RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

CP23211 ADVANCED SOFTWARE ENGINEERING LAB

Laboratory Record
Note Book

Name :……VASEEKARAN R……………………………………………...

Year / Branch / Section : ……1st YEAR/ ME-CSE….. …………………….

University Register No. : ………2116…….. …………………

College Roll No. : ……………2116230711013…………………………………….

Semester : ……………………2…………………………………………………

Academic Year : ………2023-2024…………………………………………….

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

BONAFIDE CERTIFICATE

Name: VASEEKARAN R

Academic Year: 2023-2024 Semester: 2 Branch: ME-CSE

Register No:	2116230711013

Certified that this is the bonafide record of work done by the above student in the CP23211-Advanced Software Engineering Laboratory during the year 2023- 2024

Signature of Faculty-in-charge

Submitted for the Practical Examination held on 22/06/2024

Internal Examiner External Examiner

INDEX

UNDERWATER IMAGE ENHANCEMENT USING FUZZYLOGIC AND XGBOOST

OVERVIEW OF THE PROJECT:

Underwater imaging poses significant challenges due to the inherent attenuation and scattering of light in water, leading to degraded contrast and color fidelity in captured images. In this paper, we propose a novel underwater image enhancement method based on optimizing both contrast and attenuation difference. Our approach leverages the physical properties of underwater light propagation to enhance visibility and improve image quality. First, we model the attenuation of light in water using an empirical function, which enables us to estimate the attenuation difference between the original scene and the captured image. Next, we introduce a contrast enhancement scheme that adapts to the estimated attenuation difference, effectively restoring contrast while preserving image details. Additionally, we integrate a color correction mechanism to address the color distortion caused by underwater conditions, resulting in visually pleasing and natural-looking images. To evaluate the performance of our method, we conduct experiments on various underwater images captured under different conditions. The results demonstrate that our approach outperforms existing methods in terms of both subjective visual quality and objective image quality metrics. Moreover, our method exhibits robustness across a wide range of underwater environments, making it suitable for practical applications in underwater photography, marine research, and underwater surveillance.

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

EXP.NO: 1 **DATE: 5/03/2024**

CONTENTS

UNDERWATER IMAGE ENHANCEMENT USING FUZZY LOGIC AND XGBOOST

1. Introduction

1.1 Purpose:

The Purpose of Underwater image enhancement is to improve the visibility and clarity of photographs taken underwater. Due to the unique challenges posed by water, such as light absorption and scattering, images captured beneath the surface often suffer from poor visibility, color distortion, and reduced detail. This specialized app utilizes algorithms designed to mitigate these issues by enhancing contrast, adjusting color balance, and reducing noise, thereby restoring the natural beauty of underwater scenes.

1.2 Scope:

The scope of underwater image enhancement extends from scientific research to cultural preservation and recreational enjoyment, making significant contributions to our understanding and appreciation of the underwater world. By utilizing modern algorithms and methods, we can further improve the quality of the images.

2. Overall Description

2.1 Product Perspective:

The integration of fuzzy logic with XGBoost ensures adaptability to different underwater conditions and scenarios, providing a robust solution that can continuously learn and improve with user feedback. The product cater to a range of users and applications.

2.2 Features

2.2.1 Color Correction and Enhancement:

- The user can use Fuzzy logic can be employed to analyze and adjust color tones, compensating for the absorption and scattering of light in water.

- XGBoost can further refine these adjustments based on learned patterns from a vast dataset of underwater images

- Ability to alter and enhance color to suit the purpose.

2.2.2 Contrast Enhancement:

- Fuzzy logic can intelligently adjust contrast levels to enhance visibility and detail in underwater scenes

- XGBoost can optimize these adjustments to maintain natural-looking contrasts while enhancing image clarity.
- Ability to adjust the contrast of the captured images.

2.2.3 Noise Reduction:

- The system can utilize fuzzy logic to identify and reduce noise caused by water particles and low-light conditions.
- Ability to enhancing image sharpness and reducing graininess.
- XGBoost can refine noise reduction algorithms based on statistical patterns.

2.2.4 User Interface and Control:

- The product can feature an intuitive user interface that allows users to adjust enhancement parameters manually .
- The user can also select automated modes based on specific underwater conditions
- This flexibility ensures usability across different user skill levels and application scenarios.

2.3 User Classes and Characteristics:

- Marine Biologists and Researchers: Uses the application forunderwater data collection and analysis. They prioritize accuracy and reliability in image enhancement.
- Underwater Photographers and Videographers: Capture artistic and documentary underwater images and videos

3. Specific Requirements

3.1 Functional Requirements:

3.1.1 Color correction and Enhancement:

- Users should be able to correct,change the color scheme.
- Utilize the algorithms fully to correct the color of the image.

- A proper interface for color correction and enhancement should be available.

3.1.2 Contrast Enhancement:

- Users can enhance and alter the color contrast of the captured image.
- User should be able to see the result of the enhanced image.
- Implement filters options for displaying various effects due to contrast enhancement .

3.1.3 Noise Reduction:

- Able to identity different types of disturbances.
- Categorize the disturbances and mitigate them.
- Reduce the noise and improve the overall quality of the image.

3.1.4 User Interface and Control:

- Proper user interface should be established.
- User should be able to give the captured image as input and get enhanced image.
- Overall interaction should not be tedious and easy to understand.

3.2 Non-Functional Requirements:

3.2.1 Usability

- The interface should be user-friendly and intuitive.
- Should be usable across all the platforms.

