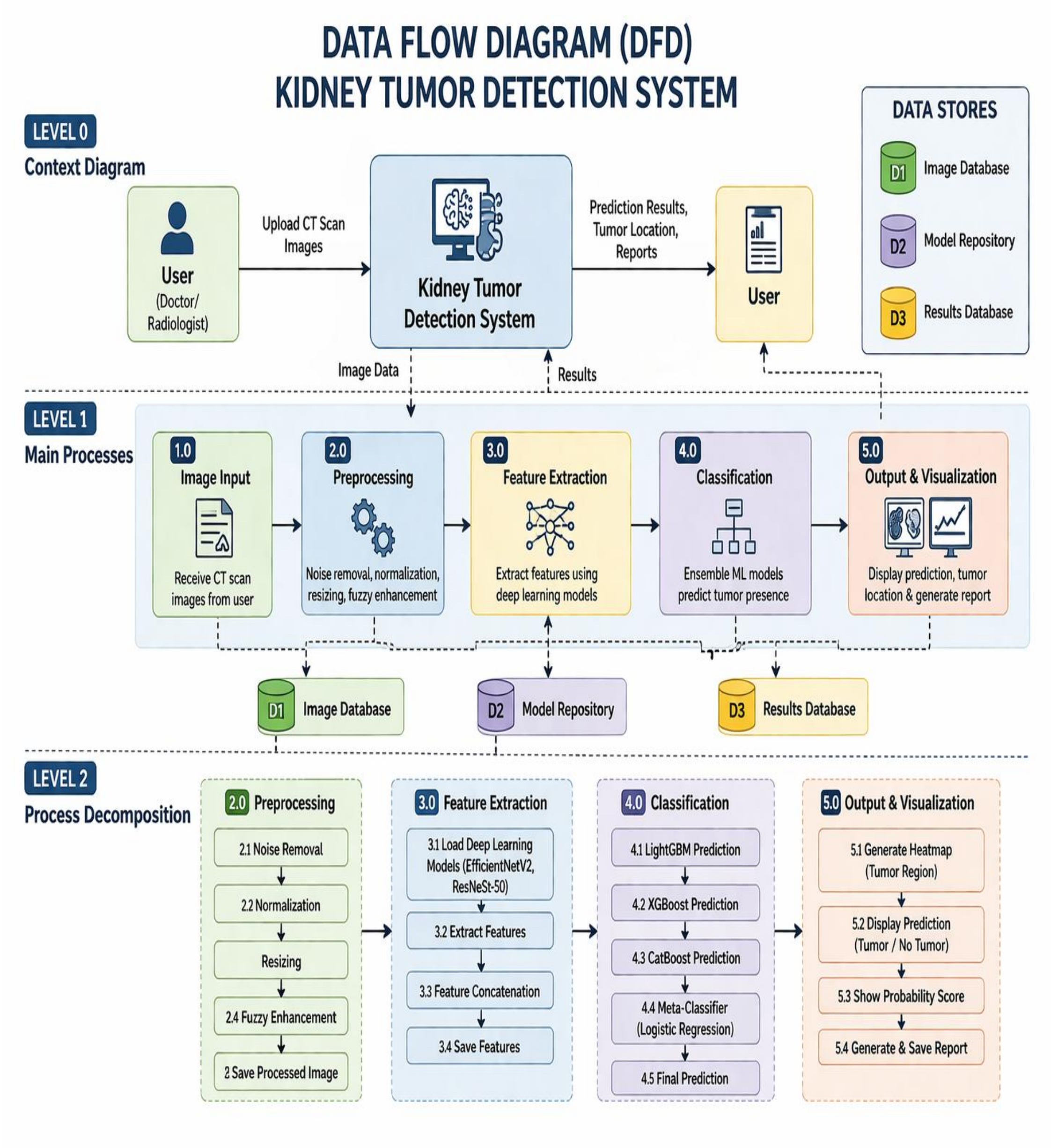


Fig 5.2 Data Flow Diagram

### 5.2.1 Key Components of Data Flow:

- User Input:

  Users provide comma-separated numerical data through the web interface.

- Input Handling:

  Flask application receives and processes the input data.

- Data Preprocessing:

  Input data is normalized using a standard scaler for model compatibility.

- GNN Model Prediction:

  The processed data is passed through GAT-based neural network layers to generate predictions.

- Label Mapping:

  Numeric predictions are converted into meaningful class labels.

- Prediction Display:

  The predicted class is shown to the user on the interface.

- Database Storage:

  Input data and predictions are stored in an SQLite database.

- Predictions History:

  Stored records are displayed for user reference.

- Graph Generation:

  Visual representations (pie, bar, line charts) are generated from stored prediction data.

## 5.3 UML Diagrams

UML (Unified Modeling Language) is a standardized language used for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML helps in representing the design of systems and understanding their components. Created by the Object Management Group (OMG), UML 1.0 was proposed in January 1997. UML is closely associated with object-oriented analysis and design.

The Behavioral UML diagrams describe the behavior of the system, its actors, and the interaction between the components. On the other hand, Structural UML diagrams depict the static structure of the system, showing its components and relationships. UML has been integrated as a standard by OMG, and its primary goals are to provide a formal basis for understanding modeling languages, offer a ready-to- use expressive tools.

**Goals of UML:**

To provide a standard visual representation of the system design.
To simplify understanding of system architecture for developers and reviewers. To improve communication among team members during development.
To model both structural and behavioral aspects of the system.
To document the system clearly for future reference and maintenance

**Types of UML Diagrams**

The different types are broken down as follows:

1. Sequence diagram
2. Use case Diagram
3. Activity diagram
4. Collaboration diagram
5. Class Diagram
6. Deployment Diagram

### 5.3.1. SEQUENCE DIAGRAM

A sequence diagram Fig 5.3.1 simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

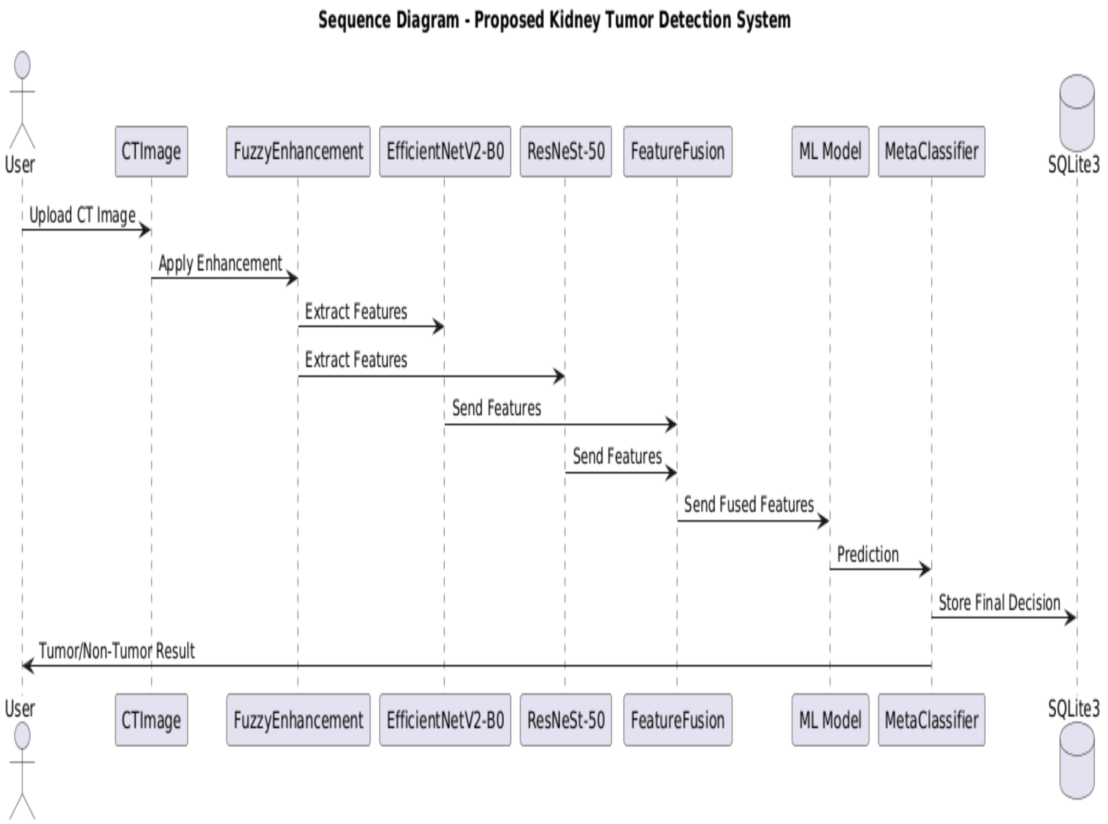


Fig 5.3.1 Sequence Diagram

## 5.3.2 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases, and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted

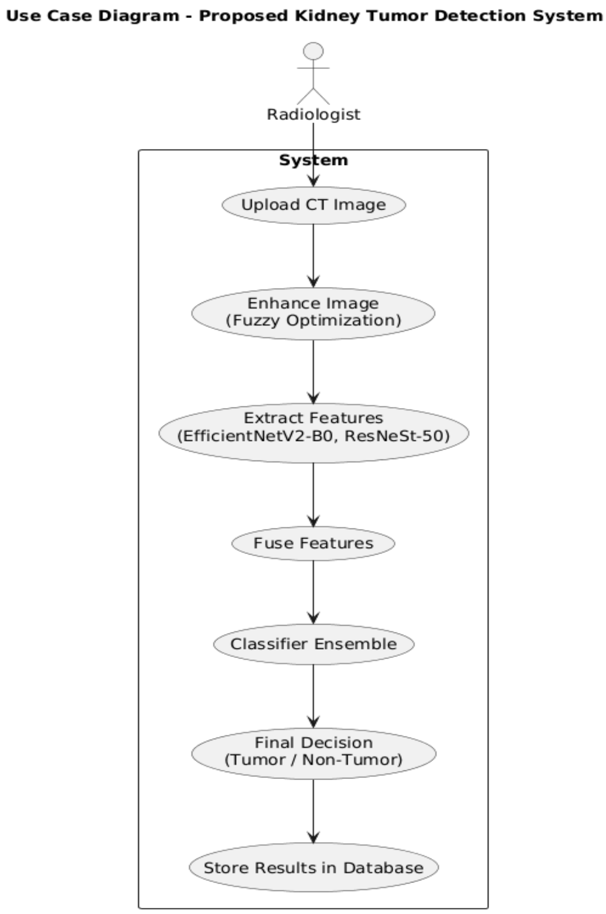


Fig 5.3.2 Use Case Diagram

## 5.3.3 COLLABORATION DIAGRAM

The collaboration diagram represents the interaction and communication between different components of the proposed kidney tumor detection system. It illustrates how various elements such as the user, system interface, preprocessing module, feature extraction models, classification models, and database work together to achieve the final prediction.

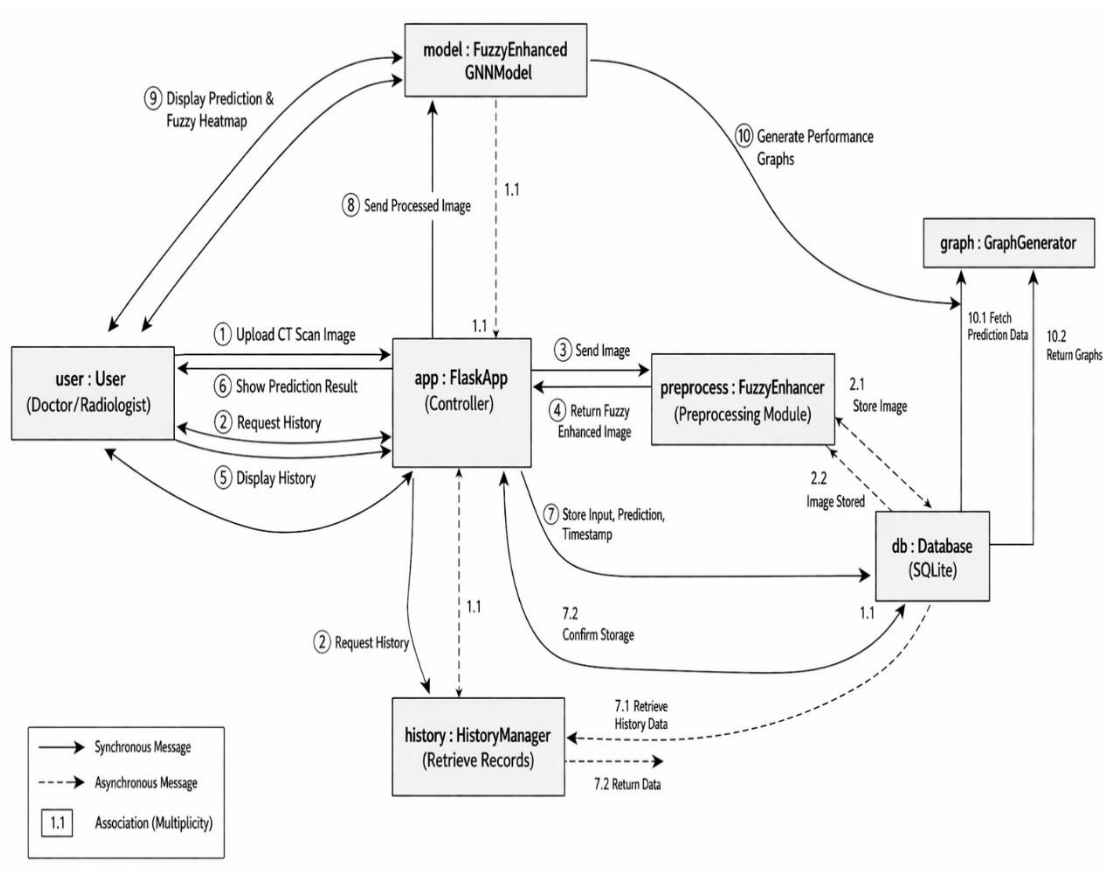


Fig 5.3.3 Collaboration Diagram

## 5.3.4 ACTIVITY DIAGRAM

The Fig 5.3.4 Activity diagrams are graphical representations of work flows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step work flows of components in a system. An activity diagram shows the overall flow of control.

