Software Requirements Specification

For

SharpView: Low Light Image Enhancement using C++

7th October 2024

Prepared By

Specialization	SAP ID	Name
AIML Hons B2	500107864	Sagar Thapliyal
AIML Hons B3	500108516	Aviral Khanna
AIML Hons B2	500107623	Saksham Siwach
AIML Hons B1	500107135	Arshdeep Kaur

Cluster: Artificial Intelligence and Machine Learning
School Of Computer Science
University of Petroleum & Energy Studies,
Dehradun- 248007, Uttarakhand

Table of Content

REVISION HISTORY

Date	Change	Reason for Changes	Mentor Signature
28.08.2024	Changed the source of images to ExDark Dataset	Manually, collecting low-light images was tedious and lesser efficient task to perform.	
18.09.2024	Removal of CLAHE, SA-DWT and Inverse SA-DWT	Outputs were not satisfactory. Instead, image was being converted to Grayscale.	

1. INTRODUCTION

1.1. Purpose of the Project

This project aims to design a lightweight, yet strong C++ program that enhances the visual quality of images captured in either low-lighted or otherwise degraded conditions. This program applies several enhanced techniques of image processing: Gamma Correction, Brightness Adjustment, Contrast Enhancement, Color Balancing and Gaussian Smoothing, entirely without the need for libraries, hence suitable for resource-constrained systems.

1.2. Target Beneficiaries

Direct beneficiaries of this project are the users that work on low-end or resource-constrained devices like embedded systems or mobile platforms, which are too expensive in terms of resource usage. This also constitutes an area of academical interest since it clearly presents the basic image enhancement techniques in C++. Also, it would be a great benefit for security systems.

1.3. Project Scope

With this project, it develops and implements a C++ light weighted enhancement system to improve the visual quality of degraded images captured in low light conditions. Without any external library dependences for compatibility, it offers core enhancement techniques such as brightness adjustment, contrast enhancement, gamma correction and shadow handling and Gaussian smoothing-only. It also considers the output quality based on predetermined visual standards and addresses all distortion in the image (blur, noise, low contrast). It will be implemented in such a way that new enhancement techniques can be added as required and real time enhancements will be achieved for low end devices enabling it to scale. The program should process different image formats and therefore the above example should be followed with PPM and the image quality should be evaluated based on subjective visual judgement.

1.4. References

- Research Papers
  - [1] A Dynamic Histogram Equalization for Image Contrast Enhancement M. Abdullah-Al-Wadud, Md. Hasanul Kabir, M. Ali Akber Dewan, and Oksam Chae, Member, IEEE
  - [2] Low-Light Image Enhancement: A Comparative Review and Prospects WONJUN KIM , (Member, IEEE) Department of Electrical and Electronics Engineering, Konkuk University, Seoul 05029, South Korea
  - [3] Gray World based Color Correction and Intensity Preservation for Image Enhancement N.M. Kwoka, D. Wanga, X. Jiab, S.Y. Chenc, G. Fangd and Q.P. Hae aSchool of Mechanical and Manufacturing Engineering The University of New South Wales, Australia bSchool of Information Technology
- Diagrams: Draw.io

2. PROJECT DESCRIPTION

2.1. Reference Algorithm

The image enhancement process follows a structured pipeline involving several key algorithms designed to improve visual quality:

- Gaussian Smoothing: A Gaussian filter is applied to the image to reduce noise and smooth out sharp edges. The kernel size and sigma value are customizable, allowing for flexibility in the degree of smoothing applied.

- Gamma Correction: This algorithm applies the non-linear transformation of pixel intensities by its brightness levels. Uses of pre computed gamma tables for an efficient computation.

- Brightness Adjustment: In this step, we add extra user specified amount of the pixels value of that image so that the image will become brighter or darker.

- Contrast Correction: The adjusted relative contrast of the image to a midpoint reference is the basis of it. This shifts the difference in brightness between the light and the dark parts of your image to more or less than before correction.

- Shadow Boost: We make an attempt here to mitigate the intensity of the regions endowed with shadow. Setting an image's value of a pixel from the darker part of the image by a lower number is how this is done.

- Gray World Color Balance: This algorithm balances the overall color tones in an image by adjusting the RGB values, assuming the average color in the image should be gray. This correction helps in neutralizing color casts caused by lighting conditions, ensuring more natural-looking images.

These algorithms offer strong enhancement for images while preserving efficiency, so making the system suitable for low end devices.

2.2. Data/ Data Structure

The image enhancement system primarily deals with image data, which is represented in a structured format that allows for efficient manipulation and processing. The data can be broken down into the following categories:

2.2.1. Input Data:

- Image Files : The system accepts image files in most formats (.jpg, .png, .ppm, etc). Images are loaded and converted into PPM(Portable Pixmap) format, containing image data in textual form, that allows faster processing.
- Pixel Data: Once the file for image is loaded, it is represented as 2D matrix. Each pixel actually contains color values defined by channels: Red, Green, Blue. The value of each channel ranges from 0 to 255, representing intensity of the color at that pixel.
- Metadata: All other information related to the image itself, such as dimensions: width, height, color depth and format

2.2.2. Internal Data Structures:

i. Matrix Representation
   - The image is considered as a 2D matrix (or 2D vector in C++), where every element of the matrix is a pixel. For a color image, each entry of the matrix has three values, while for a grayscale image, each entry has just one intensity value.
   - Matrix representation works well for applying transformations like Gaussian smoothing, adjustment of contrast, gamma correction, etc., typically those involving operations on the pixels.

ii. Kernel Matrices (filters)
   - Smoothing using a Gaussian kernel and other such convolutions rely on small kernel matrices (3x3, 5x5, etc.) scanned over the image to exercise blurring or sharpening effects.