3.2.2 Performance

- Quick response time for loading the enhanced images.
- Scalability to accommodate a growing user base.

3.2.3 Security

- Implement secure user authentication and authorization.

- Encryption for sensitive data in case of images used in researches.

4. External Interface Requirements

4.1 User Interfaces:

- Intuitive design with easy navigation.
- Consistent theme and layout for a seamless user experience.

4.2 Hardware Interfaces:

- Compatible with Windows, Mac, Android and iOS devices.

4.3 Software Interfaces:

- Integration with image database for frequent training of the models.
- File storage integration for storing the enhanced image.

5. Conclusion

The integration of fuzzy logic with XGBoost for underwater image enhancement represents a promising advancement in the field of underwater photography. By leveraging fuzzy logic's ability to handle uncertainties and XGBoost's robust learning capabilities, the system can effectively tackle the unique challenges posed by underwater conditions. This combination allows for precise color correction, contrast enhancement, noise reduction, and detail preservation, ultimately resulting in clearer, more vibrant underwater images.This SRS document serves as a guideline for the development team to create a robust and user-friendly application.

SCRUM METHODOLOGY

EXP.NO: 2 **DATE: 14/03/2024**

1. Introduction

The Underwater image enhancement App aims to provide an intuitive and comprehensive solution for users to enhance their captured underwater images. The Agile Scrum framework will guide the development process, ensuring iterative enhancements and timely delivery of key features.

2. Objectives

- Develop a user-friendly application to enhance the captured underwater images.
- Prioritize user needs through iterative development cycles.
- to significantly improve the quality, clarity, and fidelity of underwater photographs.

3. Product Backlog Introduction

The product backlog is a dynamic list of features, enhancements, and fixes prioritized by the product owner. It serves as a road-map for the development team.

4. Product Backlog

4.1 For Users

- **Color correction and Enhancement:**
  - Users should be able to correct,change the color scheme.
  - Utilize the algorithms fully to correct the color of the image.

- **Contrast Enhancement:**
  - Users can enhance and alter the color contrast of the captured image.
  - User should be able to see the result of the enhanced image.

- **Noise Reduction:**
  - Reduce the noise and improve the overall quality of the image

- **User interface and control**:
  - User should be able to give the captured image as input and get enhanced image

5. User Stories

- **As a User:**
  - As a marine biologist, I want to capture clear images of marin life for research.
  - As an underwater photographer, I want to capture vibrant and colorful underwater images.
  - As an underwater archaeologist, I want to capture images to detect and preserve underwater artifacts.
  - As an environmentalist and conservationalist, I want to use these images to raise awareness about the various environmental issues.

6. Sprint

- A time-boxed iteration during which a set of user stories is implemented and tested.

7. Sprint Backlog

The sprint backlog is a list of tasks selected from the product backlog for a specific sprint. In other terms Sprint Backlog could also be defined as the subset of Product backlog which is chosen for a specific sprint. In general a sprint backlog allows the development team to work on the tasks necessary to implement the User Stories within the selected sprint.

8. Sprint Review

A meeting held at the end of each sprint to review and demonstrate the completed work.

Sprint 1 Review:

- The sprint review is a meeting that includes the demonstration of implemented features at that particular sprint that is conducted at the end of each sprint.
- In the Underwater image enhancement app the sprint backlog for the first week is the Use case Diagram and the sprint backlog for the second use case is Software Requirement Specification document.

9. Software Used

- **Development Platform:** MATLAB, Image processing toolbox

10. Conclusion

The Agile Scrum framework for the Underwater image enhancement App ensures a systematic approach to development, focusing on user needs and iterative improvements. By breaking down the project into manageable sprints, the framework allows for continuous feedback, resulting in a robust and user-friendly application for students and administrators alike.

USER STORIES

EXP.NO: 3 **DATE: 26/03/2024**

As a User:

1. Color correction and enhancement
User Story: I want to alter the color scheme of the underwater images to enhance them.
Acceptance Criteria:
- The user can enhance the color scheme to enhance the images.
- The improvement/ result is shown to the user.

