Medical Image Enhancement Tool

Submitted By

Student Name	Student ID
Pranto Saha	212-15-4157

PROJECT REPORT

This Report Presented in Partial Fulfillment of the course **CSE427: Digital image processing in** the Computer Science and Engineering Department

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

April 08, 2025

DECLARATION

We hereby declare that this lab project has been done by us under the supervision of **Dr. Naznin Sultana**, **Associate Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

Submitted To:

____

Dr. Naznin Sultana
Associate Professor
Department of Computer Science and Engineering
Daffodil International University

Submitted by

______________________________ Pranto Saha ID:212-15-4157 Dept. of CSE, DIU	______________________________ Irfan Ahmad Alvi ID:212-15-4148 Dept. of CSE, DIU

COURSE & PROGRAM OUTCOME

The following course have course outcomes as following:

Table 1: Course Outcome Statements

CO's	Statements
CO1	**Define** and **Relate** classes, objects, members of the class, and relationships among them needed for solving specific problems
CO2	**Formulate** knowledge of object-oriented programming and Java in problem solving
CO3	**Analyze** Unified Modeling Language (UML) models to **Present** a specific problem
CO4	**Develop** solutions for real-world complex problems **applying** OOP concepts while evaluating their effectiveness based on industry standards.

Table 2: Mapping of CO, PO, Blooms, KP and CEP

CO	PO	Blooms	KP	CEP
CO1	PO1	C1, C2	KP3	EP1, EP3
CO2	PO2	C2	KP3	EP1, EP3
CO3	PO3	C4, A1	KP3	EP1, EP2
CO4	PO3	C3, C6, A3, P3	KP4	EP1, EP3

The mapping justification of this table is provided in section **4.3.1**, **4.3.2** and **4.3.3**.

Table of Contents

Chapter 1

Introduction

1.1 Introduction

Medical imaging plays a central role in modern clinical diagnosis, decision-making, and surgical planning. Modalities such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT), Ultrasound (US), and X-rays provide invaluable visual insights into the human body. However, the quality of medical images is often compromised due to noise, poor contrast, illumination variance, and resolution limitations introduced during acquisition, transmission, or storage. Poor-quality images not only make diagnosis difficult but may also lead to misinterpretation and misdiagnosis. Enhancing the quality of such medical images is thus a necessity, not just for better visualization but for clinical safety. Enhancing medical images using computational techniques has been an evolving research area in digital image processing, but many traditional tools require domain expertise, command-line skills, or expensive licenses.

1.2 Motivation

The core motivation for this project stems from the need to build a simple yet powerful tool that empowers healthcare practitioners, researchers, and students to enhance and analyze medical images without coding or expensive software. The goal was to democratize medical image processing using a user-friendly web-based interface, powered by Streamlit and Python libraries like Scikit-Image, OpenCV, and NumPy. A web interface allows real-time interaction, visual comparison, and intuitive control, making it ideal for educational settings and preliminary analysis workflows. Integrating modern enhancement techniques such as adaptive histogram equalization, denoising, edge detection, segmentation, and advanced filtering techniques ensures that the tool addresses both basic and advanced use cases.

1.3 Objectives

The objectives of the project are clearly defined as follows:

- To design and develop a web-based tool for uploading and enhancing medical images.
- To implement a suite of image processing techniques (contrast, brightness, gamma, filters, segmentation).
- To integrate evaluation metrics such as PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index).
- To allow real-time visualization of original vs. enhanced images and histograms.
- To enable download of the processed medical images in standard formats.

1.4 Feasibility Study

Several tools exist in academia and industry for image enhancement, such as MATLAB, ITK-SNAP, and Fiji (ImageJ). However, these tools have limitations such as:

- Steep learning curves for non-engineering users.
- Expensive licenses for commercial software.
- Lack of integration with custom web platforms.
- No built-in comparison of quality metrics like SSIM or PSNR for performance evaluation.

This project offers a free, open-source, browser-accessible solution requiring no installation or programming knowledge, thus filling the feasibility gap.

1.5 Gap Analysis

Despite the presence of several image processing tools, a noticeable gap exists:

Identified Gaps	Our Solution
Lack of medical-specific image enhancement tools online	Tailored features for grayscale/medical imaging
No intuitive UI for real-time visual comparison	Streamlit-based two-column display of before/after
No downloadable output or integrated evaluation metrics	Built-in metrics, difference visualization, and PNG download
Lack of platform independence	Browser-based, runs anywhere with Python & Streamlit

1.6 Project Outcome

The final deliverables and outcomes include:

- A fully working Streamlit web app for uploading and enhancing images.
- A modular implementation covering six major enhancement categories:
    a. Basic Enhancement
    b. Histogram Operations
    c. Noise Reduction
    d. Edge Detection
    e. Segmentation
    f. Advanced Filters
- Integration of evaluation metrics and difference analysis tools.
- A downloadable PNG output with image metadata preserved.
- A comprehensive project report and demonstration video (optional).