2.2.3. Output Data

- Enhanced Image Data: After applying the enhancements, the processed pixel information is stored in a 2D matrix once more that represent the modified image. This data is then written back into a JPEG, a standard input format.
- Also, for improvement in the output image, user input is processed to update current image in real time.

2.3. SWOT Analysis

Strengths:

- Machine Independent: It means that the system or tool can execute on different machines without specific dependencies, meaning that this system is flexible across different platforms.

- Versatility: This term means that the system should be able to take up various tasks and hence able to adapt in various use cases or environments.

- Less Overhead on the Machine: This term relates to a system being resource-efficient so that it does not place undue strain on the machine processing power or memory.

- Correct Computation: This is the dependability and the accuracy of calculations that the system computes, which is very important in most applications.

Weaknesses:

- Hard Parameter Tuning: It implies that the best adjustment for the parameters used in the system becomes cumbersome and time-consuming and essentially requires some expertise or lots of trial-and-error.

- High Complexity: Suggests that the system is one which would take a long time to learn, or hard to understand or maintain since it may have a very complex structure or dependencies.

- Plain UI: Indicates that the user interface may not be very user-friendly or appealing in looks; this might negatively affect usability especially for non-technical users.

Opportunities:

- Improvements through Feedback Loops: Appears to suggest that the system can be iteratively improved through user feedback or performance data to improve and refine the system itself.

- Potential for Real-Time Applications: It suggests potentially extending the system such that it can be used in real-time scenarios, thus enhancing its applicability in changing environments.

- Educational Value: It is suggestive of the system being a valuable learning tool by providing insight into the technical concepts, especially to students or to those new to the field.

Threats:

- Competition with Other Libraries: It is also represented in terms of the threat posed by rival systems or libraries that offer similar functionality or better for which the system might suffer a loss in the user base or market share.

- Maintenance Problem: It mentions the issues with the maintenance side, like not keeping up with the current status of the system, eliminating bugs or ensuring backward compatibility with the evolution of technology.

- Technology Change: Indicates the risk that the system will become outdated or become useless as technology is constantly evolving in modern society or it is altered by changing industry standards.

2.4. Project Features

This image enhancement system provides several salient features for the improvement of visual quality of degraded images acquired under low light conditions.

- Brightness Adjustment – It permits the alteration of brightness within an image so that visibility is improved in some underexposed regions.

- Contrast Enhancement – It creates higher contrast in images in order to make fine details more conspicuous and maintain overall visual balance.

- Gamma Correction – This corrects the gamma levels to fine-tune the overall tone of the image, thereby making shadows and over-illuminated areas more prominent.

- Gray World Color Balance – Automatically balances the color balance of the photograph assuming that, on average, the average color of any scene should be gray, eliminating color casts resulting from lighting.

- Gaussian Smoothing – Filters the noise and smoothens images based on a Gaussian filter, which is applied to eliminate random variations and artifacts while preserving integral image information.

- Simultaneous Processing – Enhances an image by combining several techniques such as contrast, gamma corrections, and color balance while ensuring a maximum balance of clarity and output realism.

- User Feedback Mechanism – Provides a feedback mode of interaction for the user to input his or her preference or change, like increasing brightness or contrast, and thus enables him or her to fine-tune the enhancement as needed.

Together these form a strong and efficient image enhancement system well suited for nearly all degraded or low-quality images.

2.5. User Classes and Characteristics

The image enhancement system is going to cater to the following classes of users:

- General Users – People having less technical knowledge and require an easy-to-use interface to enhance their personal images (especially with low-light), like casual photographers. In all such cases, users are going to appreciate the simplicity and automated enhancement nature of the system.

- Low-End Device Users: People using systems with limited resources that demand image enhancement with no overhead of resource-intensive hardware. The light construction of the system makes it compatible with these devices, such that it would fit perfectly within spaces with very limited computation resources.

- Police and Security Systems – Law enforcement agencies and security personnel that need images of higher resolution to be captured for video-surveillance purposes or are part of the process of evaluating evidence. This type of user shall take the same benefit from the system capabilities to enhance low illumination or otherwise degenerated images to facilitate investigations and surveillance operations by making important details visible.

- Photographers and Visual Artists – Professionals who would polish and fine-tune images within their creative workflow. They most probably would need fine control of brightness, contrast, as well as color balance, and might use the feedback mechanism to adjust images further.

- Researchers and Developers: These are users who have technical background, especially computer vision, AI, or image processing. They would probably use the system to analyze and optimize their image sets or test techniques of the image enhancement in order to find the most efficient processing and quality results.

These classes of users have varied expectations during interaction with the system from simple ease of use to fine-tuned control and efficient processing, which the system captures through this flexible feature set.

2.6. Design and Implementation Constraints

- Programming Language – The project will be implemented in C++ since its efficiency in low-level operations is significant for the purpose of image processing.