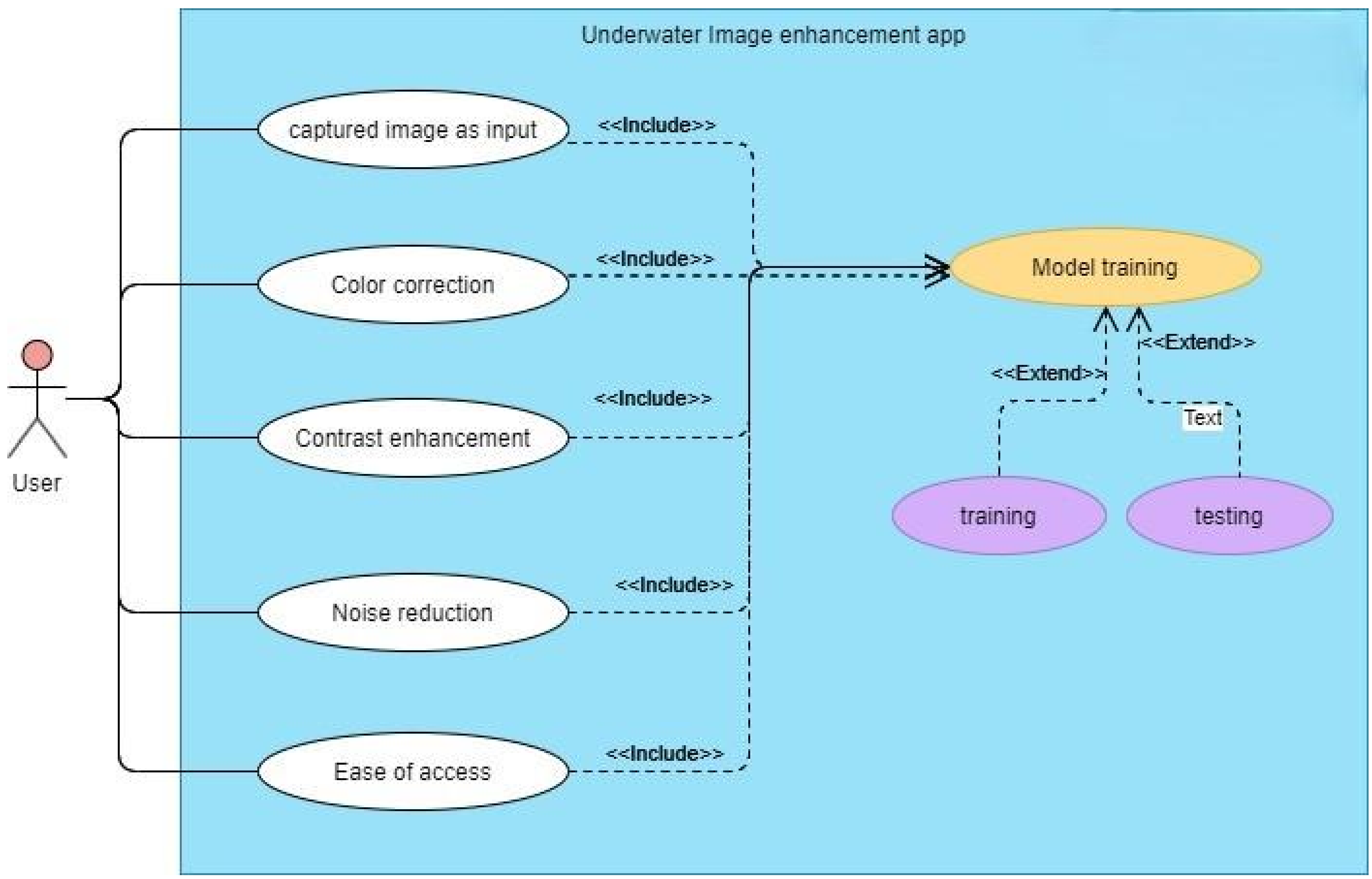
2. Contrast enhancement
User Story: I want to enhance the contrast of the capture images to improve the quality.
Acceptance Criteria:
- The user can enhance and alter the contrast of the captured image to improve the quality.
- The user can see the output to verify the improvements made.

3. Noise reduction
User Story: I want to reduce the disturbance and noise of the images.
Acceptance Criteria:
- The user can reduce or remove the disturbance in the images.
- The user can see the result and verify the improvements.

4. User interface and control
User Story: I want to be able to navigate through the app and access the features with ease.
- The user interface should be simple to learn.
- The user should be able to upload the image to be enhancement and download the enhanced images

USE CASE DIAGRAM

EXP.NO: 4 **DATE: 04/04/2024**



NON-FUNCTIONAL REQUIREMENTS

EXP.NO: 5 **DATE: 16/04/2024**

1. Performance

The Underwater image enhancement App should be highly responsive, with all major functions such as taking captured image and input, color and contrast enhancement,noise reduction and user interface. The app should handle up to 500 concurrent users without performance degradation, ensuring smooth and efficient user experience during peak times such as thebeginning of a semester.

2. Security

The app must implement robust security measures to protect user data. All user information, including login credentials, personal details, and stored images, must be encrypted both in transit and at rest. Access to the app should be controlled via secure authentication mechanisms, and administrators should have the ability to set and enforce password policies. The app should also include role-based access controls to ensure that users only have access to the features and data appropriate to their role.

3. Usability

The Underwater image enhancement app should provide an intuitive and user-friendly interface that is easy to navigate for all users, including those with limited technical skills. It should adhere to common usability principles, providing clear labels, logical flow, and accessible help documentation.

4. Reliability

The app must be reliable, with an uptime of at least 99.9%. It should include mechanisms for automatic fail over and recovery to minimize downtime and ensure continuous availability. Regular backups should be performed to prevent data loss, and in the event of a failure, the system should recover without data corruption or significant downtime.

5. Scalability

The Underwater Image enhancement App should be scalable to accommodate future growth in the number of users and the volume of data. The system architecture should support the addition of new features and integration. The app should be able to handle an increasing number of simultaneous users and data entries with minimal impact on performance.