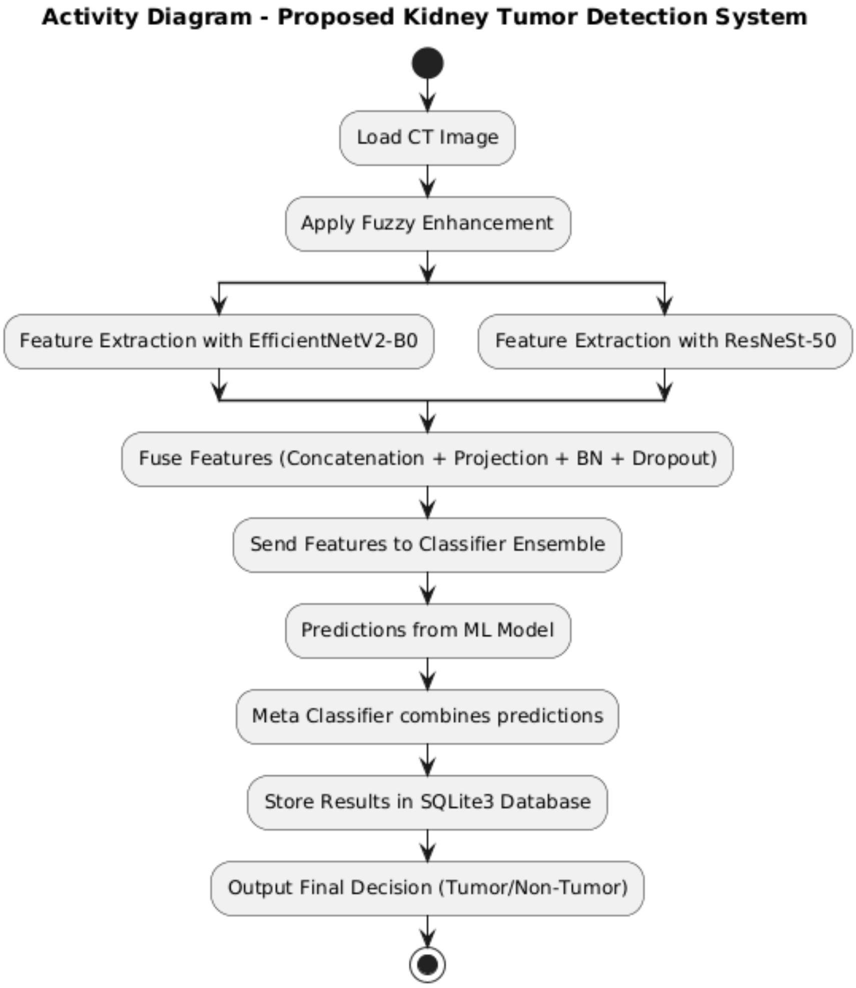


Fig 5.3.4 Activity Diagram

## 5.3.5 CLASS DIAGRAM

The class diagram of the fuzzy enhanced kidney tumor detection system represents the structure and relationships between different classes involved in the system. The main classes include User, FlaskApp (Controller), Preprocessing (FuzzyEnhancer), FeatureExtractor, Model (CNN/GNN), Database, GraphGenerator, and HistoryManager. The User class interacts with the FlaskApp to provide input and receive results. The FlaskApp acts as the central controller, coordinating communication between all components. The Preprocessing class handles image enhancement using fuzzy techniques, while the FeatureExtractor and Model classes are responsible for extracting features and performing tumor prediction. The Database class stores input data, predictions, and history records. The GraphGenerator class is used for generating visual reports, and the HistoryManager manages past records. These classes are interconnected through associations, ensuring smooth data flow and efficient system operation.

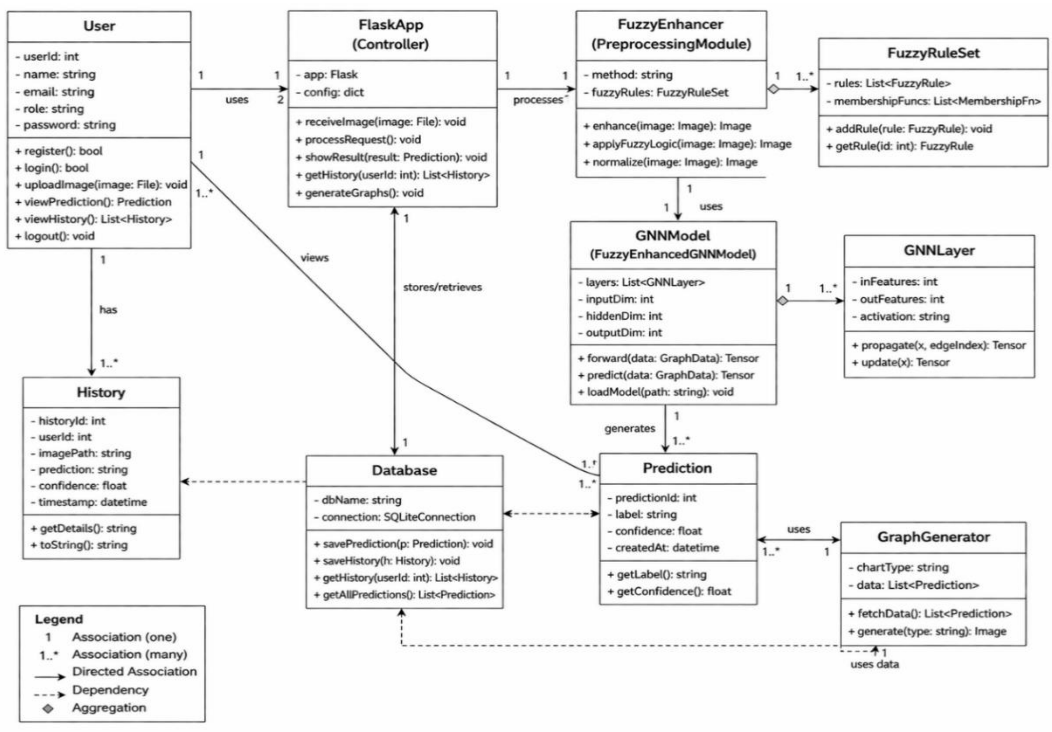


Fig 5.3.5 Class Diagram

### 5.3.6 DEPLOYMENT DIAGRAM

The deployment diagram represents the physical arrangement of hardware and software components in the kidney tumor detection system. It includes user devices, a web server, application server, and database server connected through a network. The user interacts with the system via a web interface, which communicates with the Flask-based application hosted on the server. The application processes input data, runs the trained model, and stores results in the database. The system may also utilize cloud or local servers for model execution and data storage. Overall, the deployment diagram illustrates how the system is deployed and operates in a real-world environment.

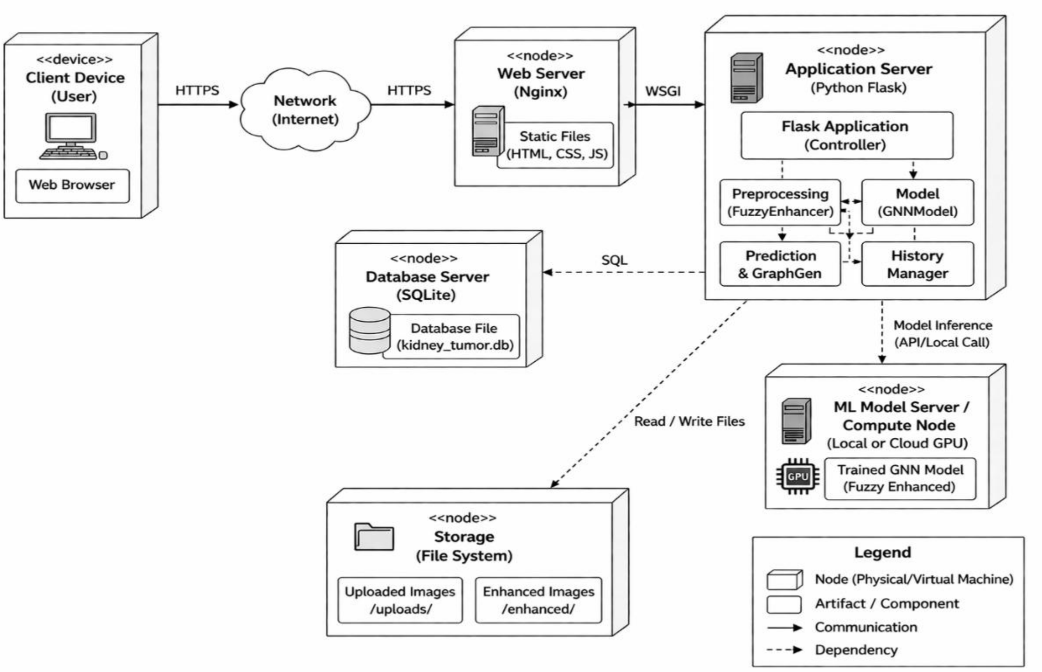


Fig 5.3.6 Deployment Diagram

# CHAPTER-6

# IMPLEMENTATION

# AND

# CODING

# 6. IMPLEMENTATION AND CODING

## 6.1 Source Code

**App.py**

```
from __future__ import division, print_function

# coding=utf-8

import sys

import os

import glob

import re

import numpy as np

# Keras"""

From tensorflow.keras.applications.imagenet_utils
import preprocess_input, decode_predictions

from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing import image

# from tensorflow.applications.imagenet_utils import
preprocess_input, decode_predictions

# from tensorflow.models import load_model

# from tensorflow.preprocessing import image

# Flask utils

from flask import Flask, redirect, url_for, request,
render_template

from werkzeug.utils import secure_filename

import sqlite3

app = Flask(__name__)

UPLOAD_FOLDER = 'static/uploads/'
```

```
# Get the path to the normal and pneumonia sub-
directories
# allow files of a specific type

ALLOWED_EXTENSIONS = set(['png', 'jpg', 'jpeg'])

# function to check the file extension

def allowed_file(filename):

   return '.' in filename and \

          filename.rsplit('.', 1)[1].lower() in
ALLOWED_EXTENSIONS

model_path2 = 'kidney_model.h5' # load .h5 Model

CTS = load_model(model_path2)

from tensorflow.keras.preprocessing.image import
load_img, img_to_array

def model_predict2(image_path,model):

   print("Predicted")

   image = load_img(image_path,target_size=(28,28))

   image = img_to_array(image)

   image = image/255

   image = np.expand_dims(image,axis=0)

   result = np.argmax(model.predict(image))

   if result == 0:

   return "Cyst","after.html"

elif result == 1:

   return "Normal","after.html"
```

```
elif result == 2:
    return "Stone","after.html"
elif result == 3
return "Tumor","after.html"
@app.route("/")
@app.route('/home')
def home():
        return render_template('home.html')
@app.route('/predict2',methods=['GET','POST'])
def predict2():
    print("Entered")
    print("Entered here")
    file = request.files['files'] # fet input
    filename = file.filename
    print("@@ Input posted = ", filename)
    file_path = os.path.join(UPLOAD_FOLDER,
filename)
    file.save(file_path)
    print("@@ Predicting class......")
    pred, output_page = model_predict2(file_path,CTS)
    return render_template(output_page, pred_output =
pred, img_src=UPLOAD_FOLDER + file.filename)
if __name__ == '__main__':
    app.run(debug=False)
```