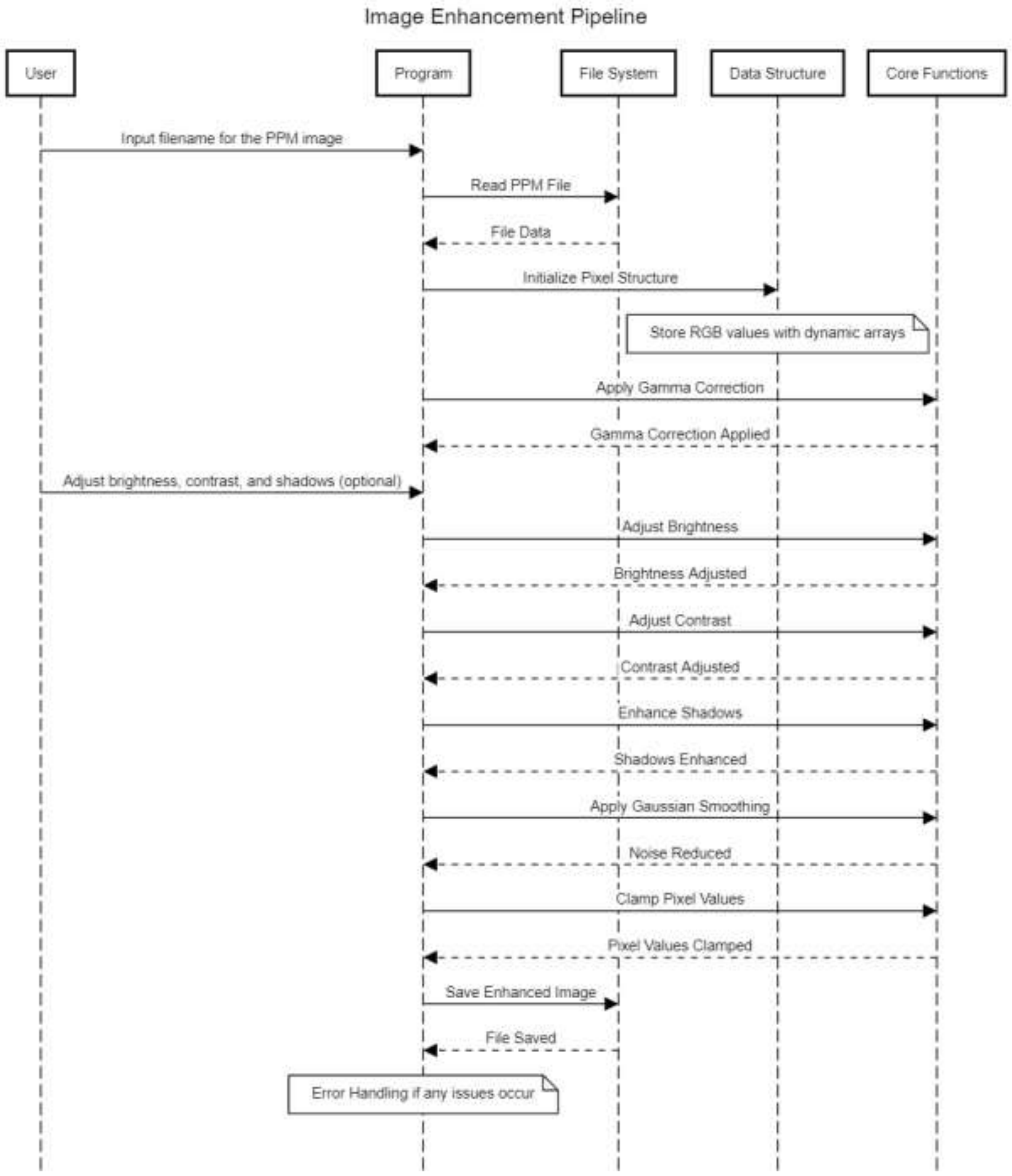
- Image Format Support – The software will only support PPM format which takes away most of the functionality as it will not support other formats like JPEG or PNG. This constraint focuses on optimizing the operations for the characteristics of PPM files.

- Performance Limitations – It should handle resolutions up to 4K to gain in terms of execution time. Big images might mean big delays of processing or bigger consumption of memory.