6. Maintainability

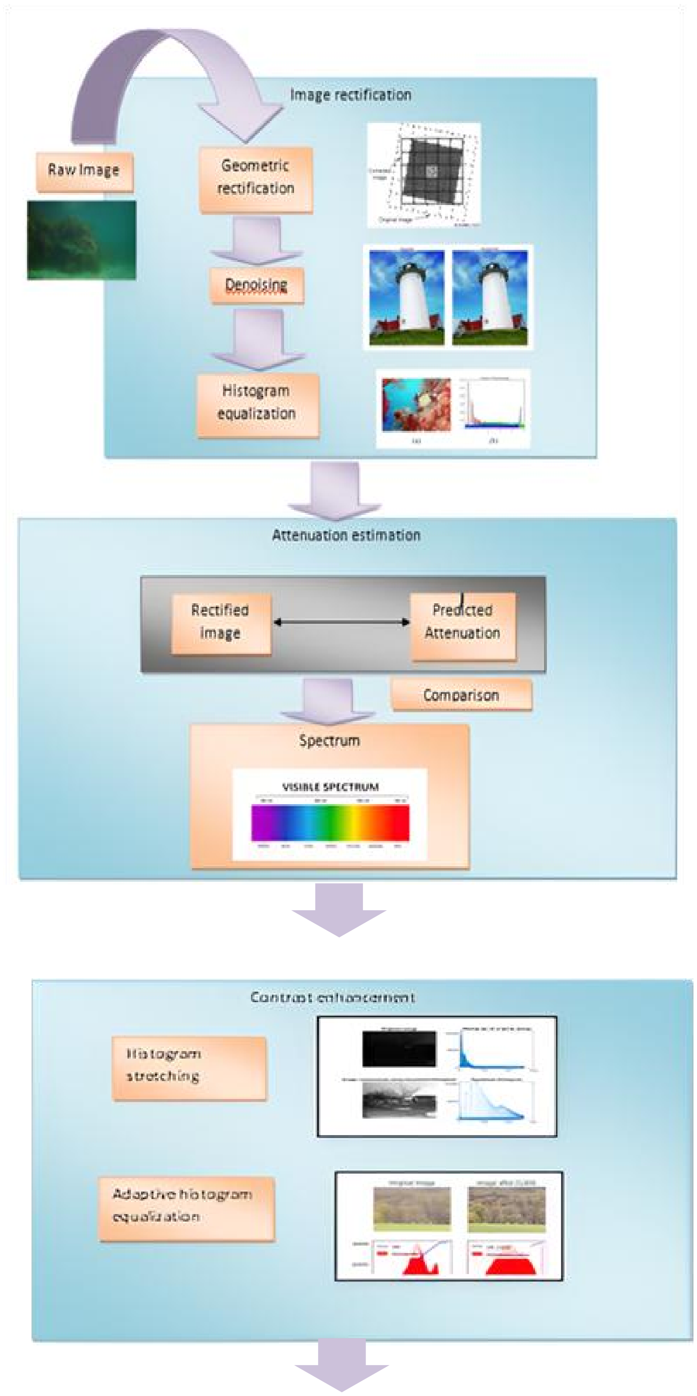
The app should be designed with maintainability in mind, enabling easy updates and bug fixes. The codebase should follow standard coding practices and be well-documented to facilitate understanding and modifications by different developers. Modular design principles should be applied, allowing individual components to be updated or replaced without affecting the entire system.

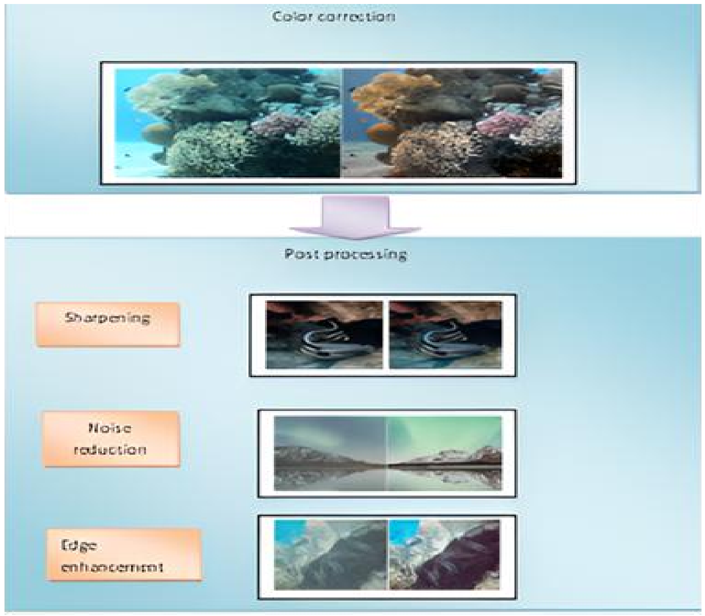
7. Compliance

The App must comply with relevant regulations and standards, including data privacy laws such as GDPR for users in Europe. The app should ensure that all data handling practices meet legal requirements for data protection and user privacy. Regular audits should be conducted to verify compliance, and any necessary adjustments should be made promptly to address new regulatory changes.

OVERALL PROJECT ARCHITECTURE

EXP.NO: 6 **DATE: 25/4/2024**





1. **Data Collection:** Underwater imaging surveys, ROVs, and underwater cameras are just a few of the ways we collect data on the ocean floor. There is a wide variety of water depths, clarity levels, and other environmental factors represented in the acquired data.

2. **Preprocessing:** Prior to improvement, the raw underwater photographs undergo preprocessing to eliminate noise, rectify lens distortion, and standardize exposure. To get the pictures ready for augmentation, preprocessing operations should include geometric rectification, denoising, and histogram equalization.

3. **Attenuation Estimation:** To calculate the attenuation disparity between the actual scene and the acquired picture, we use physical models of the propagation of light underwater. To do this, we must compare the measured spectrum features of the acquired picture to the predicted attenuation according to parameters like particle concentration, turbidity, and depth in the water.

4. **Contrast Enhancement:** To restore contrast while maintaining picture features, we dynamically alter the contrast enhancement settings based on the predicted attenuation difference. Histogram stretching, adaptive histogram equalization, and local contrast enhancement are some of the methods we use to boost visibility and picture quality.