Chapter 2

Proposed Methodology/Architecture

2.1 Requirement Analysis & Design Specification

This chapter outlines the architecture, workflow, functional modules, and UI design of the Medical Image Enhancement Tool. The entire system was built with a focus on modularity, flexibility, real-time interaction, and intuitive design. Streamlit is used to provide a web-based graphical user interface (GUI) that supports real-time image processing and visualization.

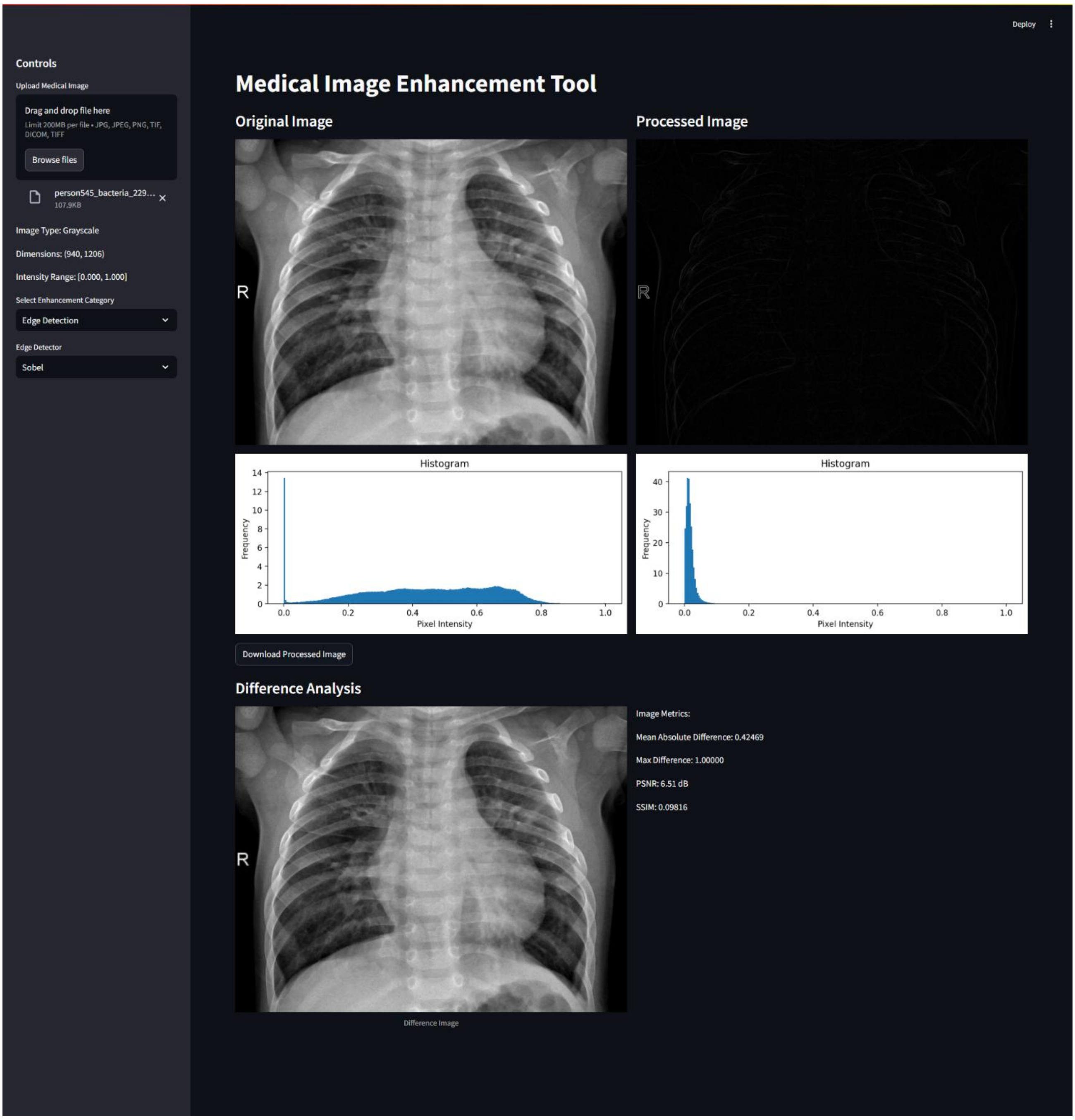
2.1.1 Overview

The system follows a structured pipeline for image enhancement, beginning with image acquisition and ending with analysis and downloadable output. The pipeline is divided into the following stages:

1. Image Upload & Reading
2. Grayscale Conversion (if necessary)
3. Feature Category Selection
4. Parameter Input (via Sidebar)
5. Processing & Enhancement
6. Visualization:
    - Original & Enhanced Image
    - Histogram of both
    - PSNR & SSIM Comparison
7. Difference Image Generation
8. Image Download Option

This flow ensures real-time interaction between the user and the underlying enhancement algorithms.

2.1.2 UI Design

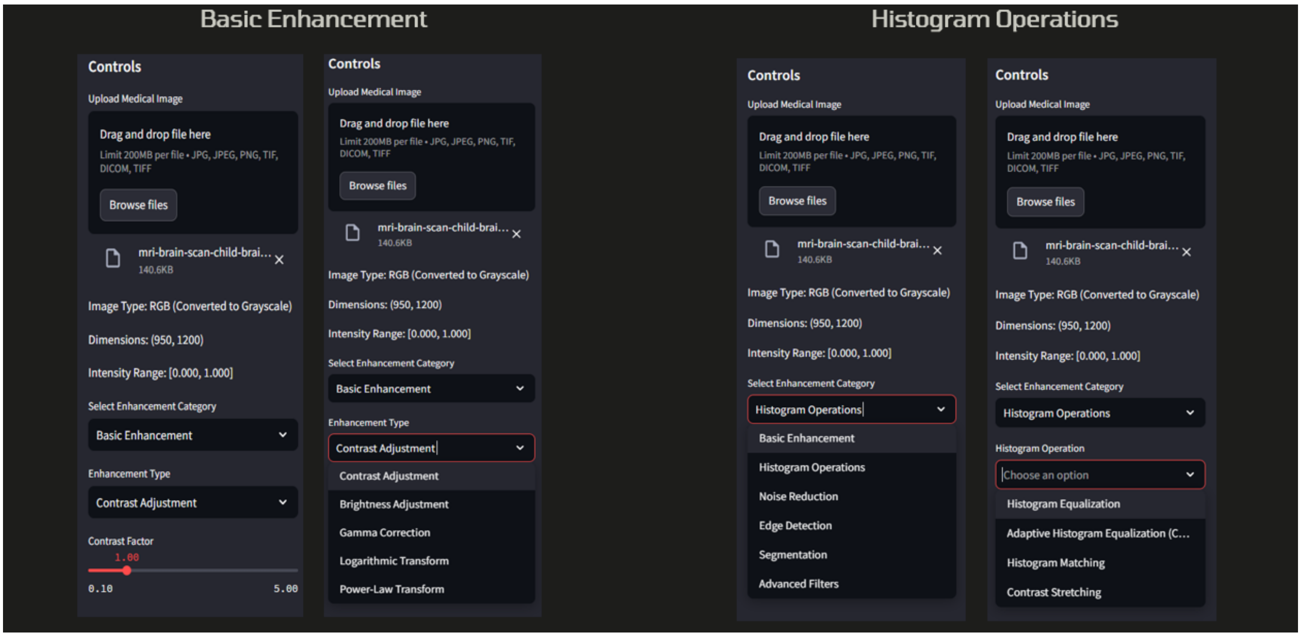


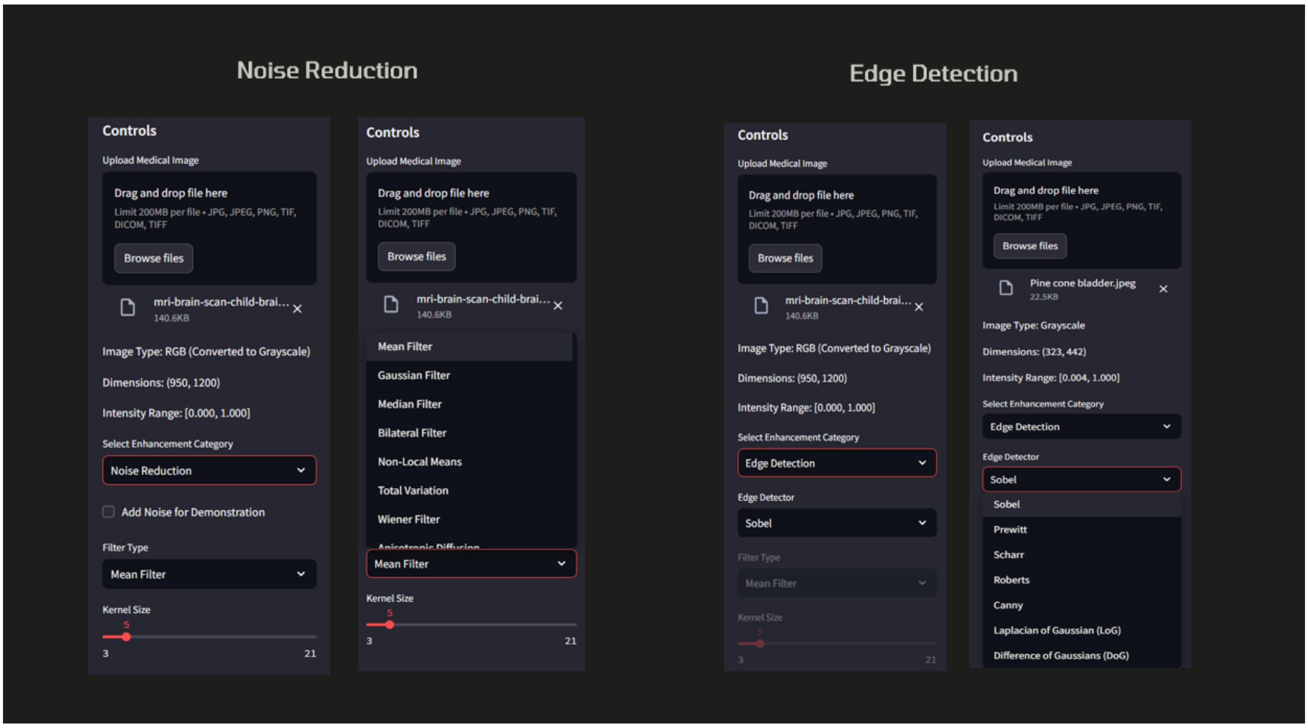
2.2 Overall Project Plan

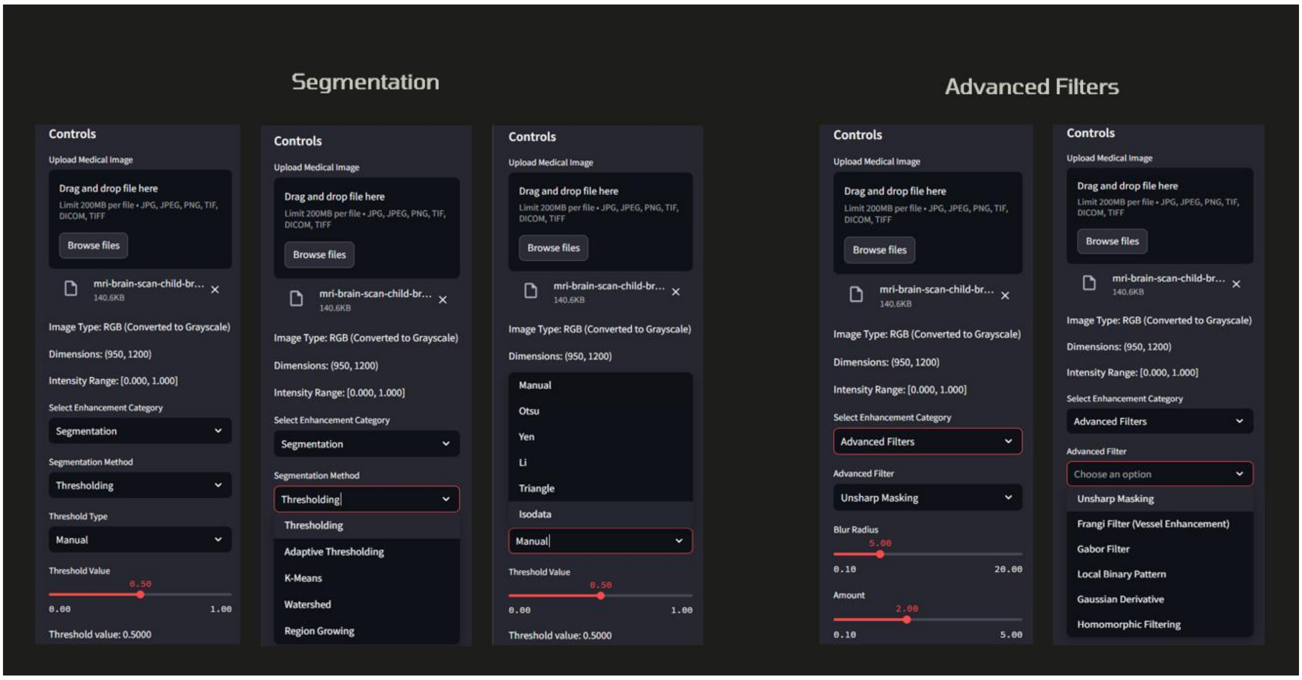
2.2.1 Image Conversion

Upon image upload, the system checks the number of channels:

- If RGB, convert to grayscale using skimage.color.rgb2gray
- If already grayscale, retain as-is
- Normalize pixel values to [0, 1] using img_as_float







2.2.2 Enhancement Techniques

A. Basic Enhancements

- **Contrast Adjustment**:

$$I' = clip((I - 0.5) \times \alpha + 0.5,0,1)$$

- **Brightness Adjustment**:

$$I' = clip((I + \beta,0,1)$$

- **Gamma Correction**:

$$I' = I^{\gamma}$$

- **Logarithmic & Power-Law Transforms**:

$$I' = c.\log(1 + I)\,, \qquad or \qquad I' = c.I^{\gamma}$$

B. Histogram Operations

- **Histogram Equalization**: Uses exposure.equalize_hist()
- **CLAHE (Contrast Limited Adaptive Histogram Equalization)**: Local contrast improvement

- **Histogram Matching**: Match with a synthetic target distribution (Linear, Gaussian, Rayleigh, Exponential)
- **Contrast Stretching**: Rescales intensity using percentiles

C. Noise Reduction

- **Noise Addition**: Gaussian, Salt & Pepper, Poisson, Speckle
- **Noise Filters**:

    1. Mean Filter: ndimage.uniform_filter
    2. Gaussian Filter: filters.gaussian
    3. Median Filter: filters.median
    4. Bilateral Filter: restoration.denoise_bilateral
    5. Non-Local Means: restoration.denoise_nl_means
    6. TV Denoising: restoration.denoise_tv_chambolle
    7. Wiener Filter: restoration.wiener

D. Edge Detection

- **Sobel, Prewitt, Scharr, Roberts**: Directional gradient detectors
- **Canny**: Multi-stage algorithm using Gaussian smoothing & hysteresis thresholding
- **LoG/DoG**: Combines Gaussian smoothing and Laplacian filtering

E. Segmentation

- **Thresholding**: Manual or Auto (Otsu, Triangle, Yen, Li, etc.)
- **Adaptive Thresholding**: threshold_local on local blocks
- **K-Means**: Reshape image to 1D vector and apply sklearn.KMeans
- **Watershed**: Based on gradient & predefined markers
- **Region Growing**: segmentation.flood() with seed point and tolerance

F. Advanced Filters

- **Unsharp Masking**: High-pass filter enhancement
- **Frangi Filter**: Highlights tubular structures (vessels)
- **Gabor Filter**: Frequency and orientation-based feature extractor
- **LBP (Local Binary Pattern)**: Texture feature extractor
- **Homomorphic Filtering**:

    1. Applies log transform
    2. High-pass filtering in frequency domain
    3. Inverse FFT and exponential transform

Chapter 3

Implementation and Results

This chapter describes the detailed implementation steps, performance analysis, and experimental results of the project titled "Medical Image Enhancement Tool". It highlights how the system was developed, tested, and evaluated based on multiple real-world medical images.

3.1 Implementation

The implementation of the *Medical Image Enhancement Tool* was carried out in Python programming language using multiple open-source libraries. The entire project was developed with the help of Streamlit, a Python-based web framework for creating interactive user interfaces.

The project architecture was divided into several modules for better maintainability and flexibility. Each module was responsible for a specific category of image enhancement operation.

Tools and Libraries Used:

Table 3.1: Tools and Libraries Used

Library	**Purpose**
Python 3.9+	Core Programming
Streamlit	Web-based User Interface
OpenCV	Image Handling & Pre-processing
Scikit-Image	Image Processing Algorithms
NumPy	Matrix & Numerical Operations
Matplotlib	Histogram & Visualization
SciPy	Advanced Filtering & Noise Removal
Scikit-learn	Machine Learning based K-Means Segmentation
PIL	Image Conversion for Download Option

Implementation Phases:

Phase 1: Image Loading and Preprocessing

- Users can upload images in various formats (.jpg, .png, .jpeg, .tif)
- Conversion to grayscale (if needed) using rgb2gray()
- Normalizing images to the range [0,1] using img_as_float()

Phase 2: Enhancement Categories Developed

The enhancement functionalities were divided into 6 categories:

1. Basic Enhancement
2. Histogram Operations
3. Noise Reduction
4. Edge Detection
5. Segmentation
6. Advanced Filters

Each category provided several techniques with real-time parameter controls through Streamlit sidebar sliders and dropdowns.