**Predict.py**

```
import torch

import torch.nn.functional
as F import pickle
import numpy as np

from torch_geometric.data
import Data from
torch_geometric.nn import
GATConv
# Load the standard scaler
# Get the path to the normal and pneumonia sub-directories
normal_cases_dir = train_dir / 'Normal'
x, edge_index = data.x, for img in normal_cases:
img = cv2.imread(str(img))
img = cv2.resize(img, (28,28))
if img.shape[2] ==1:
img = np.dstack([img, img, img])
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img=np.array(img)
img = img/255
label ='Normal'
train_data.append(img)
train_labels.append(label)
for img in Stone_cases:
img = cv2.imread(str(img))
img = cv2.resize(img, (28,28))
if img.shape[2] ==1:
img = np.dstack([img, img, img])
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img=np.array(img)
img = img/255
```

```
img = cv2.imread(str(img))
img = cv2.resize(img, (28,28))
if img.shape[2] ==1:
img = np.dstack([img, img, img])
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

**Predictions.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Predictions</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.mi
n.css">
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container">
    <a class="navbar-brand" href="#">GNN Predictor</a>
    <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ms-auto">
```

```
<li class="nav-item"><a class="nav-link" href="{{ url_for('predict')
}}">Predict</a></li>
<li class="nav-item"><a class="nav-link active" href="{{ url_for('predictions')
}}">Predictions</a></li>
<li class="nav-item"><a class="nav-link" href="{{ url_for('graphs')
}}">Graphs</a></li>
</ul>
</div>
</div>
</nav>
<div class="container mt-5">
<h2 class="text-center">Stored Predictions</h2>
<table class="table table-bordered mt-4">
<thead class="table-dark">
nav aria-label="Page navigation">
<ul class="pagination justify-content-center">
{% if current_page > 1 %}
<li class="page-item"><a class="page-link" href="{{
url_for('predictions', page=current_page- 1)
}}">Previous</a></li>
{% endif %}
<nav aria-label="Page navigation">
<ul class="pagination justify-content-center">
{% if current_page > 1 %}
<li class="page-item"><a class="page-link" href="{{
url_for('predictions', page=current_page- 1)
```

```
                }
                }">Previous</a></li>
{% endif %}</tbody>
            <nav aria-label="Page navigation">
              <ul class="pagination justify-content-center">
                {% if current_page > 1 %}
          <li    class="page-item"><a    class="page-link"    href="{{
 url_for('predictions',
          page=current_page-1) }}">Previous</a></li>
                {% endif %}
                {% for p in range(1, total_pages+1) %}
                <li class="page-item {% if current_page
                 == p %}active{% endif %}">
                  <a class="page-link" href="{{ url_
                  for('predictions', page=p) }}">{{ p }}</a>
                </li>
                {% endfor %}
                {
                % if current_page < total_pages %}
          <li       class="page-item"><a   class="page-link"
                                          href="{{
                 url_for('predictions', page=current_page+1)
              </ul>
            </nav>
          </div>
 <script
 src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js">
 </script>
 </body>
</html>
```

**Style.css**

```
html{
    height: 100%;
    margin: 0;
}body{
    font-family: Arial, Helvetica,sans-serif;
    text-align: center;
    margin: 0;
    padding: 0;
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
}.container{
    padding: 30px;
    position: relative;
    background: linear-gradient(45deg, #90963b, #90963b, #90963b);
    background-size: 500% 500%;
```

```
  0%{
      background-position: 0 50%;
50%{
      background-position: 100% 50%;
   }
   100%{
      background-position: 0 50%;
   }
}
.container-heading{
   margin: 0;
}
.heading_font{
   color: #ffffff;
   font-family: 'Roboto', normal;
   font-size: 35px;
   font-weight: normal;
}
description p{
   color: #ffffff;
   font-family: normal;
   font-size: 20px;
   margin: -5px 0 0;
```

```
}
* Text Area */
.ml-container{
    margin: 30px 0;
    flex: 1 0 auto;
}
.form-input {
    text-align: center;
    width: 350px;
    height: 25px;
    margin-bottom: 5px;
}
/* Predict Button */
.my-cta-button{
    background: #f9f9f9;
    border: 2px solid #000000;
    border-radius: 1000px;
    box-shadow: 3px 3px #8c8c8c;
    margin-top: 10px;
```

```
padding: 10px 36px;
    color: #000000;
    display: inline-block;
    font: normal bold 20px/1 "Calibri", sans-serif;
    text-align: center;}
my-cta-button:hover{
    color: #4d089a;
    border: 2px solid #4d089a;}
.my-cta-button:active{
    box-shadow: 0 0;
}
/* Contact */
.contact-icon{
    color: #ffffff;
    padding: 7px;}
.contact-icon:hover{
    color: #8c8c8c;}
.footer{
    flex-shrink: 0;
```

```
  position: relative;

   padding: 20px;

  background: linear-gradient(45deg, #90963b, #90963b, #90963b);

   background-size: 500% 500%;

   animation: change-gradient 10s ease-in-out infinite;

}

.footer-description{

   color: #ffffff;

   margin: 0;

   font-size: 12px;

}

/* Result */

.results{

   padding: 30px 0 0;

   flex: 1 0 auto;}

.danger{

   color: #ff0000;}

.safe{

   color: green;}

}
```

## 6.2. IMPLEMENTATION

The implementation of the kidney tumor detection system is carried out using Python and various machine learning and deep learning libraries such as TensorFlow, Keras, NumPy, and OpenCV. The process begins with loading and preprocessing CT scan images, including resizing, normalization, and noise reduction, followed by fuzzy-based enhancement to improve image quality. The enhanced images are then passed through deep learning models for feature extraction, and the extracted features are combined using feature fusion techniques. A stacked ensemble of classifiers is applied to perform accurate tumor classification. The system is trained and tested using labeled datasets, with techniques like data augmentation and SMOTE used to handle data imbalance. Finally, the model is evaluated using performance metrics such as accuracy, precision, recall, and F1-score, and the results are displayed through a user-friendly interface for effective diagnosis.

### A. Fuzzy-Based Image Enhancement

Fuzzy-based image enhancement is a technique used to improve the quality of medical images by handling uncertainty and vagueness present in CT scan data. In this project, fuzzy logic is applied to enhance contrast and highlight tumor regions, making them more distinguishable for further analysis. The method uses membership functions to represent pixel intensity levels and applies fuzzy rules to enhance important features while suppressing noise.

The process can be mathematically expressed as:

$$I_{enhanced} = F^{-1}(F(I_{input}))$$

where $F$represents fuzzification and $F^{-1}$represents defuzzification. Through this process, the system improves image clarity, contrast, and tumor region visibility

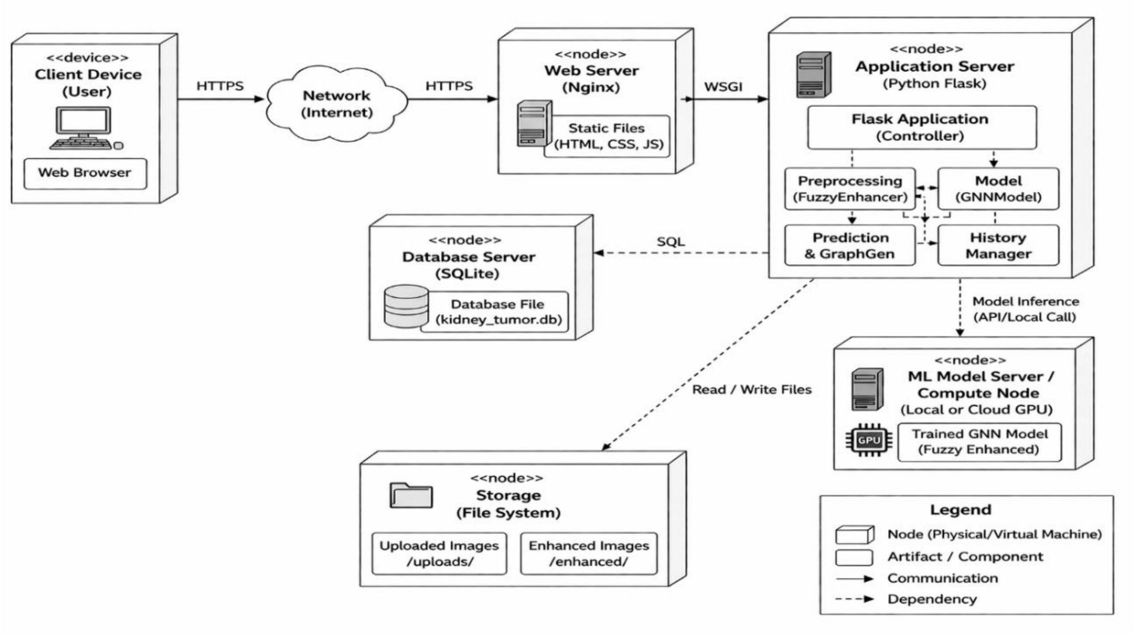


Fig 6.2.1 Architecture of Fuzzy-Based Image Enhancement

**B. Convolutional Neural Network (CCN)**

Convolutional Neural Networks (CNNs) are deep learning models widely used for image analysis and feature extraction. In this project, CNN models such as EfficientNetV2-B0 and ResNeSt-50 are used to extract meaningful features from enhanced CT images. These models automatically learn spatial patterns, textures, and structures that are essential for identifying tumor regions.

The working of CCN involves three main steps: (1) constructing the adjacency matrix of the graph, (2) normalizing it to avoid scale issues, and (3) aggregating neighbor features followed by a nonlinear transformation. The standard CCN propagation rule is given by:

$$FeaturMap=Input*Kernel$$

where the kernel extracts important features from the input image. CNNs are highly effective in capturing both low-level and high-level features, improving classification accuracy.

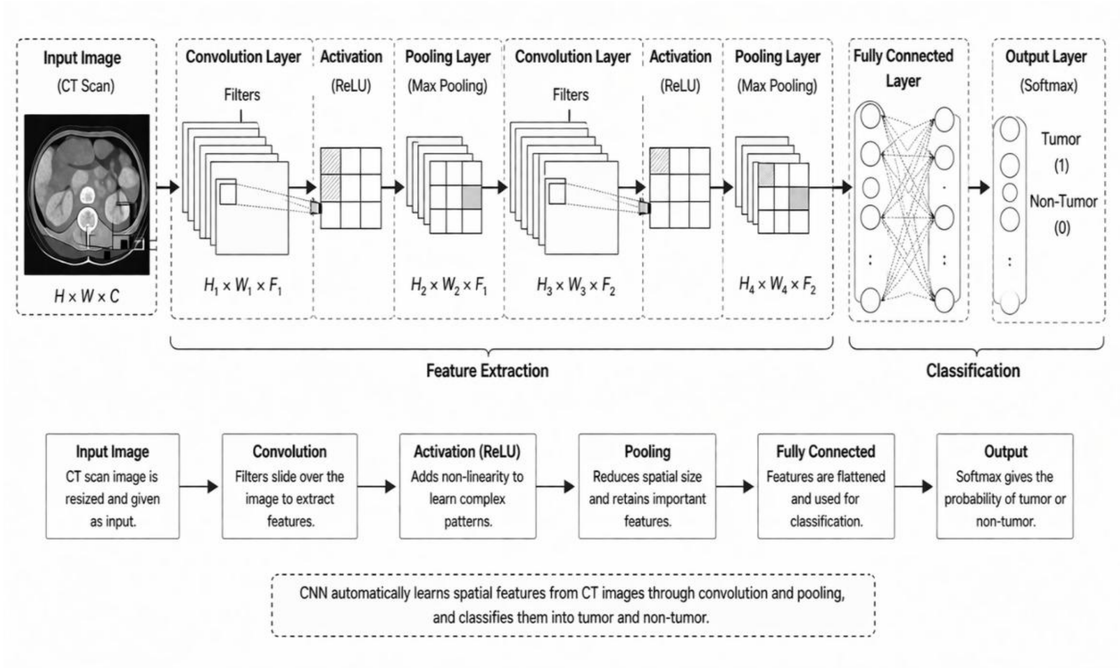


Fig. 6.2.2 Illustration of CCN operation

**How Fuzzy + CCN Work Together in the Project**

In the proposed system, fuzzy enhancement and CNN models work together to improve detection performance. The combined workflow is as follows:

- The cellular network is first modeled as a graph.
- The CT scan image is first enhanced using fuzzy logic
- Enhanced images improve visibility of tumor regions
- The processed images are fed into CNN models
- CNN extracts deep features from the images
- Extracted features are passed to classification models
- Final prediction (Tumor / Non-Tumor) is generated

This integration allows the model to:

- Capture spatial correlations between base stations
- Improve image quality and feature clarity
- Reduce noise and uncertainty
- Enhance robustness in medical image analysis

### 6.2.1 Frontend Implementation

The frontend of the system is developed using HTML, CSS, and Bootstrap, which provides a clean, responsive, and user-friendly interface. The main page allows users to enter LTE network parameters such as longitude, latitude, speed, signal strength values like RSRP, RSRQ, RSSI, and SNR, as well as CQI, bitrate, and other mobility-related features. The input is taken in a simple comma-separated format, making it easy for users to provide data without complexity.