5. **Colour Correction:** By adjusting the colour balance of the improved picture using colour correction techniques, we can fix the colour distortion that occurs underwater. This requires restoring colours that seem realistic by altering colour channels depending on the predicted attenuation difference and accounting for the selective utilization of wavelengths of light in water.

6. **Ensemble Fuzzy logic-Improved XGBoost:** Each pixel's value (intensity, color component, etc.) is converted into a membership degree for various fuzzy sets. These sets represent concepts like "bright, dark, edge-like or smooth." Membership functions define the degree to which a pixel belongs to each set. Imagine a bell-shaped curve for "bright" - pixels with very high intensity would have a membership degree of 1 (fully bright), while those with lower intensity would have decreasing degrees towards 0 (not bright).

   Here, the fuzzy rules come into play. These rules are defined by the user and represent relationships between image features. They use fuzzy terms like "very, somewhat, or slightly" to capture the ambiguity in image data. For example, a rule might be "If a pixel's intensity is very bright and its neighbor's intensity is very dark, then the pixel belongs to a strong edge."

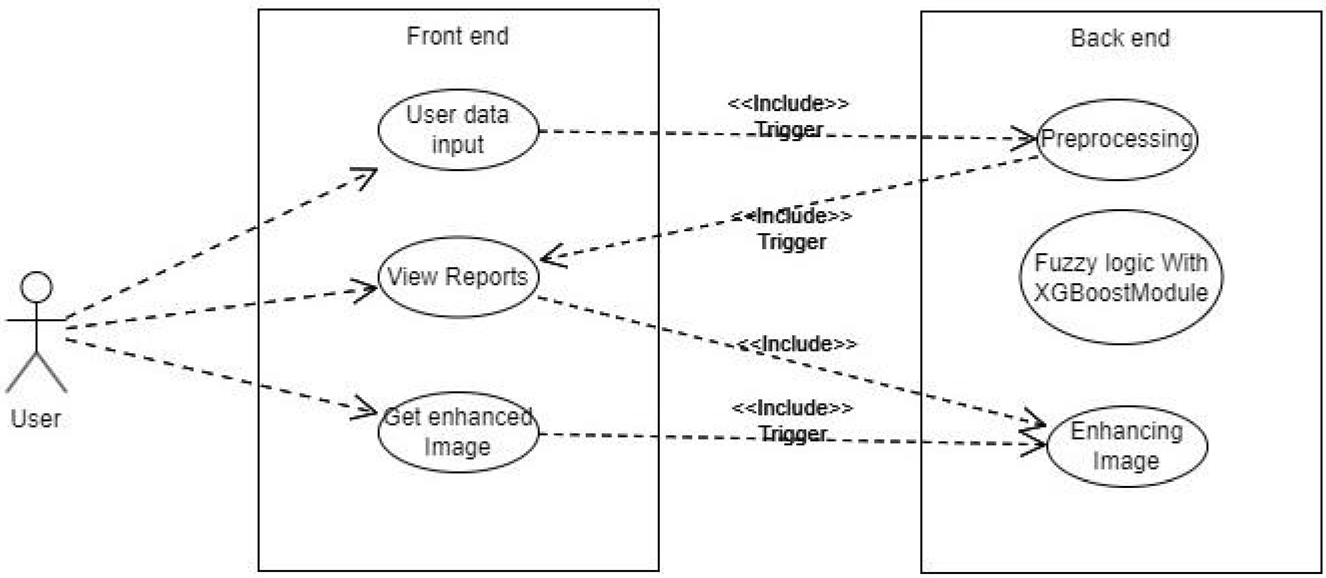
   While XGBoost is a powerful machine learning algorithm, it's not directly used in the typical fuzzy logic flow for image processing. XGBoost excels at supervised learning tasks, meaning it requires labeled data for training. In image processing using fuzzy logic, the focus is on defining fuzzy sets and rules based on human expertise rather than training a model. However, XGBoost can be a complementary tool in an image processing pipeline that
   leverages Fuzzy Logic.

7. **Post-processing:** Further post-processing is applied to the improved photos after enhancing the contrast and colour correction. In order to improve the overall quality and look of the picture, this may include sharpening, reducing noise, and enhancing the edges.

8. **Evaluation:** Our improved photos are lastly tested using subjective and objective measures to determine their efficacy. Human observers examine images for quality and authenticity as part of subjective judgment. We can quantify the efficacy of our enhancement strategy using measures like structural similarity index (SSIM), colour fidelity, and signal-to-noise ratio (SNR).

BUSINESS ARCHITECTURE DIAGRAM

EXP.NO: 7 **DATE: 02/05/2024**



Actor:

- User: Represents the primary user who interacts with the Underwater image enhancement application.This actor is involved in all frontend use cases.