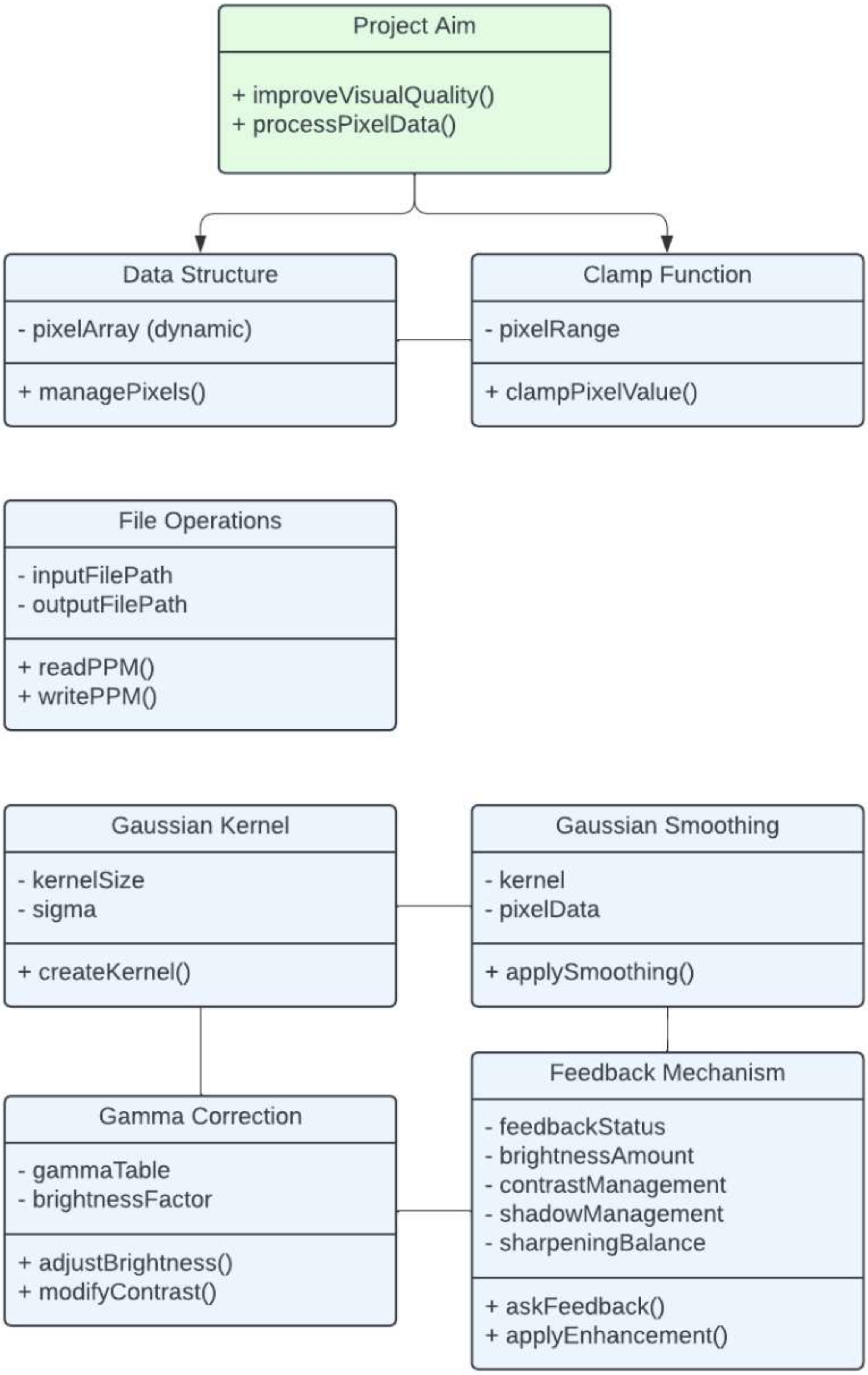
- Memory – Code will be not powerful in the usage of memory because of work with big images or with some types of memory-expensive operations like generating large kernels for a Gaussian filter.

- User Interaction – User interaction is based on a command line interface (CLI). This is a limitation which will be reflected in the count of GUI features. This makes its implementation rather simpler though can have a bearing on the user experience for non-technical users.
- Error Handling – The application should implement error checking during the implementation of file operations, that is checking valid file paths also what to do with unsupported file formats or corrupted images.