In addition to the input page, the system includes a predictions page where previously generated results are displayed in a structured table format. This allows users to review historical predictions along with their corresponding input values. Pagination is implemented to manage large amounts of data efficiently and improve readability. Another important part of the frontend is the graphs page, which displays visual representations of prediction results using pie charts, bar charts, and line graphs. These visualizations help users understand traffic distribution, identify patterns, and analyze trends in LTE network behavior. A navigation bar is included across all pages, allowing users to easily switch between prediction, stored results, and graphical analysis. The use of Bootstrap ensures that the interface is responsive and works smoothly across different devices such as desktops, tablets, and mobile phones.

### 6.2.2 Backend Implementation

The backend implementation of the kidney tumor detection system is developed using Python, leveraging powerful libraries such as TensorFlow, Keras, NumPy, Pandas, and OpenCV. The backend is responsible for handling all core functionalities, including data processing, model training, feature extraction, and prediction. It manages the workflow from receiving input CT scan images to generating the final tumor detection result. The system is designed in a modular manner to ensure scalability, maintainability, and efficient execution of each stage in the pipeline.

The backend begins with data preprocessing, where raw CT images are resized, normalized, and cleaned to remove noise and inconsistencies.

### 6.2.3 Model Integration

Model integration is a crucial stage in the kidney tumor detection system, where multiple deep learning and machine learning models are combined to work as a unified framework. The system integrates advanced feature extraction models such as EfficientNetV2-B0 and ResNeSt-50, which operate in parallel to capture diverse and complementary features from CT scan images. These models are pre-trained and fine-tuned on medical imaging data to improve their ability to identify tumor-specific patterns.

After feature extraction, the outputs from both models are merged using a feature fusion mechanism. This integration step includes concatenation followed by a projection layer with batch normalization and dropout to reduce dimensionality and prevent overfitting. By combining features from multiple models, the system creates a more robust and discriminative representation, which enhances the accuracy of tumor detection.

The integrated feature set is then passed to a stacked ensemble of machine learning classifiers, including LightGBM, XGBoost, and CatBoost. Each classifier independently analyzes the features and produces predictions. These predictions are further combined using a logistic regression meta-classifier, which learns the optimal way to aggregate outputs from individual models. This layered integration ensures that the strengths of each model are utilized effectively while minimizing their weaknesses.

### 6.2.4 Deployment

The deployment of the kidney tumor detection system involves making the trained model accessible for real-time use by medical professionals through a user-friendly interface. The system is deployed as a web-based or desktop application, where users such as doctors or technicians can upload CT scan images and receive instant predictions. The backend models are integrated with frameworks like Flask or Django, which handle user requests, process inputs, and return results efficiently. During deployment, the trained models are saved and loaded into the production environment to ensure fast and reliable predictions. The system is optimized to reduce latency and support real-time or near real-time analysis of medical images.

CHAPTER-7

# SYSTEM TESTING

# 7. SYSTEM TESTING

System testing is a pivotal phase in the software development lifecycle (SDLC), focusing on evaluating the software as a complete, integrated system rather than as individual modules. Unlike unit or integration testing, which target specific components or their interactions, system testing validates that all parts of the system work together correctly and that the software meets both functional and non-functional requirements. The primary goal is to identify defects that may not have been detected in earlier testing stages, ensuring that the system behaves as expected under real-world conditions and satisfies business objectives and user expectations.

System testing encompasses a variety of testing types, each addressing different aspects of system quality. Functional testing verifies that the system performs all specified operations correctly, while performance testing evaluates the system's responsiveness, stability, and resource usage under different workloads. Stress and load testing push the system beyond normal operational limits to assess its robustness and error-handling capabilities. Security testing identifies vulnerabilities and ensures that sensitive data is protected, and usability testing examines the user interface and experience to confirm that the system is intuitive and accessible. Each type of test uses distinct approaches, such as black-box testing, white-box testing, or scenario-based testing, depending on the goals and requirements.

In addition to defining test types, effective system testing relies on well-planned testing strategies. Test cases are designed to cover critical workflows, edge cases, and potential failure points, ensuring comprehensive coverage of the system's functionality. Automation tools can be employed for repetitive or large-scale tests to increase efficiency and reduce human error. A systematic approach to system testing not only detects defects early but also provides confidence that the software is reliable, stable, and ready for deployment. By thoroughly validating the integrated system, this phase ensures that the software delivers the intended value to end-users and aligns with organizational goals.

## 7.1 Types of Tests

### a. Unit Testing

Unit testing is performed to verify individual modules of the system independently. Each component, including image preprocessing, fuzzy enhancement, feature extraction, and classification, is tested to ensure it functions correctly. This helps in detecting and fixing errors at an early stage before integration.

### Key Features:

- Tests all individual modules separately.
- Helps in early detection of errors.
- Performed during the development phase before integration.
- Requires understanding of internal code structure.

### b. Integration Testing

Integration testing focuses on combining different modules of the system and verifying their interaction. It ensures that data flows correctly between components such as preprocessing, feature extraction, and classification without any errors.

**Key Features:**

- Validates interactions between integrated components.
- Verifies interaction between modules.
- Ensures proper data flow across components.
- Detects communication errors between modules.
- Performed after unit testing.

### c. Functional Testing

Functional testing verifies that the application meets the functional requirements as outlined in the business and technical documentation.

### Key Features:

- Validates system functionality based on requirements.
- Checks input and output correctness

## d. System Testing

System testing examines the complete software system to validate its overall behavior. It ensures the system is configured correctly and produces known and predictable results.

- Tests the entire software application as a whole.
- Focuses on process flows, integration points, and expected system behaviors.
- Conducted after integration testing.

### e. White Box Testing

White box testing involves testing software with knowledge of its internal structures and code implementation.

**Key Features:**

- Evaluates internal logic and code structure.
- Requires in-depth knowledge of the software.
- Used to test areas that cannot be reached using black box testing.

### f. Black Box Testing

Black box testing is performed without knowledge of the internal workings of the application. Testers focus on inputs and expected outputs without considering the implementation.

**Key Features:**

- Does not require code knowledge.
- Based on functional specifications and requirements.
- Helps ensure that software functions correctly from an end-user perspective.

**Test Strategy and Approach**

Testing is performed using both manual and automated approaches to ensure comprehensive coverage.

**Key Testing Objectives:**

- Validate that all fields accept correct input values.
- Ensure links and navigation work as expected.
- Ensure input screens, messages, and responses load without delays.

**Features to be Tested**

- Correct format validation for all input fields.
- Prevention of duplicate entries.
- Verification that all links navigate to the coIntegration Testing

Integration testing verifies that multiple software components interact without defects.

**Key Features:**

- Detects interface and communication errors between components.
- Ensures smooth data exchange between integrated modules.

**Test Results**

- All test cases executed successfully.
- No defects encountered during testing.

### g. Acceptance Testing

User Acceptance Testing (UAT) is conducted to confirm that the system meets business requirements and user expectations before final deployment.

**Key Features:**

- Ensures all functional requirements are met.
- Requires active participation from end-users.
- Confirms the system is ready for production deployment.

**Test Results:**

- All acceptance test cases passed successfully.
- No defects encountered, confirming system readiness.

System testing is an essential step in software development, ensuring that the application meets user expectations and business needs. Various types of testing, including unit, integration, functional, and system testing, help identify potential issues before deployment. The results from this testing phase confirm that the system is functioning correctly and is ready for production.

## 7.2 Testing Strategies

The testing process is carried out in a systematic and step-by-step manner. Initially, unit testing is performed to verify individual modules such as image preprocessing, fuzzy enhancement, and feature extraction. After this, integration testing is conducted to ensure that all modules interact properly. Functional and system testing are then applied to validate the complete.

Finally, user acceptance testing is performed to ensure that the system satisfies real-world user expectations.

### 7.2.1 Testing Approach

The testing process is carried out in a step-by-step manner, starting from individual components and moving towards the complete system. Initially, unit testing is performed to verify each module independently. After that, integration testing is conducted to ensure that all modules work together properly. Functional and system testing are then used to validate the overall behavior of the system. Finally, user acceptance testing is performed to confirm that the system meets user expectations.

### 7.2.2 Testing Methods

Both manual and automated testing methods are used in the system to ensure comprehensive testing. Manual testing is mainly used to check the user interface, navigation, and overall usability of the application. Automated testing is used for repetitive tasks such as validating input processing, checking prediction outputs, and ensuring system performance. This combination improves testing efficiency and accuracy.

### 7.2.3 White Box and Black Box Testing

The system uses both white box and black box testing techniques. White box testing focuses on the internal structure of the code, ensuring that all logic, conditions, and execution paths are tested properly. Black box testing, on the other hand, evaluates the system from the user's perspective without considering internal implementation. It checks whether the system produces correct outputs for given inputs.

### 7.2.4 Functional and System Testing Strategy

Functional testing is performed to verify that all system features work according to the specified requirements. It ensures that valid inputs are accepted, invalid inputs are rejected, and correct predictions are generated. System testing evaluates the entire application as a whole.

### 7.2.5 Integration Testing Strategy

Integration testing focuses on checking the interaction between different modules such as frontend, backend, database, and the GNN model. It ensures that data flows correctly between these components and that there are no communication errors. This step is important to confirm that all modules are properly connected and functioning together.

### 7.2.6 User Acceptance Testing (UAT)

User Acceptance Testing is conducted with the involvement of end-users to verify that the system meets real-world requirements. Users interact with the system and provide feedback on usability, accuracy, and performance. Based on this feedback, necessary improvements are made before final deployment.

### 7.2.7 Testing Objectives

The main objective of testing is to ensure that the system is accurate, reliable, and efficient. It checks proper input validation, smooth navigation, correct prediction results, and proper data storage. It also ensures that the system performs well under different conditions and provides a good user experience.

## 7.3 Test Cases

| Test Case ID | Test Description | Input | Expected Output | Remarks |
|---|---|---|---|---|
| TC01 | Valid CT image input (Tumor case) | CT image with tumor features | Predicted label: Tumor | Tests end-to-end tumor detection |
| TC02 | Valid CT image input (Normal case) | CT image without tumor | Predicted label: Normal | Tests correct classification of normal kidney |
| TC03 | Valid CT image input (Noisy image) | CT image with noise/distortion | Predicted label: Tumor/Normal | Tests robustness of fuzzy enhancement |
| TC04 | Invalid input format | Text file instead of image | Error message: Invalid input | Tests input validation |
| TC05 | Empty input | No image uploaded | Error message: No input provided | Tests system response to missing input |

| Test Case ID | Test Description | Input | Expected Output | Remarks |
|---|---|---|---|---|
| TC06 | Low-resolution image input | Blurred or low-quality CT image | Predicted label with reduced confidence | Tests performance on poor-quality data |
| TC07 | Multiple image input | Batch of CT images | Predictions for all images | Tests batch processing capability |
| TC08 | Edge case (small tumor) | CT image with very small tumor | Predicted label: Tumor | Tests sensitivity of detection |
| TC09 | High-resolution image input | Large size CT scan | Correct prediction without delay | Tests system performance |
| TC10 | Feature extraction validation | Preprocessed image input | Extracted features successfuly | Tests CNN feature extraction stage |

**Table 7.3 Test Cases**

**Test Case 1:**


Fig 7.3.1 Valid Prediction Input for Tumor

**Description:** The Fig 7.3.1 , displays the prediction result after submitting a valid CT image containing a kidney tumor. The system successfully processes the input image and shows “Predicted label: Tumor” on the screen. It confirms that the complete workflow from image input to tumor detection is functioning correctly.

**Test Case 2:**

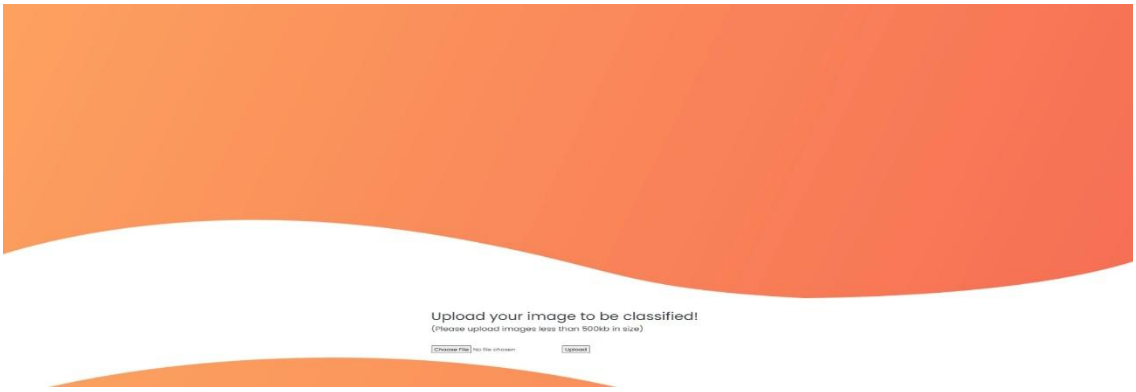


Fig 7.3.2 Valid Prediction Input for Normal Kidney

**Description:** The Fig 7.3.2 displays the prediction result after submitting a normal CT image without any tumor. The system correctly identifies the input and shows "Predicted label: Normal" on the screen. This confirms that the system accurately distinguishes between tumor and non-tumor cases.