Frontend Use Cases:

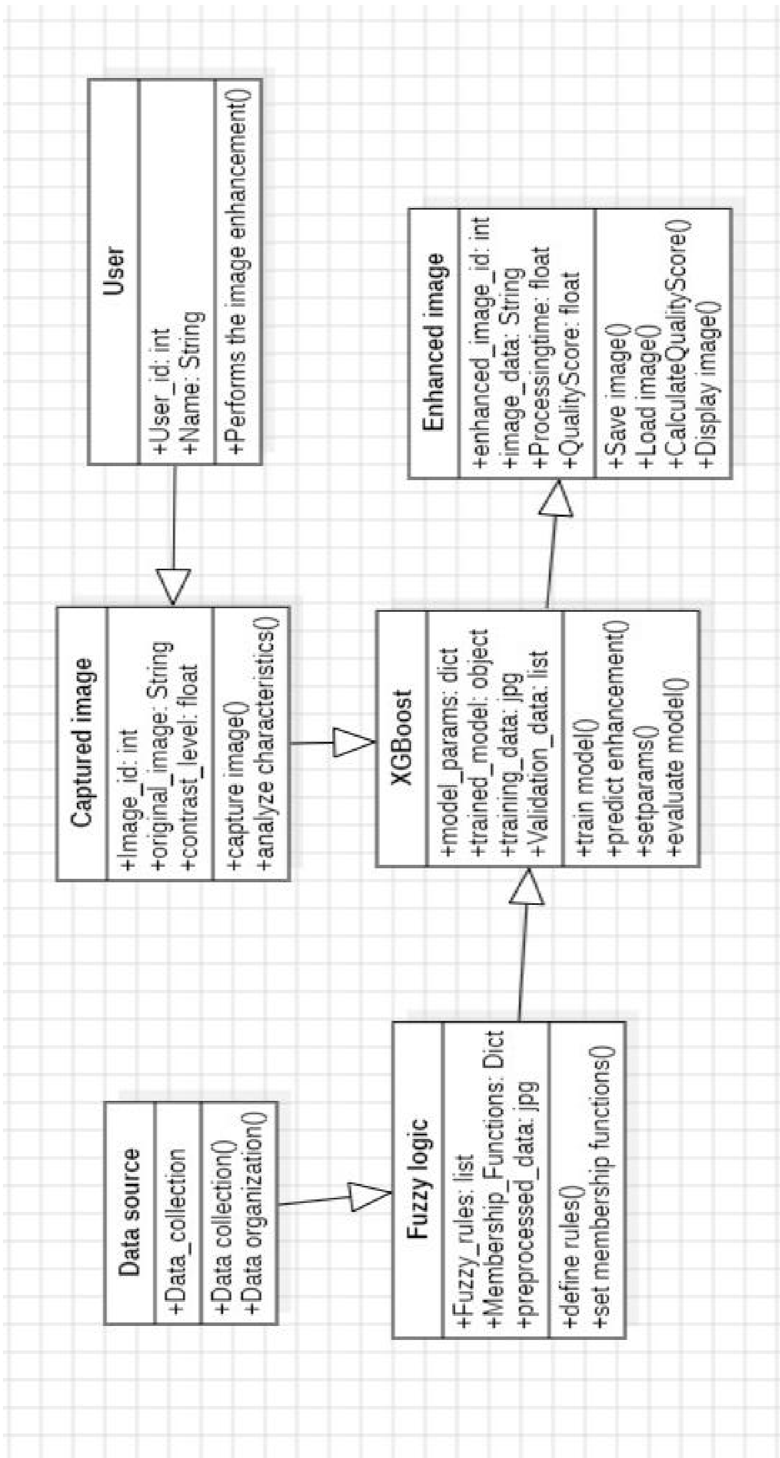
- **User data input:** User have to give their capture image as input.
- **View Report:** User can view and compare their original image and the captured image at each stages of the enhancement.
- **Get enhanced image:** The user gets the enhanced image with improved visual clarity as their output.

Backend Use Cases:

- **Preprocessing:** The image given by the user is taken as input and undergoes preprocessing.
- **Fuzzy logic with XGBoost**: The module used for enhancing the captured image by enhancing the color, contrast, reducing noise.
- **Enhancing image**: The module takes the user input as test data and enhances the image to give as output for the user.

CLASS DIAGRAM

EXP.NO: 8 **DATE: 07/05/2024**



Classes:

1. **Data source**: Collection of underwater image data for training the model.

2. **Captured image**: The underwater image captured by the user.