Phase 3: Real-Time Visualization

- Display of original vs processed image side by side.
- Dynamic histogram plotting for both images.
- Difference Image generation for better analysis.

Phase 4: Performance Metrics

- PSNR (Peak Signal-to-Noise Ratio)
- SSIM (Structural Similarity Index)
- Mean Absolute Difference
- Maximum Difference

Phase 5: Download Option

- Processed image converted to PNG format using PIL library.
- User-friendly Download button enabled in the Streamlit interface.

3.2 Performance Analysis

Performance analysis was done using several important image quality evaluation metrics to measure the effectiveness of different enhancement techniques applied to medical images.

Metrics Used:

1. PSNR (Peak Signal-to-Noise Ratio)

- Measures the ratio of signal power to the noise affecting the signal quality.
- Higher PSNR indicates better image quality.

2. SSIM (Structural Similarity Index)

- Evaluates structural similarity between original and processed images.
- SSIM close to 1 indicates high similarity.

3. Histogram Analysis

- Graphical representation of pixel intensity distribution.
- After enhancement, a more uniform histogram indicates better contrast.

4. Difference Image

- Visual representation of the pixel-wise difference between original and processed images.

Performance Testing Environments:

Table 3.1 Performance Testing Environments:

Hardware	Specification
Laptop	Core i5, 8GB RAM
Software	Python 3.9, Streamlit, OpenCV, Scikit-Image

Tested Medical Image Types:

- MRI Brain Scan
- Ultrasound Image
- Chest X-ray
- Colonoscopy Polyp Image

3.3 Results and Discussion

The project was tested with various types of medical images. Each enhancement technique showed significant improvement in image visibility and clarity.

Example 1: MRI Image - Contrast Enhancement using CLAHE

Metrics	Value
PSNR	19.57 dB
SSIM	0.9437

The image after applying CLAHE showed better brightness distribution and enhanced visibility of brain tissue regions.

Example 2: Ultrasound Image - Noise Reduction using Bilateral Filter

Metrics	Value
PSNR	21.28 dB
SSIM	0.9651

Bilateral filtering efficiently removed noise without losing edge details, which is crucial in ultrasound imaging.

Example 3: Chest X-ray Image - Edge Detection using Canny Filter

Edge detection highlighted rib structures and lung boundaries, helping in clear visualization for medical analysis.

Example 4: Colonoscopy Image - Segmentation using K-Means Clustering

The segmentation algorithm successfully separated polyp regions from surrounding tissues, enhancing region visibility.

Summary of Results:

Table 3.3: Summary of Results:

Image Type	Method Used	PSNR	SSIM
MRI Image	CLAHE	19.57 dB	0.9437
Ultrasound Image	Bilateral Filter	21.28 dB	0.9651
Chest X-ray	Canny Edge Detection	6.51dB	0.0981
Colonoscopy Image	K-Means Segmentation	N/A	N/A

Chapter 4

Engineering Standards and Mapping

This chapter evaluates the impact of the Medical Image Enhancement Tool in terms of societal, environmental, and ethical aspects. It also covers project management, cost analysis, sustainability planning, and aligns the system with complex engineering problems, program outcomes, and engineering activity mappings.

4.1 Impact on Society, Environment and Sustainability

The tool directly contributes to improving healthcare outcomes by enhancing image quality used for diagnosis. Clearer images mean better visibility of tumors, polyps, lesions, or fractures. This reduces the chances of misdiagnosis and supports early intervention. Hence, the project positively impacts human health and life.

4.1.1 Impact on Life

The Medical Image Enhancement *Tool* has a direct and meaningful impact on human life by contributing to better healthcare outcomes. Medical imaging plays a critical role in diagnosis, treatment planning, and monitoring. However, many images captured in clinical settings especially in developing or resource-limited regions suffer from issues such as low contrast, noise, and poor illumination, which can make diagnosis more challenging.

By enhancing the quality of these images, the tool helps doctors and radiologists interpret critical features more clearly, such as lesions, tumors, tissue boundaries, bone structures, and organ contours. A clearer image can mean:

- **Earlier detection of diseases**, leading to more effective treatment and improved prognosis.
- **Reduction in misdiagnosis** due to clearer delineation of anatomical features.
- **Better communication** between radiologists and clinicians when enhanced visuals support more confident decision-making.