2.7. Design Diagrams

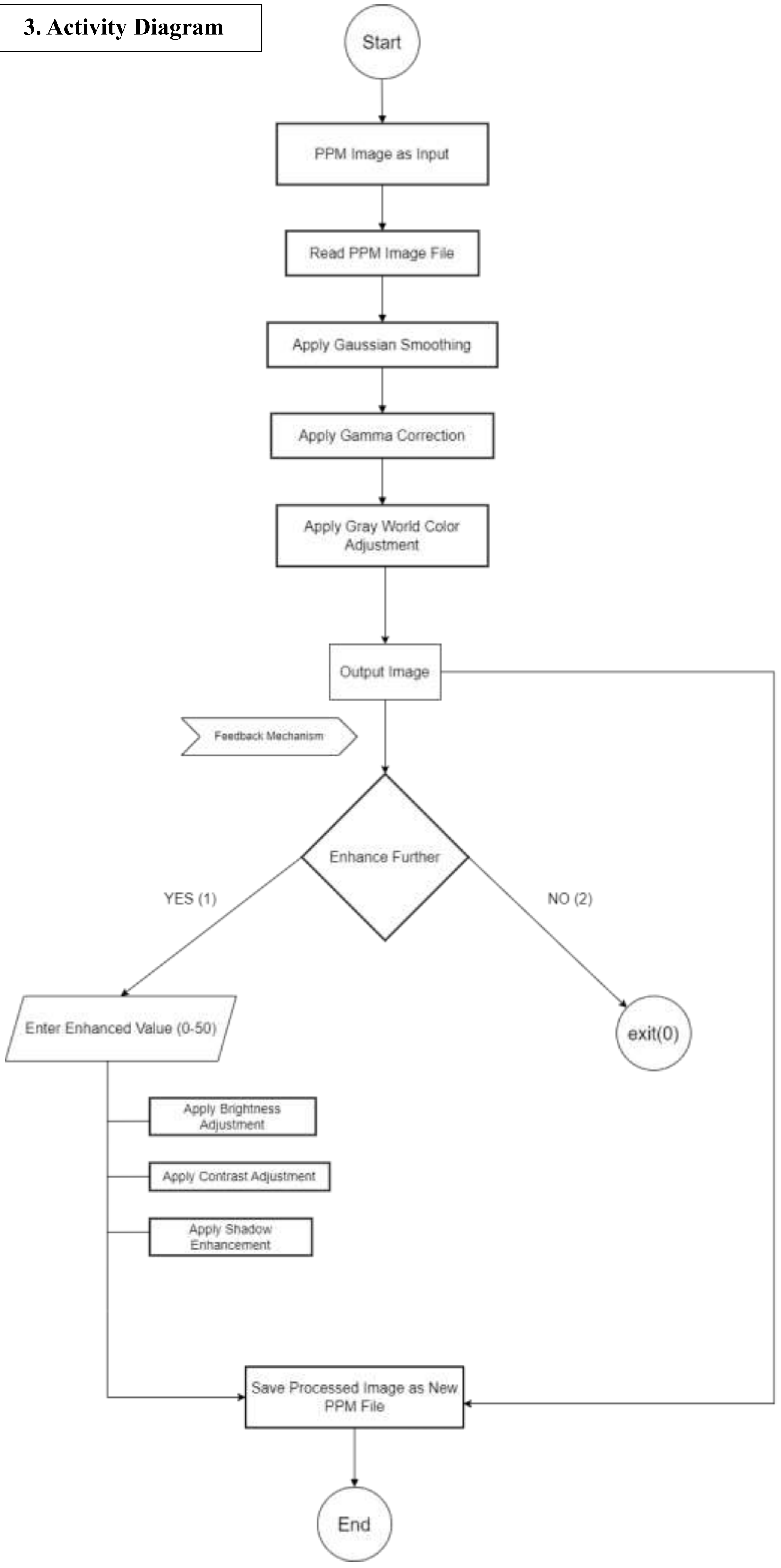


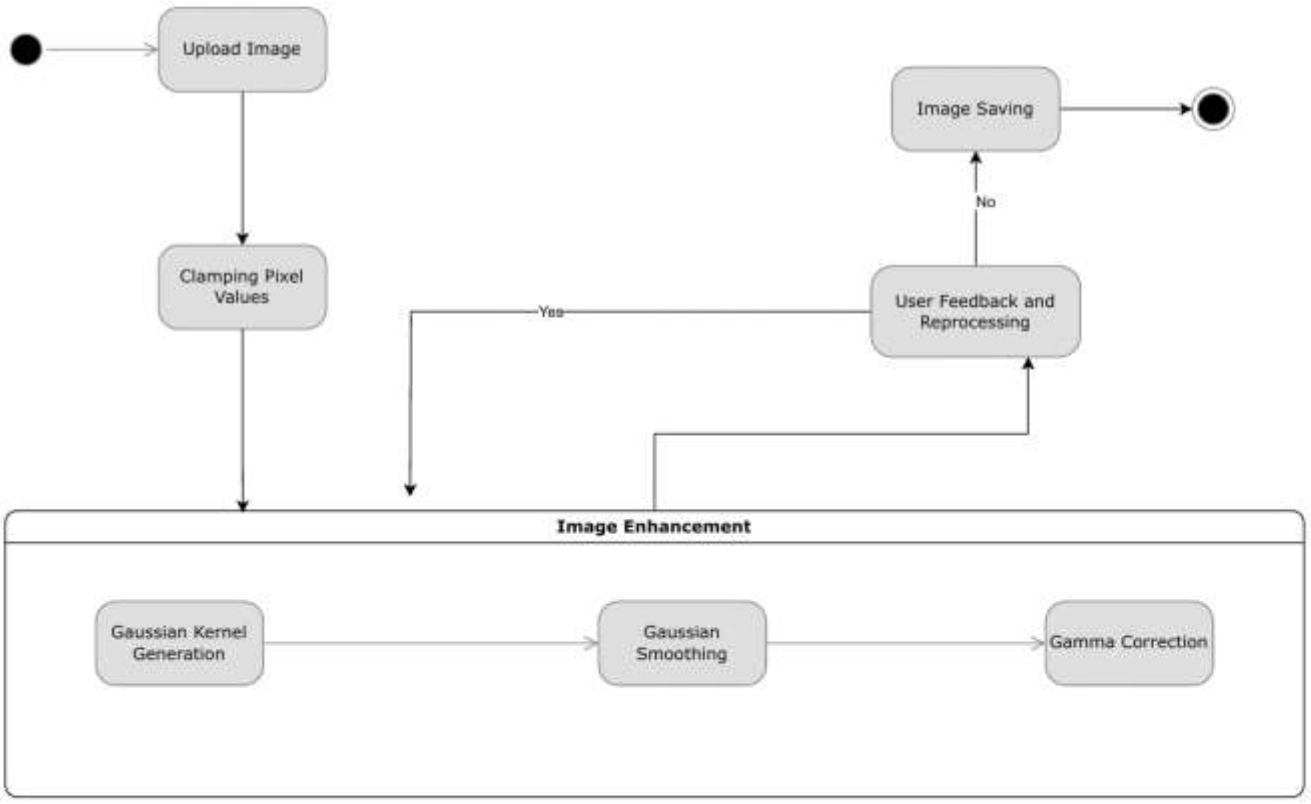
1. Sequence Diagram



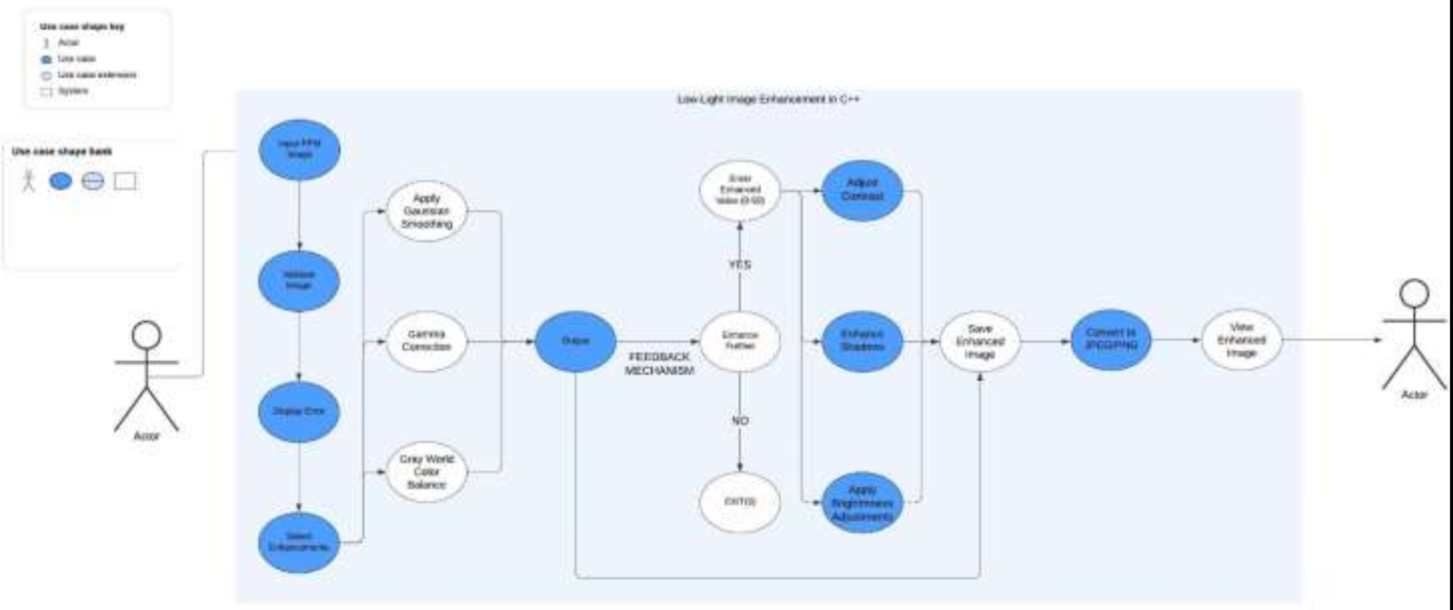
2. Class Diagram

3. Activity Diagram





4. State Diagram



5. Use Case Diagram

6. DFD (Level 2)

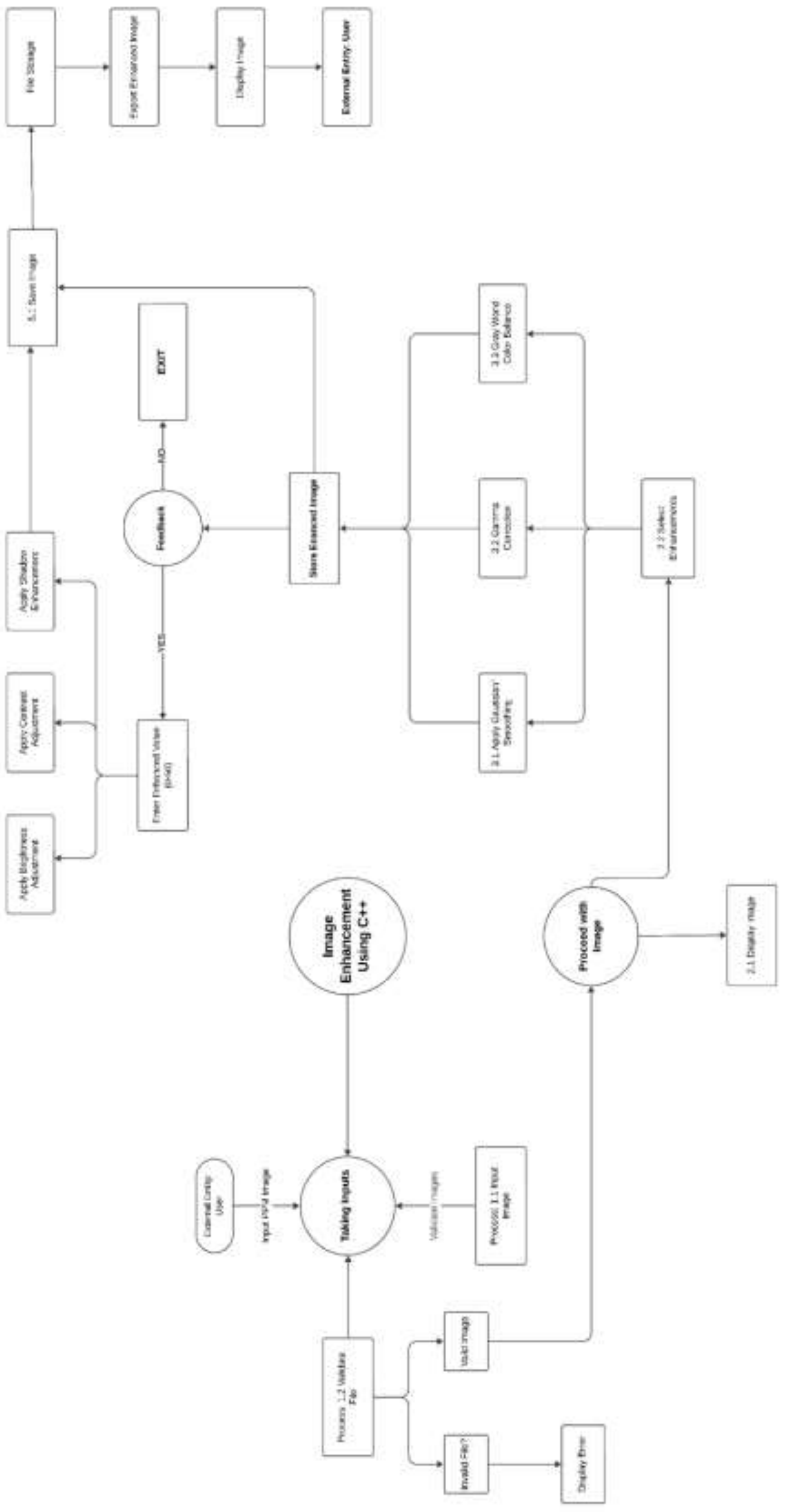


Image Enhancement Using C++
Taking Inputs
External Entity: User
Input PPM Image
Validate Images
Process: 1.1 Input Image
Process: 1.2 Validate File
Valid Image
Invalid File?
Display Error
Proceed with Image
2.1 Display Image
2.2 Select Enhancements
3.1 Apply Gaussian Smoothing
3.2 Gamma Correction
3.3 Gray World Color Balance
Store Enhanced Image
Feedback
YES
NO
EXIT
Enter Enhanced Value (0-50)
Apply Brightness Adjustment
Apply Contrast Adjustment
Apply Shadow Enhancement
5.1 Save Image
File Storage
Export Enhanced Image
Display Image
External Entity: User

2.8. Assumptions and Dependencies

- User Knowledge – It is assumed that the user is aware of the basics of image formats and also of enhancement techniques because the software does not include tutorials that are lengthy or even guide to be used throughout all phases of development.

- Computing Resources – The project will have access to computing resources as needed: the additional RAM and processing power to run the application easily and efficiently, even for more greatly dimensioned images.

- Development Environment – The assumptions are that the implementation is built within a standard C++ development environment (for example, Visual Studio or GCC), and thereby compatibility to most common C++ libraries is assured.

- Input File Integrity – The users are expected to provide valid PPM files to be processed. For this scenario, the application will have error handling for files which are not valid but it will not check them intensely beyond some basic checks.