3. User: Represents the user using the system.

4. Fuzzy logic: Preprocesses the data using fuzzy rules.

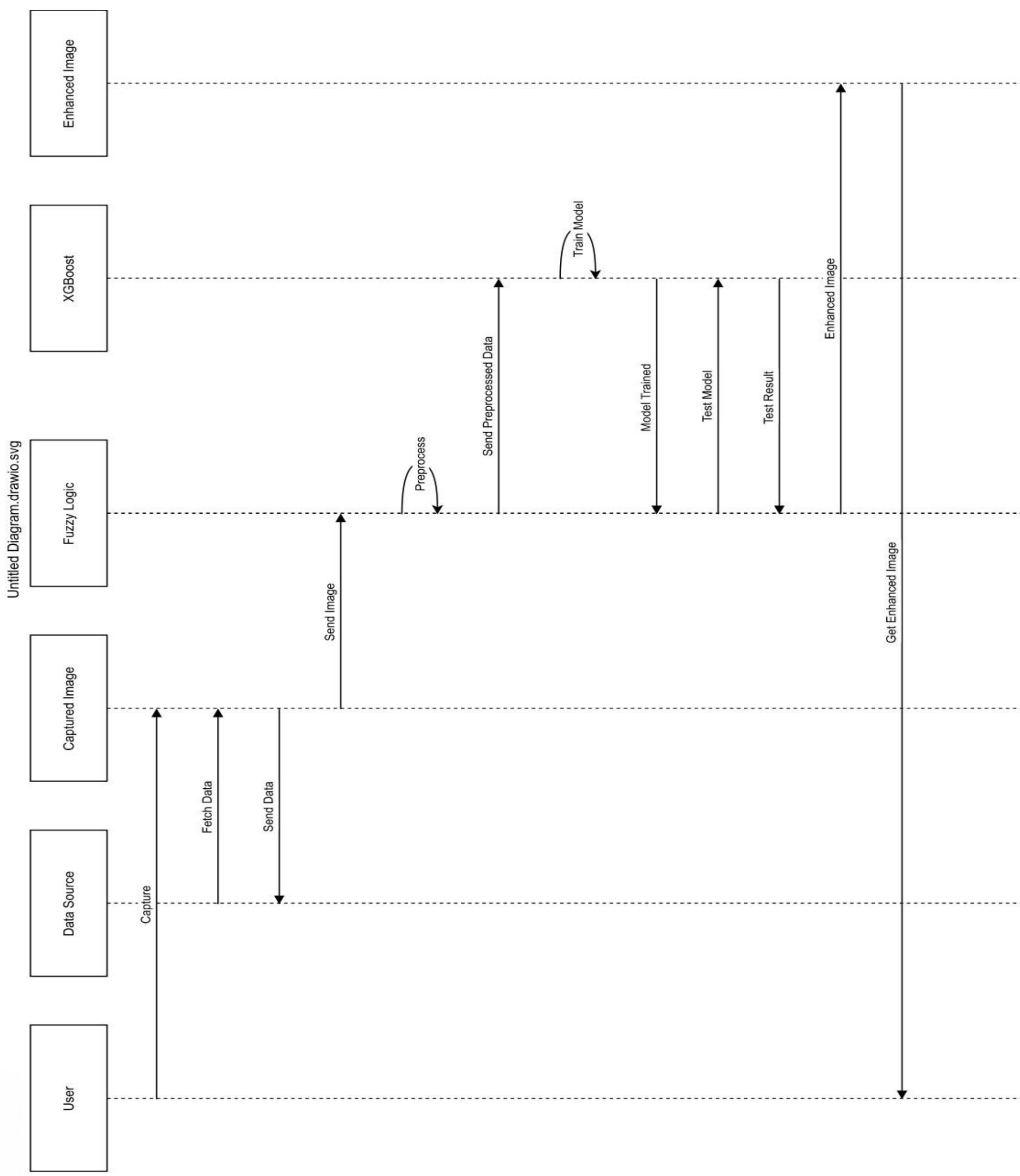
5. XGBoost: we use XGBoost to train and test the model.

6. Enhanced image: The user gets the enhanced image after processing.

Relationships: In the underwater image enhancement project, the classes are interconnected to facilitate a cohesive workflow from image processing to evaluation. The Data source class is for storing datasets for training the model, the Captured image class if the image captured and given as input by the user, the user class represents the user using the underwater image enhancement app, the Fuzzy logic class is for applying fuzzy logic and preprocess the image, the XGBoost class is the algorithm for model training, and gradient boosting and the Enhanced image class is the enhanced image obtained by the user through this app. This holistic approach ensures that underwater photographs are effectively enhanced and evaluated based on their visual appeal and technical quality, leveraging both empirical attenuation modeling and machine learning techniques.

SEQUENCE DIAGRAM

EXP.NO: 9 **DATE: 16/05/2024**



Sequence Diagram Explanation

The sequence diagram represents the interactions between various components in a system designed to process and enhance underwater images using fuzzy logic and XGBoost. Here is a detailed explanation of the actors, system components, and the sequence of events.

Actors and System Components

1. **User**: The individual who captures the underwater image and eventually receives the enhanced image.
2. **Data Source**: The collection of underwater image data used for training the model.
3. **Captured Image**: The underwater image captured by the user.
4. **Fuzzy Logic**: A component that preprocesses the data using fuzzy rules to handle the uncertainty and variability in the data.
5. **XGBoost**: An algorithm used to train and test the model on the preprocessed data.
6. **Enhanced Image**: The final processed image that the user receives after the enhancement process.

Sequence of Events

1.User Captures Image:

1. **User** captures an underwater image.
2. The captured image is represented by the **Captured Image** component.

2.Fetch Data from Data Source:

1. The **Data Source** fetches the necessary data for processing from its collection.
2. The **Data Source** sends this data to the **Captured Image** component.

3.Send Captured Image:

1. The **Captured Image** component processes the captured image.
2. It then sends the image to the **Fuzzy Logic** component for preprocessing.

4.Preprocess Image using Fuzzy Logic:

1. The **Fuzzy Logic** component applies fuzzy rules to preprocess the image.
2. After preprocessing, the **Fuzzy Logic** component sends the preprocessed data to the **XGBoost** component.

5.Train and Test Model using XGBoost:

1. The **XGBoost** component trains the model using the preprocessed data.

2. Once the model is trained, the **XGBoost** component tests the model.
3. The test results are then sent back to the **Fuzzy Logic** component.