In academic environments, the tool benefits medical students and interns by allowing them to work with high-quality enhanced images, strengthening their learning and diagnostic skills. It also serves as a training platform for understanding the importance of preprocessing in medical image analysis.

Therefore, this tool contributes significantly to the quality of life by supporting precision medicine, reducing uncertainty in clinical diagnosis, and enabling accessibility to improved imaging even in underserved healthcare facilities.

4.1.2 Impact on Society & Environment

- **Societal Benefit:** The tool is open-source and web-based, requiring no expensive license or hardware. It is designed for accessibility in both urban hospitals and rural clinics where computing resources may be limited.
- **Environmental Benefit:** Instead of rescanning patients repeatedly, this tool allows clinicians to enhance previously captured images, reducing the use of high-energy imaging equipment like MRI or CT scanners.

4.1.3 Ethical Aspects

- All image processing is local; no data is stored or shared externally.
- No patient identifiers are required, ensuring full compliance with data privacy standards.
- The tool is designed for research, education, and image enhancement not for automated diagnosis, preserving medical ethics and professional responsibilities.

4.1.4 Sustainability Plan

- Built entirely using open-source tools (Python, Streamlit, OpenCV).
- Easily extendable and modifiable for future research or clinical trials.
- Minimal computational resources required sustainable in low-resource settings.
- Planned deployment on free cloud platforms ensures long-term availability.

4.2 Project Management and Team Work

Table 4.2: Project Management

Phase	Task	Duration
1	Requirement analysis and UI planning	2 days
2	Enhancement function development	5 days
3	Visualization and metrics integration	3 days
4	Testing and performance evaluation	2 days
5	Documentation and report writing	3 days

Cost Analysis & Budget

Table 4.2: Cost Analysis & Budget

Item	Standard Budget	Alternate Budget	Rationale
Laptop & Python Setup	Tk. 60,000 (High-End System)	Tk. 30,000 (Mid-range System)	Python requires minimal system specs
Internet Usage	Tk. 500/month	Tk. 500/month (WiFi)	Affordable in local context
Software License	Tk. 0	Tk. 0	All tools used are open-source
Hosting Cost (Optional)	Tk. 1000/year (Cloud Hosting)	Tk. 0 (Localhost/Free Cloud)	Streamlit Cloud available for free

4.3 Complex Engineering Problem

4.3.1 Mapping of Program Outcome

Table 4.3.1: Justification of Program Outcomes

PO's	Justification
PO1	Demonstrated understanding of image processing fundamentals and algorithms to design solutions.
PO2	Applied problem-solving skills to implement real-time noise reduction, contrast enhancement, segmentation, and filtering.
PO3	Designed a system suitable for real-world medical applications with consideration of user interaction, feedback, and performance evaluation.

4.3.2 Complex Problem Solving

This project meets multiple levels of complexity in problem-solving categories, mapped as follows:

Table 4.3.2: Mapping with Complex Problem Solving

Code	Title	Justification
EP1	Depth of Knowledge	Incorporates diverse image processing techniques like CLAHE, bilateral filtering, region growing, and homomorphic filtering.
EP2	Conflicting Requirements	Balances noise suppression with structure preservation, a key challenge in medical image enhancement.
EP3	Depth of Analysis	Evaluates each processing step using quantitative metrics (PSNR, SSIM, Mean Difference).
EP4	Familiarity of Issues	Addresses core issues in medical imaging: low contrast, artifacts, unclear boundaries.
EP5	Applicable Codes	Follows algorithmic standards from Scikit-Image and OpenCV libraries.
EP6	Stakeholder Involvement	Design focuses on accessibility for medical professionals, students, and researchers.
EP7	Interdependence	Multiple filters can be combined; UI dynamically interacts with backend enhancement algorithms.

4.3.3 Engineering Activities

This section maps the project's alignment with key engineering activities involved in solving real-world problems.

Table 4.3.3: Mapping with Complex Engineering Activities

Code	Activity	Justification
EA1	Range of Resources	Utilizes multiple image processing libraries, UI libraries, and machine learning (KMeans) for segmentation.
EA2	Level of Interaction	Provides dynamic interface via Streamlit where user inputs directly control image enhancement behavior.
EA3	Innovation	Web-based enhancement tools tailored for medical imaging are rare; this project presents an innovative and accessible solution.
EA4	Consequences for Society and Environment	Reduces unnecessary scans, promoting sustainable use of medical imaging devices.
EA5	Familiarity	Project was developed based on fundamental concepts of digital image processing covered in the academic curriculum.