- Dynamic Memory Management – The implementation assumes that appropriate dynamic memory management techniques are used so memory is dynamically allocated for and released from data structures holding the images. Memory leak is also assumed not to occur during development.

- Libraries Available – For this project, it is assumed that any available mathematical operations library is available and compatible with the chosen development environment.

- User Permissions – We assume that users have read and write permissions from and to the file system, especially to be able to save the processed images.

3. SYSTEM REQUIREMENTS

3.1. User Interface

- Input Handling: User will enter the system interactively through a console-based input giving the name of the image file and will have options to provide additional enhancement, such as brightness enhancement. User will enter numeric input for brightness levels (0-70) and choices on continuing image enhancements.

- Output Display: The status about the successful application of gamma correction, brightness adjustments, and saving of enhanced images is displayed through the console.

3.2. Software Interface

- Operating System – This system is more flexible and can be used with general-purpose operating systems that support C++, such as Windows and Linux.
- Development Environment – The program uses standard C++ libraries like <iostream>, <fstream>, <vector>, <string>, <algorithm>, and <cmath>. These can compile with any C++ compiler, including GCC or MSVC.

3.3. Database Interface

The project does not involve any database. Images are directly read and written from and to the file system in PPM format. The system treats input/output of binary image files making use of objects of ifstream and ofstream.

3.4. Protocols