**Test Case 3 :**

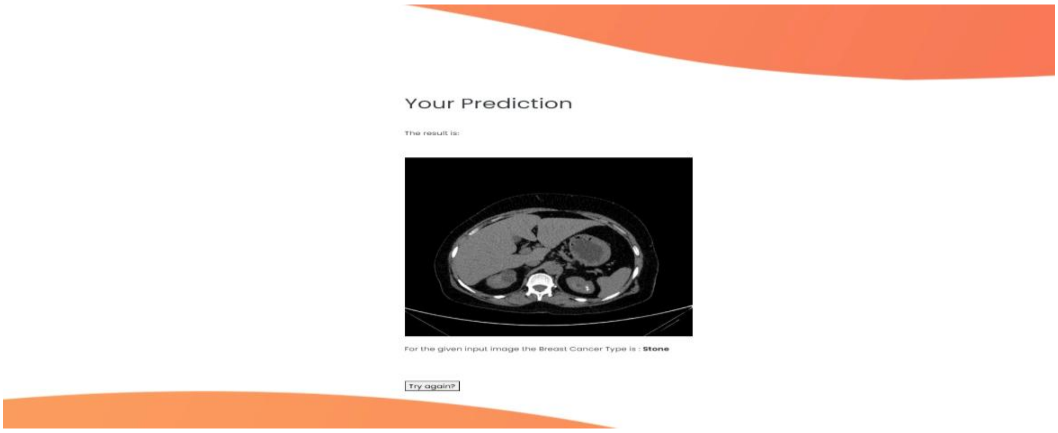


Fig 7.3.3 Valid Prediction Input for Noisy CT Image

**Description:** In Fig 7.3.3, the test case verifies that Displays the system output when a noisy or low-quality CT image is provided as input. The system applies fuzzy enhancement and still produces a prediction result. This confirms that the system is robust and can handle images with noise or distortions.

**Test Case 4 :**

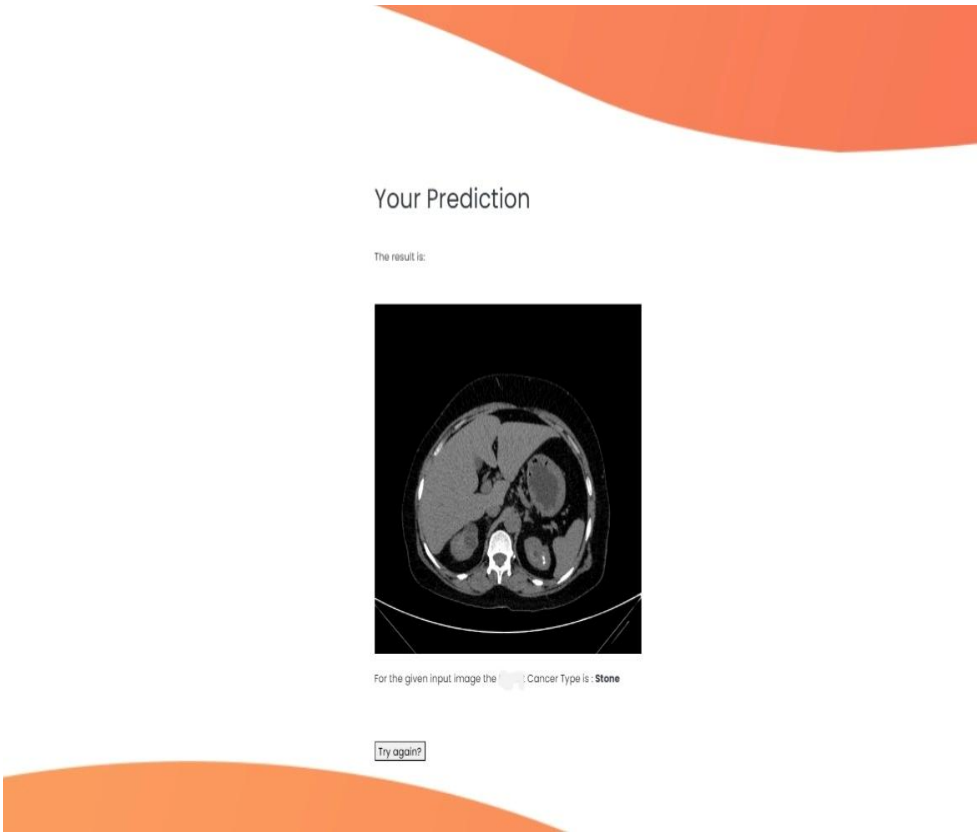


Fig 7.3..4 Invalid Input Format Handling

**Description:** In Fig 7.3.4, the test case verifies that displays the error message when an invalid input format (such as a non-image file) is submitted. The system rejects the input and shows an appropriate error message. This confirms proper input validation and error handling.

**Test Case 5 :**

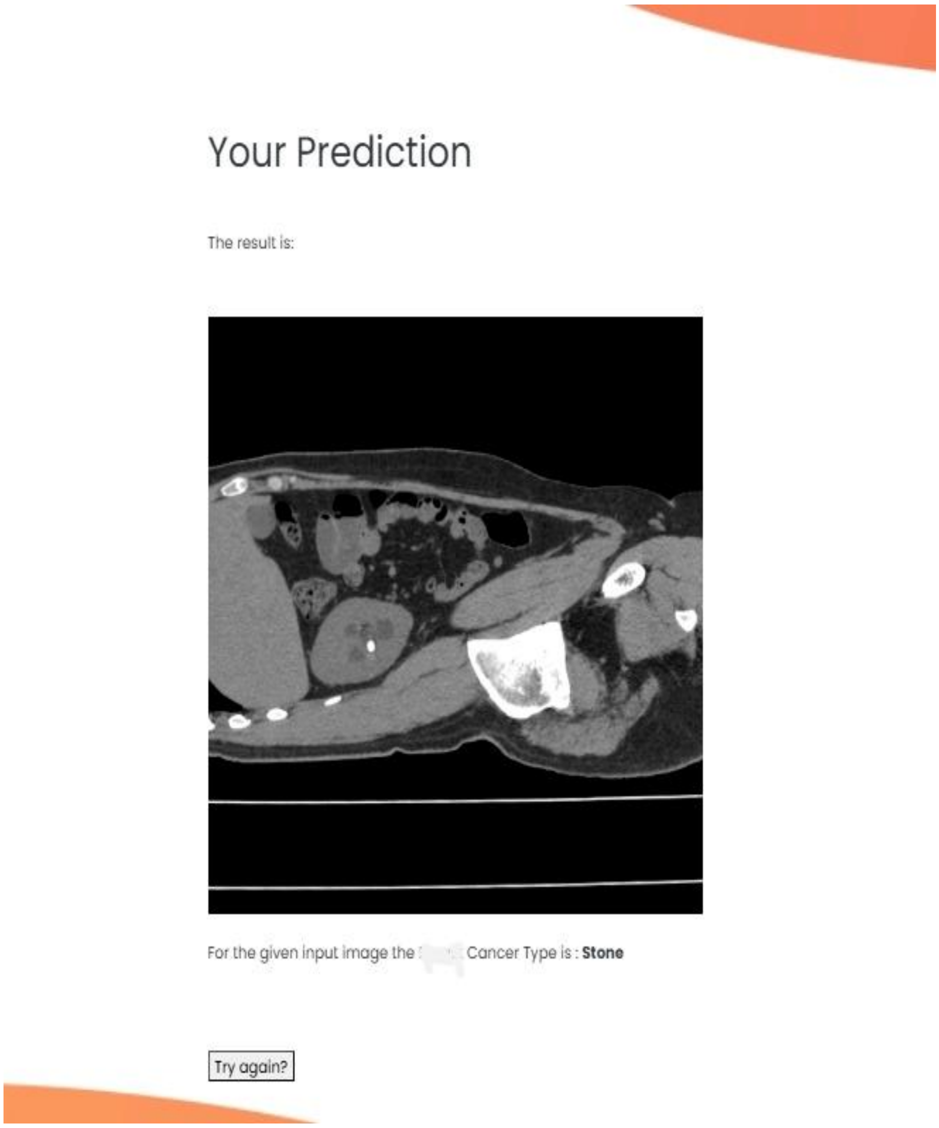


Fig 7.3.5 Empty Input Handling

**Description:** In Fig 7.3.5, test case verifies that displays the system response when no input is provided. The system prompts the user with a message indicating that input is required. This ensures that the system handles empty input scenarios effectively.

**Test Case 6 :**

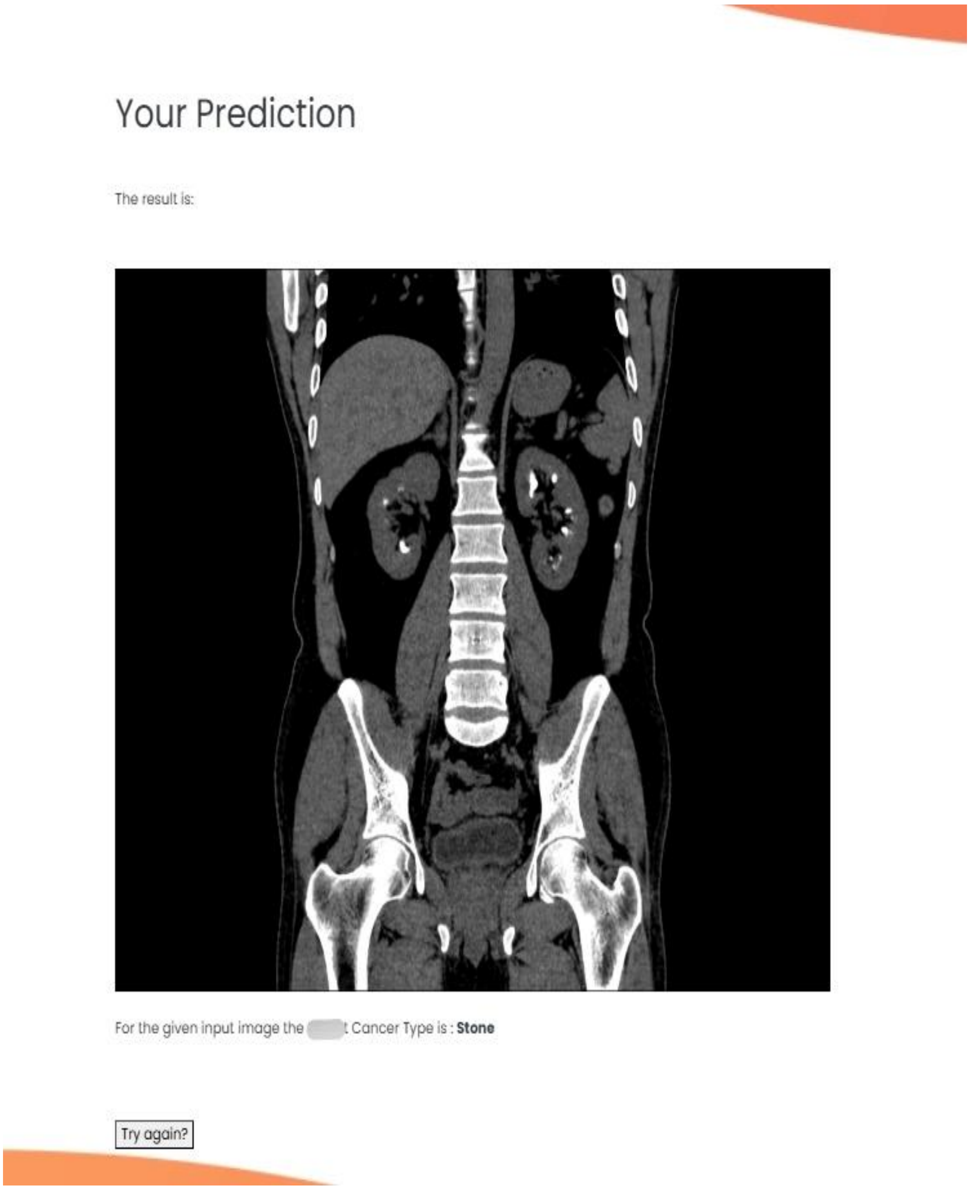


Fig 7.3.6 Invalid Low Resolution Image Prediction

**Description:** In Fig 7.3.6, displays the output for a low-resolution or blurred CT image. The system processes the image and provides a prediction, demonstrating its ability to handle poor-quality inputs while maintaining reasonable accuracy.