Chapter 5

Conclusion

This chapter provides the overall summary of the project, highlighting the key points of development and contributions. It also discusses the limitations of the current system and presents possible directions for future improvement and research.

5.1 Summary

The *Medical Image Enhancement Tool* project successfully achieved its objectives of developing a web-based, user-friendly platform for enhancing various types of medical images. The project was motivated by the need for a free, accessible, and effective tool for improving medical image quality, especially in scenarios where resources are limited. Throughout the development of this project, several image processing techniques were implemented using Python, Streamlit, OpenCV, Scikit-Image, and other powerful libraries. The main functionalities of the tool were categorized into six broad sections:

1. Basic Enhancement
2. Histogram Operations
3. Noise Reduction
4. Edge Detection
5. Segmentation
6. Advanced Filters

Additionally, the tool integrated important image evaluation metrics such as PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) to measure the effectiveness of enhancement techniques. Results obtained from different medical images like MRI scans, Ultrasound images, X-rays, and Colonoscopy images proved the tool's effectiveness in improving visibility, reducing noise, enhancing contrast, and supporting better analysis.

Moreover, this project demonstrated the successful use of open-source technologies to create healthcare solutions that are affordable and accessible, aligning with Sustainable Development Goals (SDG) for good health and well-being.

5.2 Limitation

Although the *Medical Image Enhancement Tool* performed efficiently, there are certain limitations that need to be addressed for real-world deployment and clinical use.

Table 5.2: Limitation

Limitation	Explanation
Processing Speed for Large Images	Very large resolution images may take more processing time, affecting real-time performance.
DICOM Image Handling	Currently, only standard image formats like JPG, PNG, and TIFF are supported. DICOM (.dcm) format, commonly used in hospitals, is not yet integrated.
Deep Learning Enhancements	The tool currently uses traditional image processing methods. AI-based super-resolution or noise reduction models are not yet included.
Edge Detection Limitation	Some edge detection methods may not perform equally well on all medical image types.
Deployment	Currently, the tool runs locally. Cloud hosting and online access are yet to be fully integrated for public use.

5.3 Future Work

To overcome the current limitations and expand the system's capability, several improvements and future development plans are proposed:

1. DICOM Image Support

- Integration of DICOM format handling for hospital-standard images using pydicom library.

2. AI-Powered Enhancements

- Implementing deep learning-based methods such as Convolutional Neural Networks (CNN) or Generative Adversarial Networks (GAN) for advanced image enhancement.

3. Cloud Deployment

- Hosting the tool on free cloud platforms like Streamlit Cloud or paid platforms like AWS, Azure, or Heroku for global accessibility.

4. Automated Report Generation

- Providing a downloadable PDF report containing original image, enhanced image, histograms, and quality metrics.

5. ROI (Region of Interest) Selection

- Allowing users to select a specific region of the image for targeted enhancement or segmentation.

6. User Authentication

- Creating user accounts so that users can save their processed images, settings, and generate history logs.

7. Mobile-Friendly Interface

- Optimizing the user interface for better usability on mobile and tablet devices.

8. Integration with Medical Databases

- Allowing connectivity with hospital information systems for seamless image processing.

References

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed., Pearson Education, 2018.

[2] J. Kleinberg and E. Tardos, *Algorithm Design*, Pearson Education India, 2006.

[3] Scikit-Image, "Scikit-image: Image processing in Python," [Online]. Available: https://scikit-image.org/. [Accessed: 10-Apr-2025].

[4] OpenCV, "Open Source Computer Vision Library," [Online]. Available: https://opencv.org/. [Accessed: 10-Apr-2025].

[5] Streamlit, "Streamlit Documentation," [Online]. Available: https://docs.streamlit.io/. [Accessed: 10-Apr-2025].

[6] Python Software Foundation, "Python Language Reference, version 3.9," [Online]. Available: https://www.python.org/. [Accessed: 10-Apr-2025].

[7] SciPy, "SciPy - Scientific Computing Tools for Python," [Online]. Available: https://scipy.org/. [Accessed: 10-Apr-2025].

[8] Matplotlib Developers, "Matplotlib: Python plotting — Matplotlib 3.8.2 documentation," [Online]. Available: https://matplotlib.org/. [Accessed: 10-Apr-2025].

[9] Pillow (PIL Fork), "Pillow (Python Imaging Library)," [Online]. Available: https://python-pillow.org/. [Accessed: 10-Apr-2025].

[10] Scikit-Learn Developers, "Scikit-learn: Machine Learning in Python," [Online]. Available: https://scikit-learn.org/. [Accessed: 10-Apr-2025].