No network protocols are involved with this code. All operations are local to the machine, reading images from disk and processing them in memory in line.

4. NON-FUNCTIONAL REQUIREMENTS

4.1. Performance Requirements

- Efficiency – It should be really efficient on low-end systems. Thus, it doesn't rely on any external libraries so it stays lightweight. Gaussian smoothing and gamma correction should be rather fast for moderate-sized images (up to few MBs).

- Memory Usage – It is optimized so images can be worked on in memory using vectors so that there can be minimal usage of memory.

4.2. Security Requirements

- File Handling Security – It supports file input but doesn't deal with sensitive data, databases, etc. It is required to only handle errors properly, namely an unsupported format, as well as a file that can't be opened.

4.3. Software Quality Attributes

- Maintainability – The code has a modularity structure such that the improvement technique for every technique is encapsulated in its own separate function, making it easier to maintain or extend by incorporating new filters or processing methods.

- Portability – The program will be written in standard C++ and will just run in any platform which supports C++ with minimal changes.

- Usability – The console-based interface is primitive but of course not sophisticated in functionalities compared to a graphical interface. A user has to enter a valid image filename and numeric input for enhancement controls.

- Reliability – The system implements the basic error handling, ensuring that unsupported formats, etc., are handled without causing any system crashes.

5. OTHER REQUIREMENTS

- Portability – The developed image enhancement system must be portable, running on environments with the presence of Windows and Linux with minimal, if any change at all. This includes low-end devices where computation resources will be limited.