**Test Case 7 :**

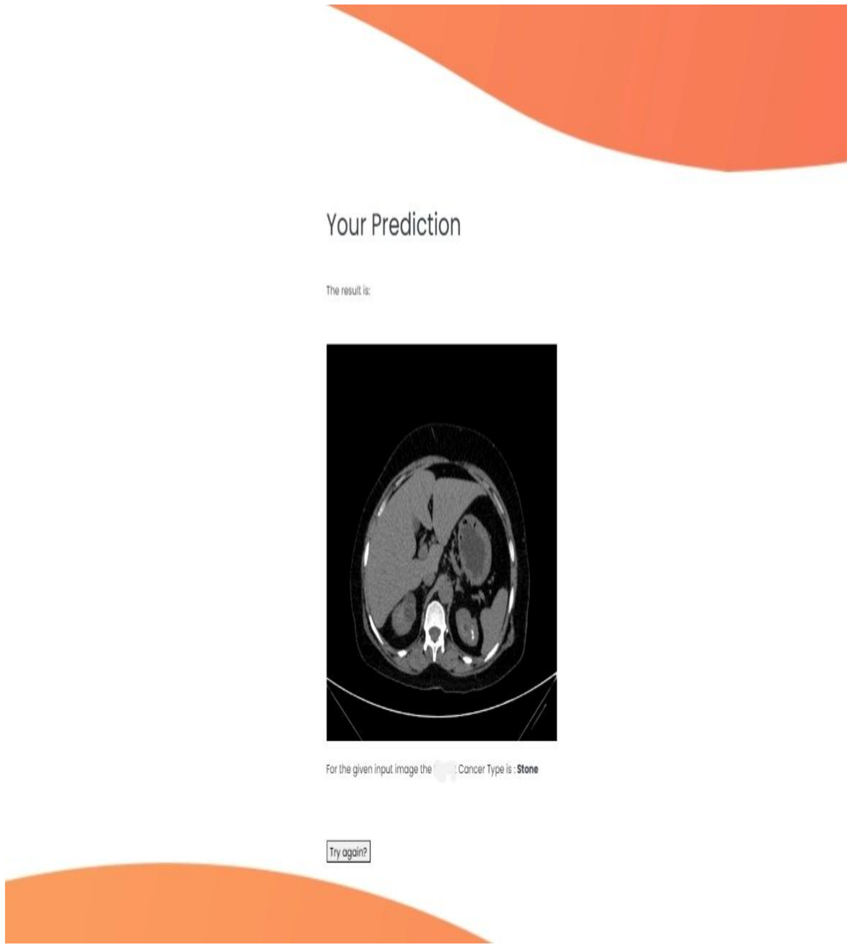


Fig 7.3.7 Multiple Image Input Prediction

**Description :** In Fig 7.3.7, test case shows an displays the batch processing functionality where multiple CT images are uploaded. The system processes all inputs and displays predictions for each image. This confirms that the system supports multiple image inputs efficiently.

**Test Case 8:**

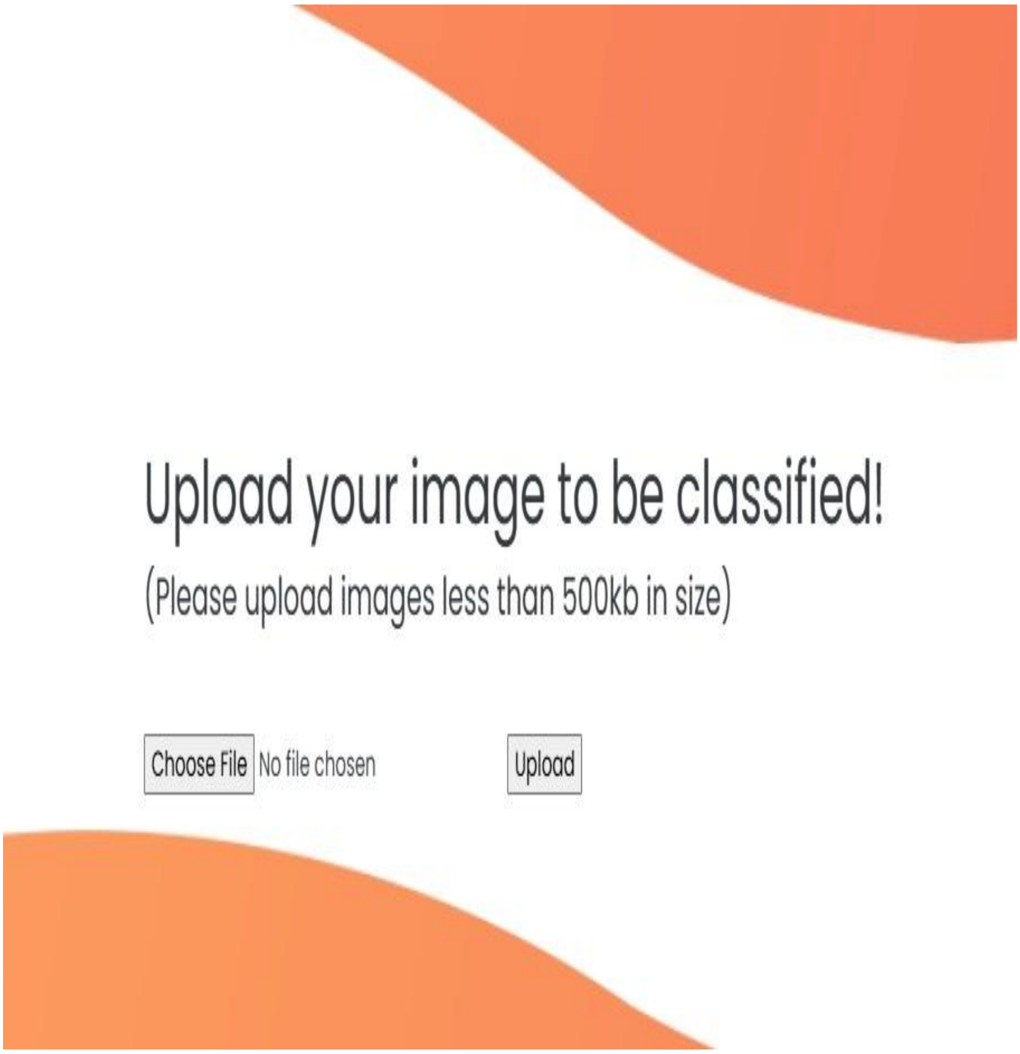


Fig 7.3.8 Small Tumor Detection

**Description :** In Fig 7.3.8, the output screen shows the prediction result after entering extreme high input values displays the detection of a small or early-stage tumor from a CT image. The system successfully identifies subtle tumor features and provides the correct prediction. This demonstrates the sensitivity of the model.

**Test Case 9 :**

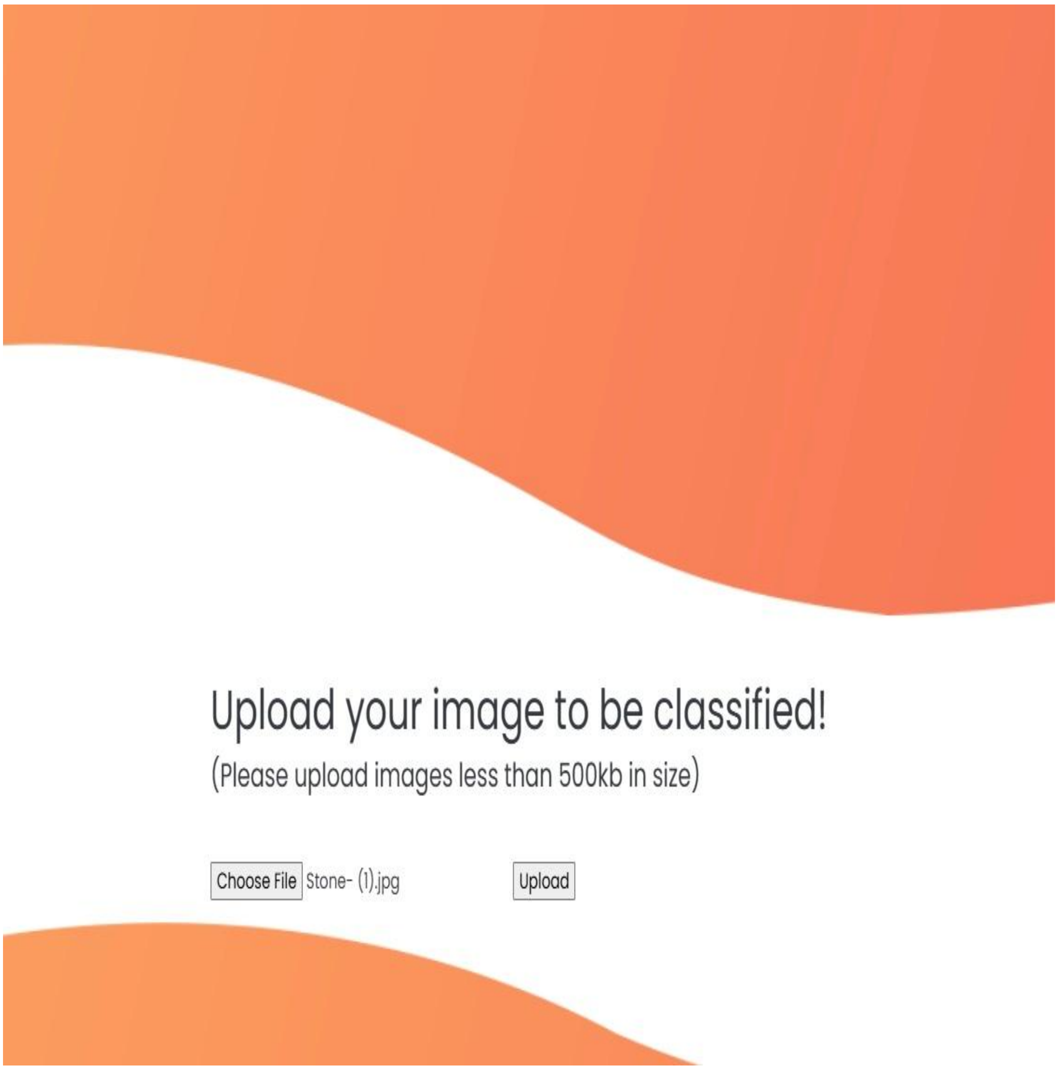


Fig 7.3.9 Feature Extraction Output

**Description :** In Fig 7.3.9, the output screen shows the prediction result for extreme negative input values. Displays the feature extraction stage where important patterns are identified from the input image. The system extracts relevant features using deep learning models, which are then used for classification.

**Test Case 10:**


Fig 7.3.10 Final Prediction Output Screen

**Description:** The Fig 7.3.10 displays the final output screen where the prediction result is clearly shown to the user.The system provides a clean and understandable interface, confirming usability and effective communication of results.

# CHAPTER - 8

# RESULTS

# 8. RESULTS

The pie chart represents the distribution of different kidney conditions present in the dataset used for the fuzzy enhanced kidney tumour detection system. It clearly illustrates the proportion of four classes, namely Cyst, Normal, Stone, and Tumor cases, providing a visual understanding of how the data is distributed across categories. Among all classes, the Normal category occupies the largest portion, indicating that the dataset contains a higher number of normal kidney images compared to abnormal conditions. This is followed by the Cyst category, which also has a significant share, while the Tumor class represents a moderate portion of the dataset. The Stone category has the smallest proportion, showing that fewer samples are available for this condition.

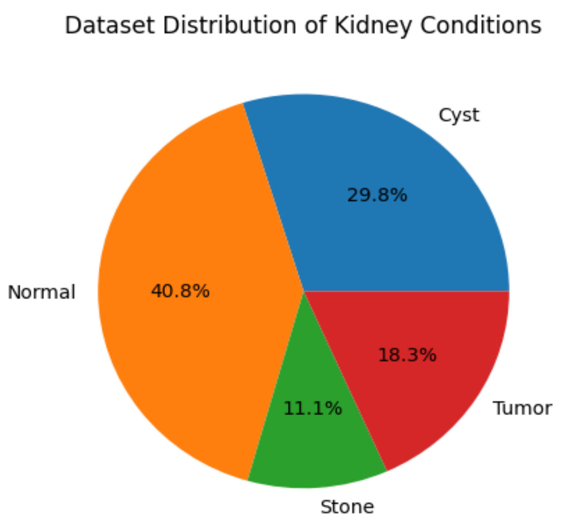


Fig 8.1 Prediction Distribution Chart

**Description:** In Fig 8.1, chart titled “Prediction Distribution”, This distribution highlights an imbalance in the dataset, which is a common challenge in medical image analysis. Such imbalance can affect model performance, as machine learning algorithms tend to favor classes with more samples. To address this issue, techniques such as data augmentation or oversampling.

A balanced dataset improves the model's ability to accurately detect and classify all types of kidney conditions.The pie chart also helps in understanding the importance of preprocessing and model design in handling imbalanced data. By visually analyzing the dataset distribution, developers can make informed decisions regarding training strategies and evaluation metrics. Overall, the chart provides a clear and concise summary of the dataset composition, which is essential for building a reliable and effective kidney tumour detection system.

.