6.Generate Enhanced Image:

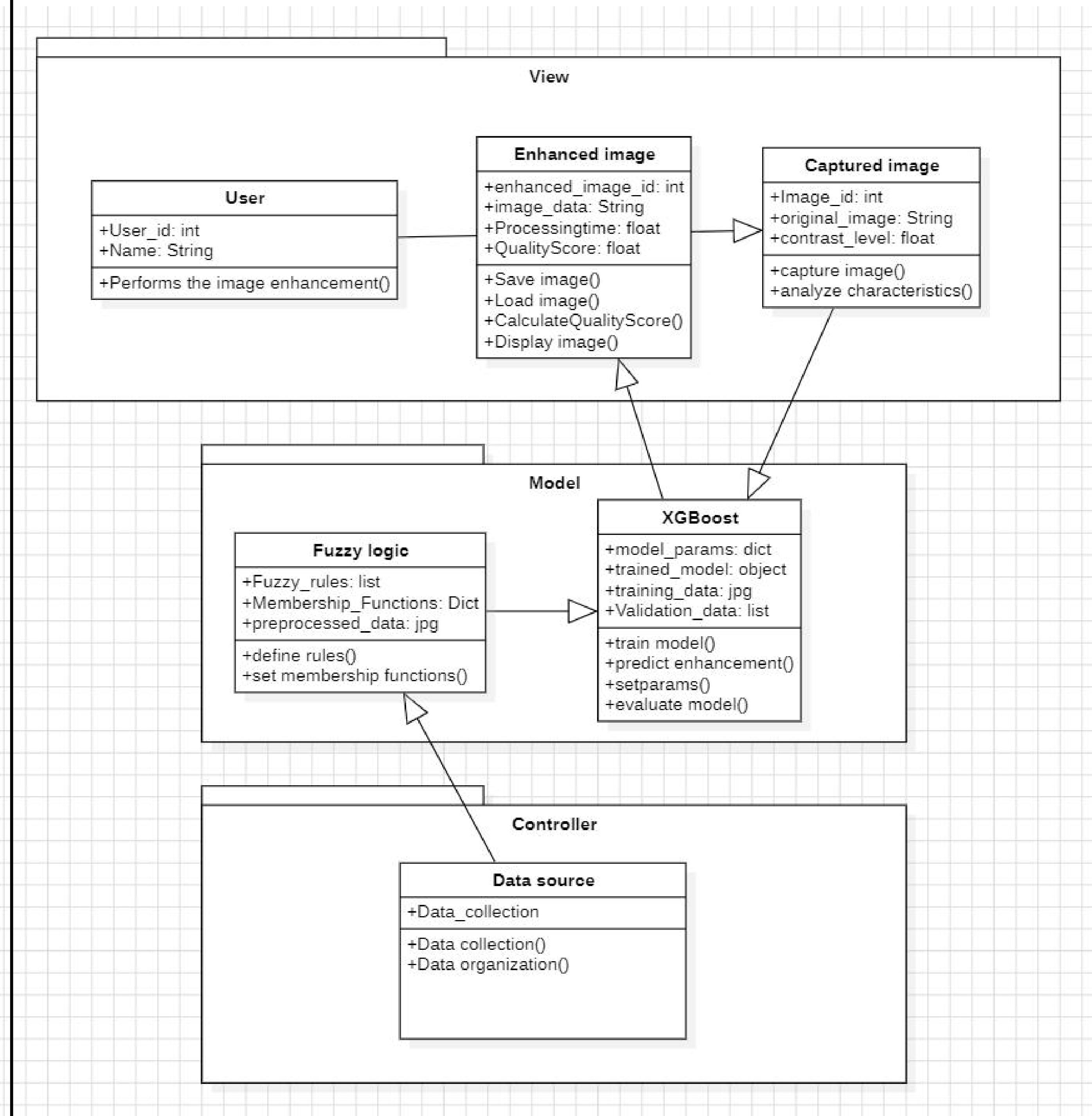
1. Based on the test results, the **Fuzzy Logic** component processes the data to create the enhanced image.
2. The enhanced image is then sent to the **Enhanced Image** component.

7.User Receives Enhanced Image:

1. The **Enhanced Image** component provides the enhanced image to the **User**.

ARCHITECTURAL PATTERN (MVC)

EXP.NO: 10 **DATE: 28/05/2024**



Model:

- **Fuzzy Logic:** Fuzzy sets can be defined to represent concepts like "low contrast," "high brightness," or "excessive noise." These sets wouldn't have strict boundaries but rather gradual transitions based on membership functions.Image features can be mapped to these fuzzy sets, providing a more nuanced representation of the image characteristics. For example, instead of a single value for brightness, the model might receive information about its degree of membership in "low," "medium," and "high" brightness sets.The XGBoost model can then learn from this richer feature representation to make more accurate enhancement predictions.

- **XGBoost:** The XGBoost model predicts the initial enhancement parameters.A fuzzy inference system takes these parameters and additional information (like user preferences) into account.The fuzzy system uses fuzzy rules to potentially refine the enhancement parameters based on the image characteristics and user preferences.

View:

- **User:** Buttons or options that allow the user to perform enhancement, save the enhanced image and adjust the enhancement parameters
- **Captured Image**: This could be displayed in an image viewer component where the user can see the original image they captured or loaded.
- **Enhanced image :** Another image viewer component could be dedicated to displaying the enhanced version of the image.It would ideally be positioned next to the captured image for easy comparison.

Controller:

- This acts as the intermediary between the Model and the View.
- It receives user input from the View (like clicking a button to capture an image) and interacts with the Model accordingly (like calling the Model to perform image enhancement).
- It then updates the View based on the Model's response (showing the enhanced image).

Relationships:

The Model, View, and Controller components interact as follows:

- The Model and View never directly interact. All communication goes through the Controller. This separation keeps the code cleaner and easier to maintain.
- The Controller interprets user input and translates it into actions for the Model. It also retrieves data from the Model and presents it to the View in a suitable format.
- Each part focuses on its own responsibility: the Model manages data, the View handles presentation, and the Controller coordinates interactions.