- Maintainability – Such a codebase should be well commented, modular, and easy to extend or adapt functionality by future developers or users of the system. Clean code practices must be followed to facilitate understanding and modification.

- Usability – The system should be user-friendly enough for a general user but also provide options for professionals. Feedback mechanisms are supposed to be simple and intuitive so that even a non-technical user can enhance pictures without much effort.

- Regulatory Compliance – If it's being utilized by the police or other security agencies, then in any event it needs to be in compliance with the appropriate data privacy or security regulations; thus, image information would be treated with care as sensitive.

- Deployment Considerations – The system should have guidelines on how to deploy in different environments including providing a lightweight version for low-resource devices and a high-performance version for resource-rich systems.

This means the additional requirements ensure the system is versatile and easy to maintain, adaptable for user needs and their environments.

APPENDIX A: GLOSSARY

1. Image Enhancement – The process of enhancing the visual quality of an image by applying image processing and enhancement techniques such as contrast adjustment, brightness adjustment, grain removal etc.

2. Pixel – The smallest unit in an image. Represents a single color or shade. It is defined by either black or white, in grayscale images or by red, blue or green, in RGB images.

3. JPEG/ JPG – Stands for "Joint Photographic Experts Group). Commonly used image format that used lossy compression to reduce file size while maintaining acceptable image quality. Widely used for storing and sharing photographic images.

4. PPM (Portable Pixmap Format) – A simple image file format which stores pixel data in textual format, with each pixel defined by its red, green or blue (RGB) components.

5. Low-End Devices – Devices with limited processing power and memory, often requiring optimized software solutions for efficient performance.

6. RGB Channels – The three-color channels (Red, Green, Blue) used to represent images. Each channel controls the intensity of its respective color in a pixel.

7. Vectors in C++ – A dynamic array structure in C++. Allows elements to be stored and accessed in sequence. Used for storing image pixel data during processing.

8. Kernel – A small matrix used for convolution operations in image processing. It defines how each pixel in the image should be transformed by its neighbors.

9. Gaussian Noise – A type of statistical noise with a probability distribution similar to the Gaussian(normal) distribution, often found in low-light images or during image acquisition.

10. Gaussian Smoothening – A filtering technique that uses a Gaussian function to smoothen the image by reducing noise and fine details.

11. Gaussian Kernel – A specific type of kernel that applies the Gaussian function to perform smoothing on an image, giving more weight to nearby pixels in the convolution process.

12. Gamma Correction – A nonlinear operation used to encode and decode the brightness levels of an image, enhancing visibility in low-light conditions.

13. Gray World Color Balance – A technique for color correction that assumes the average color in an image is gray. It adjusts the color channels to balance the overall color composition.

14. User Feedback Mechanism – A system that allows user to provide input on how they wish to enhance the image, dynamically adjusting the processing algorithm based on their preferences.

APPENDIX B: ANALYSIS MODEL

The structural and behavioral analysis of the image enhancement system for this project revolves around ensuring that the applied algorithms enhance the quality of the image under various conditions. The main components that make up the analysis are:

1) Image Processing Flow – Load the input PPM image and pass it through a series of enhancement steps including brightness adjustment, contrast modification, shadow enhancement, gamma correction, and Gaussian smoothing. Each step applied in sequence improves the quality of the visual image but not effects on the artifacts.

2) Mathematical Models –
   - Gaussian Smoothing. Utilizes a Gaussian kernel, effectively reducing noise of an image by blurring it; its properties rely on the spread of intensity from each pixel to its neighbors.

   - Gamma Correction: It adjusts the brightness and color intensity using a mathematical formula that is designed to transform pixel values based on a power-law function so that the final image has realistic brightness levels across all regions.

   - Gray World Color Balance. A channel-average method for adjusting RGB color levels so that the color tone and all effects created will be neutrally relative.

3) Data Structure Analysis –
   - This system heavily depends on the use of vectors, where dynamic storage and manipulation of pixel data prevail. Every pixel is represented by a structure which holds its RGB values, and during enhancement processes, it is modified.

4) Performance Considerations –
   - In order to maintain the efficiency, passes over the image will be minimized and memory usage optimized especially in filter and transformation applications.
   - Pre-computed Gaussian kernels and lookup tables for gamma correction reduce redundant computations; therefore, a net processing time is minimal.

These implementations ensure the proper systematic application of the enhancement techniques. The pipeline is flexible to adjust for various image data variations.

APPENDIX C: ISSUES LIST

1) User Feedback Mechanism Integration:
   Status – Open
   Description – The system does not have a robust feedback mechanism whereby users can, depending on their preference, propose further adjustment of images.
   Action – Incorporate a feedback loop from the user for fine-tuning improvement such as brightness, contrast, and smoothing.

2) Gray World Color Balance Accuracy:
   Status – Open
   Description – The Gray World Color Balance algorithm was working fine, but to be more accurate concerning color representation, it needs a test on many low-light and high-noise images.
   Action – Calibration process and testing on various datasets

3) Integration of UI with Backend:
   Status – Open
   Description – The Front-end user interface and the backend C++ image processing engine require a lot more smoothening in it especially in terms of real-time updates and very friendly interactions.
   Action – Make sure proper information flows from the UI to the backend processing.

4) Optimization for Images of High Resolution in Memory:
   Status – Open
   Description – The current system seriously gobbles up memory when working on the images in high resolutions.
   Action Plan – Learn about memory management techniques and optimize the image processing pipeline such that large images are processed with better efficiency.