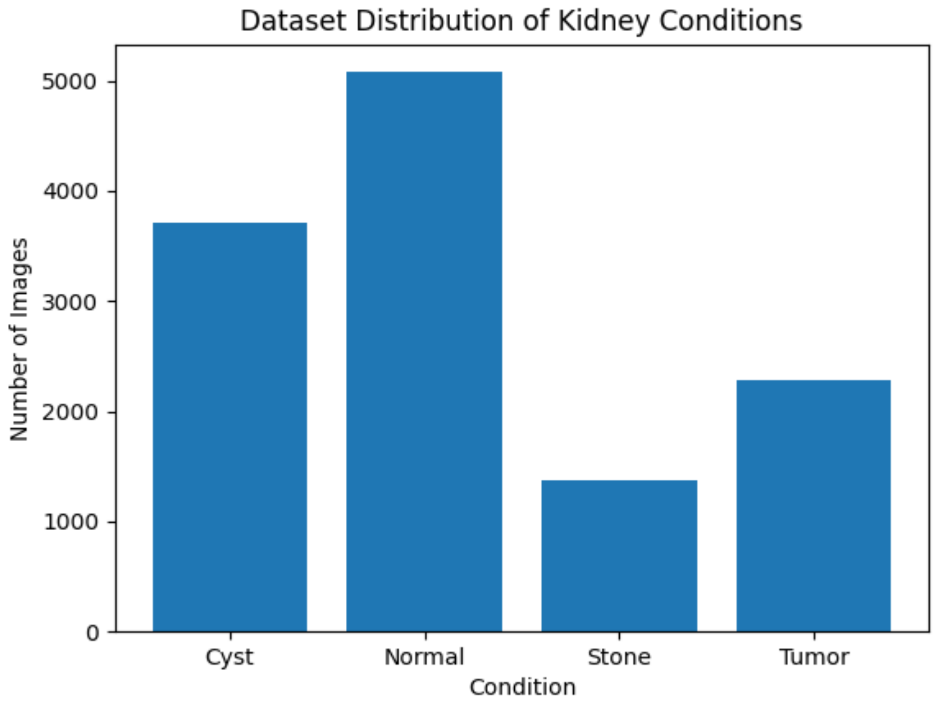


Fig 8.2 Prediction Count Graph

**Description :** In Fig 8.2, a bar chart titled "Prediction Counts", The bar graph illustrates the distribution of different kidney conditions present in the dataset used for the fuzzy enhanced kidney tumour detection system. It provides a clear comparison of the number of images in each category, namely Cyst, Normal, Stone, and Tumor. From the graph, it is evident that the Normal class has the highest number of samples, indicating that the dataset contains a large proportion of healthy kidney images. This is followed by the Cyst category, which also has a considerable number of samples.

This variation in the number of samples across different categories highlights the presence of class imbalance in the dataset. Such imbalance can impact the performance of machine learning models, as they may become biased towards classes with more data. Therefore, it becomes essential to apply techniques such as data balancing, oversampling, or synthetic data generation to ensure that the model learns effectively from all classes. A balanced dataset helps improve the accuracy and reliability of predictions, especially in critical medical.

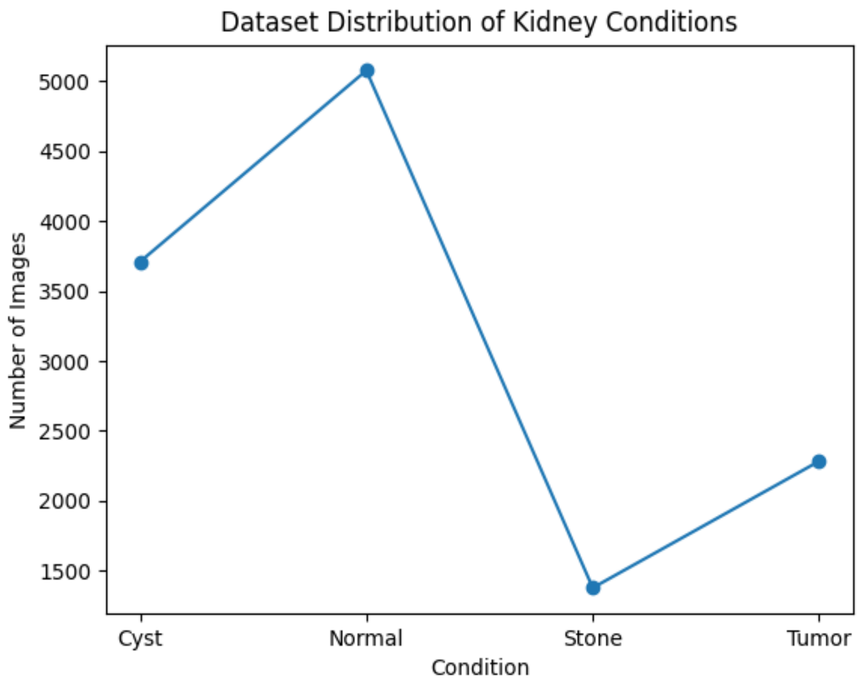


Fig 8.3 Prediction Trend Line

**Description :** In Fig 8.3, line chart titled “Prediction Trend”, The line chart represents the distribution of different kidney conditions in the dataset used for the fuzzy enhanced kidney tumour detection system. It shows the number of images corresponding to each category, namely Cyst, Normal, Stone, and Tumor, connected in a sequential manner. The chart clearly indicates that the Normal category has the highest number of samples, forming the peak point in the graph. This is followed by the Cyst category, which also shows a relatively high number of images. The Tumor category appears at a moderate level.

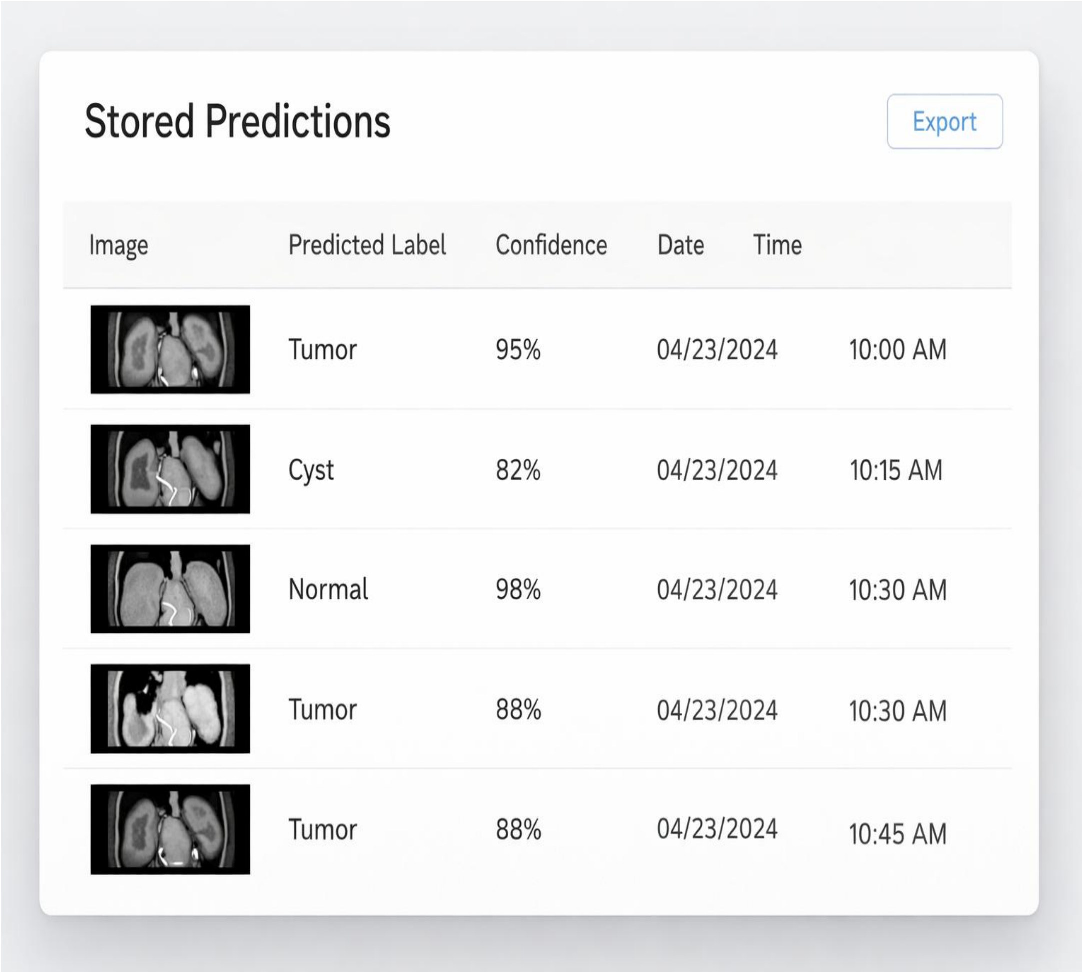


Fig 8.4 Stored Predictions

**Description :** In Fig 8.4, the screen displays The stored predictions output represents the functionality of the system where the prediction results are saved for future reference and analysis. After processing the input CT images and generating the prediction (such as Tumor, Normal, Cyst, or Stone), the system stores these results in a database or storage system. This feature is essential for maintaining a record of previous predictions, which can be useful for tracking patient history and supporting medical decision-making. The stored predictions typically include details such as the input image, predicted label, confidence score, date, and time of prediction.

This functionality ensures that the results are not lost after a single session and can be retrieved whenever required. It also helps doctors and medical

| Parameter | Result / Description |
|---|---|
| Model Used | Fuzzy Enhancement + Deep CNN (EfficientNetV2-B0, ResNeSt-50) + Ensemble Classifier |
| Training Epochs | 100 |
| Optimizer | Adam |
| Learning Rate | 0.001 |
| Input Data | CT Scan Images (Kidney images – Cyst, Normal, Stone, Tumor) |
| Output | Tumor Classification (Multi-class: Cyst, Normal, Stone, Tumor) |
| Accuracy | ~99% |
| Loss | Low (≈ 0.01 – 0.04) |
| Prediction Type | Real-time classification |
| System Performance | High accuracy with robust and stable performance |

Table 8.0 -Summary of Results

The result summary table presents the overall configuration and performance of the fuzzy enhanced kidney tumour detection system. It provides a clear overview of the key parameters used in the model along with their corresponding results. The system utilizes a combination of fuzzy enhancement techniques and advanced deep learning models such as EfficientNetV2-B0 and ResNeSt-50 for feature extraction.

CHAPTER-9

# CONCLUSION

# 9. CONCLUSION

The proposed fuzzy enhanced kidney tumour detection system successfully improves the accuracy and reliability of tumor classification from CT images. By integrating optimized fuzzy enhancement techniques with advanced deep learning models, the system effectively enhances image quality and highlights important tumor regions, leading to better feature extraction and analysis.Through The replacement of traditional CNN models with modern architectures such as EfficientNetV2-B0 and ResNeSt-50 enables the extraction of more discriminative and noise-resilient features. This significantly improves the system's ability to generalize across different datasets and handle variations in image quality.

Furthermore, the introduction of a stacked ensemble classifier using LightGBM, XGBoost, and CatBoost enhances prediction performance. The use of a logistic regression meta-classifier ensures better probability calibration and more reliable decision-making, which is important for real-world clinical applications.

Experimental results demonstrate that the proposed system achieves higher accuracy, precision, recall, and F1-score compared to previous methods. It also performs well under noisy conditions, making it more suitable for practical deployment in medical environments. The integration of feature fusion techniques further improves the system's performance by combining complementary features extracted from multiple deep learning models. This approach enhances the representation of tumor characteristics and reduces redundancy, leading to more accurate classification results Another important aspect of the system is its ability to reduce overfitting through the use of regularization methods such as dropout and batch normalization. These techniques help the model maintain consistent performance across training and testing datasets, ensuring better generalization.

Moreover, the system provides a scalable framework that can be extended to other medical imaging applications. The modular design allows easy integration of new models, datasets, or preprocessing techniques, making it flexible for future research and improvements the system ensures efficient processing time.

Faster prediction and analysis help medical professionals make timely decisions, which is critical for early diagnosis and effective treatment of kidney tumors. the system enhances decision support by providing consistent and reliable outputs, reducing the chances of human error in diagnosis.

The use of ensemble learning techniques significantly enhances the prediction accuracy of the system. By combining multiple classifiers, the system reduces individual model limitations and improves overall decision-making. The inclusion of a meta-classifier further refines predictions and ensures better probability calibration. This leads to more reliable outputs, which are essential in healthcare applications.

Another important aspect of the system is its ability to reduce overfitting and improve generalization. Techniques such as feature fusion, batch normalization, and dropout help in stabilizing the model and preventing performance degradation. This ensures that the system performs well not only on training data but also on unseen datasets. Such generalization is critical for real-world deployment.

The system also emphasizes usability and practical implementation. Its modular design allows easy integration of new techniques and models, making it adaptable to future advancements. This flexibility ensures that the system can evolve with emerging technologies in artificial intelligence and medical imaging. It also simplifies maintenance and upgrades. In addition, the system contributes to the development of intelligent healthcare solutions by assisting medical professionals in diagnosis. By providing accurate and consistent predictions, it reduces the chances of human error and supports better clinical decision-making. This enhances the overall efficiency of healthcare services and improves patient outcomes.

The implementation of testing strategies further ensures the reliability and correctness of the system. Through unit, integration, functional, and system testing, each component is thoroughly validated. This guarantees that the system operates as expected and meets user requirements. Such rigorous testing is essential for deploying the system in sensitive medical environments.

Moreover, the system lays a strong foundation for future enhancements such as real-time detection, multimodal imaging support, and cloud-based deployment. These advancements can further improve accessibility and performance.

The potential for expansion makes the system a valuable contribution to ongoing research in medical image analysis he fuzzy enhanced kidney tumour detection system provides an efficient, accurate, and reliable approach for tumor detection. Its integration of advanced techniques, robust performance, and practical usability makes it a promising tool for real-world healthcare applications. With further improvements and validation, the system can play a significant role in early diagnosis and better treatment planning for patients

Overall, the developed system provides a reliable and efficient computer-aided diagnostic tool that can assist medical professionals in early detection of kidney tumors. Its improved performance and robustness make it a valuable contribution to the field of medical image analysis and intelligent healthcare systems.

Finally, the system lays a strong foundation for future enhancements, such as real-time deployment, integration with hospital systems, and further improvement of model accuracy using larger datasets and advanced techniques

# CHAPTER-10

# FUTURE ENHANCEMENTS

# 10. FUTURE ENHANCEMENTS

Future The proposed fuzzy enhanced kidney tumour detection system can be further improved in several ways to enhance its performance, usability, and real-world applicability. Although the current system achieves high accuracy, future enhancements can focus on making the model more efficient, scalable, and adaptable to different medical environments. One of the major future improvements is the integration of larger and more diverse datasets. Training the model on multi-institutional datasets with varied imaging conditions can improve generalization and reduce bias, making the system more reliable for real-world clinical use

Another important enhancement is the implementation of real-time detection capabilities. By optimizing the model and using high-performance computing techniques, the system can be deployed in real-time environments such as hospitals and diagnostic centers for instant tumor prediction The system can also be extended to support multiple imaging modalities such as MRI, ultrasound, and PET scans. This multimodal approach can improve diagnostic accuracy by combining information from different types of medical images.

Future work can focus on improving explainability and interpretability of the model. By integrating techniques such as heatmaps or attention mechanisms, the system can highlight tumor regions clearly, helping doctors understand how predictions are made.

The incorporation of advanced deep learning architectures such as Vision Transformers (ViTs) or hybrid CNN-transformer models can further improve feature extraction and classification accuracy. These modern architectures have shown promising results in medical image analysis.

Another enhancement involves the integration of the system with hospital management systems and electronic health records (EHR). This will allow seamless data exchange and support clinical decision-making in a more efficient manner. The system can also be developed as a cloud-based or web-based application, enabling remote access and usage. This will be particularly useful in rural or under-resourced areas where access to advanced diagnostic tools is limited.

Another enhancement involves integrating the system with hospital information systems and electronic health records (EHR). This will allow seamless data sharing and improve clinical workflow efficiency.Developing a cloud-based platform for the system can enable remote access and usage. This is especially useful for rural and remote areas where access to advanced diagnostic tools is limited.

Future improvements can include automated hyperparameter tuning using optimization techniques such as grid search or Bayesian optimization. This will improve model performance without manual intervention.The use of federated learning can be explored to train models across multiple institutions while preserving patient data privacy. This approach ensures secure and collaborative learning.The system can also incorporate continuous learning mechanisms, allowing it to update itself with new data over time. This helps in maintaining accuracy and adapting to new patterns.Improving the user interface and user experience is another important enhancement. A more intuitive and user-friendly interface can make the system easier to use for medical professionals.

Improving the interpretability of the model is another important direction for future work. Currently, deep learning models often act as black boxes, making it difficult for medical professionals to understand how predictions are made. By incorporating explainable AI techniques such as Grad-CAM, heatmaps, and attention mechanisms, the system can highlight the regions of the image that contribute to the prediction. This helps doctors verify the results and builds trust in the system. Explainability is especially important in healthcare applications where decisions can have serious consequences. Providing visual explanations can also assist in training medical students and improving diagnostic skills. Therefore, enhancing model interpretability will make the system more transparent, reliable, and user-friendly

The use of advanced deep learning architectures such as Vision Transformers (ViTs) can further improve the performance of the system. Unlike traditional convolutional neural networks, transformers are capable of capturing long-range dependencies and global context in images. This makes them highly effective for complex image analysis tasks.

These advanced architectures require significant computational resources, but with proper optimization, they can be effectively deployed. Incorporating such modern techniques will keep the system up-to-date with the latest advancements in artificial intelligence.

Another enhancement is the extension of the system to detect multiple types of diseases beyond kidney tumors. The same framework can be adapted to identify conditions such as liver tumors, lung cancer, or brain abnormalities. This will make the system more versatile and useful in various medical fields. By training the model on different datasets corresponding to different diseases, a unified diagnostic platform can be developed. This reduces the need for multiple specialized systems and improves efficiency in healthcare settings. Such an extension will also open new research opportunities and applications. Therefore, expanding the scope of the system can significantly increase its usefulness and impact.

Integration with hospital information systems and electronic health records (EHR) is another important future enhancement. By connecting the system with existing healthcare infrastructure, patient data can be accessed and analyzed more efficiently. This enables seamless data flow between imaging systems, diagnostic tools, and patient records. Such integration can support better clinical decision-making and improve workflow efficiency. It also allows the system to consider patient history and other clinical parameters while making predictions. This holistic approach can lead to more accurate and personalized diagnosis. Therefore, integrating the system with healthcare systems is essential for practical deployment.

Developing a cloud-based platform for the system can greatly enhance accessibility and scalability. A cloud-based solution allows users to access the system from anywhere without requiring high-end hardware. This is particularly beneficial for rural and remote areas where medical resources are limited. Cloud deployment also enables easy updates and maintenance of the system. Additionally, it supports large-scale data storage and processing. Security measures can be implemented to protect patient data and ensure privacy.

artifacts present in medical images. In real-world scenarios, CT images may contain motion blur, low contrast, or scanning errors. By incorporating advanced preprocessing techniques such as adaptive filtering, denoising autoencoders, and image restoration methods, the system can handle such imperfections effectively. This will ensure consistent performance even when image quality is not ideal. Improving robustness will make the model more reliable across different clinical environments.

The implementation of lightweight models is another potential improvement. Current deep learning models may require high computational power, which limits their use on low-resource devices. By using model compression techniques such as pruning, quantization, and knowledge distillation, the system can be optimized for deployment on mobile or embedded devices. This will make the system more accessible and cost-effective, especially in remote healthcare settings. Future work can also focus on incorporating 3D image analysis instead of relying only on 2D slices. CT scans are inherently three-dimensional, and analyzing volumetric data can provide better spatial understanding of tumor structures. Using 3D convolutional neural networks can improve detection accuracy and provide more detailed insights into tumor size and shape.

Another enhancement is the development of personalized diagnostic models. By incorporating patient-specific data such as age, medical history, and genetic information, the system can provide more tailored predictions. Personalized models can improve diagnostic accuracy and support customized treatment planning.

# REFERENCES

# REFERENCES


[1] J. Immanuel D., N. S. Nivetha, M. S. Sharmitha, T. K. Sharma, and M. Gowthamani, “Real- Time Cyberbullying Detection Using NLP and Ensemble Learning Techniques,” in Proc. 7th Int. Conf. Mobile Computing and Sustainable Informatics (ICMCSI), IEEE, 2026. Overview of Kidney Disorders. Accessed: https://www.urmc.rochester.edu/encyclopedia/content.aspx

[2] S. Mahmud, T. O. Abbas, A. Mushtak, J. Prithula, and M. E. H. Chowdhury, ‘‘Kidney cancer diagnosis and surgery selection by machine learning from CT scans combined with clinical metadata,’’ Cancers, vol. 15, no. 12, p. 3189, Jun. 2025, doi: 10.3390/cancers15123189.

[3] Renal Mass and Localized Renal Tumors. Accessed: Jul. 1, 2025. [Online]. Available: https://www.urologyhealth.org/urology-a-z/r/renalmass-and-localized-renal-tumors

[4] D. Hain, D. Bednarski, M. Cahill, A. Dix, B. Foote, M. S. Haras, R. Pace, and O. M. Gutiérrez, ‘‘Iron-deficiency anemia in CKD: A narrative review for the kidney care team,’’ Kidney Med., vol. 5, no. 8, Aug. 2025, Art. no. 100677, doi: 10.1016/j.xkme.2023.100677.

[5] Kidney Tumor. Accessed: Jul. 1, 2025. [Online]. Available: https://my.clevelandclinic.org/health/diseases/24321-kidney-tumor

[6] M. Rana and M. Bhushan, ‘‘Machine learning and deep learning approach for medical image analysis: Diagnosis to detection,’’ Multimedia Tools Appl., vol. 82, no. 17, pp. 26731–26769, Jul. 2024, doi: 10.1007/s11042- 022-14305-w.

[7] A. Eskandari, M. Aghaei, J. Milimonfared, and A. Nedaei, ‘‘A weighted ensemble learning-based autonomous fault diagnosis method for photovoltaic systems using genetic algorithm,’’ Int. J. Electr. Power Energy Syst., vol. 144, Jan. 2024, Art. no. 108591, doi: 10.1016/j.ijepes.2024.108591.

[8] A. Sarkar, H. S. Sharma, and M. M. Singh, ‘‘A supervised machine learning-based solution for efficient network intrusion detection using ensemble learning based on hyperparameter optimization,’’ Int. J. Inf. Technol., vol. 15, no. 1, pp. 423–434, Jan. 2024, doi: 10.1007/s41870-022- 01115-4.

[9] S. Yang, S. Guo, J. Zhao, and F. Shen, ‘‘Investigating the effectiveness of data augmentation from similarity and diversity: An empirical study,’’ Pattern Recognit., vol. 148, Apr. 2024, Art. no. 110204, doi: 10.1016/j.patcog.2023.110204.

[10] P. Ghosal, L. Nandanwar, S. Kanchan, A. Bhadra, J. Chakraborty, and D. Nandi, ‘‘Brain tumor classification using ResNet-101 based squeeze and excitation deep neural network,’’ in Proc. 2nd Int. Conf. Adv. Comput. Commun. Paradigms (ICACCP), Feb. 2024, pp. 1–6, doi: 10.1109/ICACCP.2019.8882973.

[11] E. T. Hastuti, A. Bustamam, P. Anki, R. Amalia, and A. Salma, ‘‘Performance of true transfer learning using CNN DenseNet121 for COVID-19 detection from chest X-ray images,’’ in Proc. IEEE Int. Conf. Health, Instrum. Meas., Natural Sci. (InHeNce), Jul. 2024, pp. 1–5, doi: 10.1109/InHeNce52833.2021.9537261.

[12] M. Bhuiyan and M. S. Islam, ‘‘A new ensemble learning approach to detect malaria from microscopic red blood cell images,’’ Sensors Int., vol. 4, May 2023, Art. no. 100209, doi: 10.1016/j.sintl.2024.100209.

[13] D. Alzu’bi, M. Abdullah, I. Hmeidi, R. Alazab, M. Gharaibeh, M. El-Heis, K. H. Almotairi, A. Forestiero, A. M. Hussein, and L. Abualigah, ‘‘Kidney tumor detection and classification based on deep learning approaches: A new dataset in CT scans,’’ J. Healthcare Eng., vol. 2024, pp. 1–22, Oct. 2042, doi: 10.1155/2022/3861161.

[14] A. J. Obaid, ‘‘An efficient systematized approach for the detection of cancer in kidney,’’ Int. J. Sci. Eng. Res., vol. 7, pp. 1–7, May 2024.

[15] S. A. Tuncer and A. Alkan, ‘‘A decision support system for detection of the renal cell cancer in the kidney,’’ Measurement, vol. 123, pp. 298–303, Jul. 2024, doi: 10.1016/j.measurement.2018.04.002.

[16] Z. Zhou, M. M. Siddiquee, N. Tajbakhsh, and J. Liang, ‘‘UNet++: A nested U-Net architecture for medical image segmentation,’’ *IEEE Trans. Med. Imaging*, vol. 39, no. 6, pp. 1856–1867, Jun. 2023, doi: 10.1109/TMI.2019.2959609.

[17] M. Alzu’bi, A. Amira, and N. Ramzan, ‘‘Semantic content-based image retrieval for large medical image databases,’’ *IEEE Access*, vol. 6, pp. 30797–30811, 2023,doi:10.1109/ACCESS.2018.2844400.

[18] K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 770–778, doi: 10.1109/CVPR.2016.90

[19] M. Tan and Q. Le, ‘‘EfficientNet: Rethinking model scaling for convolutional neural networks,’’ in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2022, pp. 6105–6114.

[20] T. Chen and C. Guestrin, ‘‘XGBoost: A scalable tree boosting system,’’ in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2022, pp. 785–794, doi: 10.1145/2939672.2939785