

A PRELIMINARY REPORT ON
"AN APPROACH FOR UNDERWATER IMAGE PROCESSING USING
RASPBERRY PI"

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE.

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

STUDENT NAME	EXAM SEAT NO.
1. Apurva Ingale	B190074205
2. Bhagyashree Kande	B190074232
3. Shivani Kulkarni	B190074241
4. Aditi Wankhade	B190074281



DEPARTMENT OF COMPUTER ENGINEERING

PVG's COLLEGE OF ENGINEERING AND TECHNOLOGY & G K PATE(WANI) INSTITUTE OF
MANAGEMENT

44, VIDYANAGARI, PARVATI, PUNE 411009

SAVITRIBAI PHULE PUNE UNIVERSITY

2022-23



CERTIFICATE

This is to certify that the Project Report entitled "**AN APPROACH FOR UNDERWATER IMAGE PROCESSING USING RASPBERRY PI**"

Submitted by

STUDENT NAME

EXAM SEAT NO.

1. Apurva Ingale	B190074205
2. Bhagyashree Kande	B190074232
3. Shivani Kulkarni	B190074241
4. Aditi Wankhade	B190074281

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of Prof. A.A.Junnarkar and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Engineering (Computer Engineering).

Prof.A.A.Junnarkar.

Guide

Department of Computer Engineering

Prof.D.D.Sapkal

Head.

Department of Computer Engineering

(Dr. M R Tarambale)

I/C Principal,

PVG's College of Engineering and Technology & GKPM, Pune – 09

Place: - Pune

Date:

ACKNOWLEDGEMENT

It is our privilege to acknowledge with deep sense of gratitude to our Project guide Prof.A.A.Junnarkar Ma'am for her valuable suggestions and guidance throughout our course of study and timely help given to us for this project.

We express our gratitude to Prof.Deepak Sapkal Sir (Head of Computer Engineering) for his kind help and co-operation. We are highly obliged to the entire staff of Computer Engineering Department for their kind co-operation and help.

Apurva Ingale	B190074205
Bhagyashree Kande	B190074232
Shivani Kulkarni	B190074241
Aditi Wankhade	B190074281

(B.E.Computer Engineering)

ABSTRACT

Over the last few years, researchers all over world have been studying about underwater image enhancement and the ability to obtain crystal clear image. Color correction and enhancement for underwater images is challenging due to scattering and low light intensity, underwater images always suffer from color castes and look bluish. The scattering and absorption of light in water leads to degradation of images captured under the water and makes the contrast relatively low. This degradation includes diminished colors, low brightness, and undistinguishable objects in the image. Crystal clear images can be resulted by contrasting RGB values. When the distance from the imaging scene to the camera is being increased, the red channel will disappear first. The red channel map is darkened, and the value of the pixels in the red channel becomes small. In this regard, the color of such image should be corrected. Inspired by the success of color transfer techniques, it is found that clear template image-assisted color transfer is a promising strategy for underwater image enhancement, including not only color correction but also contrast and visibility improvement. Our proposed method is very simple and straightforward that contributes greatly in uplifting the visibility of underwater images. To improve the quality of degraded images, we have proposed fusion based underwater image enhancement technique that focuses on improving of color contrast of the image underwater using raspberry pi, our camera model will be trained in such a way that the captured input image will add the required resolutions and effects to the image. Thus, our model will result the underwater images.

INDEX

Sr.No.	Chapter Name	Page No.
01	Introduction	1
	1.1 Overview	1
	1.2 Motivation	1
	1.3 Problem Definition and Objectives	1
	1.4 Project Scope and Limitations	1
	1.5 Methodologies of Problem Solving	1
	1.5.1 Superior underwater colour cast neutralization	3
	1.5.2 Fusion of dual-intensity pictures based on the mean and median values	3
	1.5.3 Swarm-intelligence based mean equalization	3
	1.5.4 Unsharp masking	3
02	Literature Survey	3
	2.1 Reference Papers	3
	2.1.1 Color Correction Based on CFA and Enhancement Based on Retinex with Dense Pixels for Underwater Images	3
	2.1.2 Fast Underwater Image Enhancement for Improved Visual Perception	4
	2.1.3 Simultaneous Enhancement and Super-Resolution of Underwater Imagery for Improved Visual Perception	5
	2.1.4 An approach for underwater image enhancement based on color correction and dehazing	6

03	Software Requirement Specification		9		
	3.1	Assumptions and Dependencies		9	
	3.2	Functional Requirements		10	
		3.2.1	Camera Interfaces		10
		3.2.2	Algorithms		10
	3.3	External Interface		11	
		3.3.1	User Hardware Interface		11
	3.4	Nonfunctional Requirements		15	
		3.4.1	Performance Requirements		15
		3.4.2	Security Requirements		15
		3.4.3	Software Quality Attributes		15
	3.5	System Requirements		16	
		3.5.1	Database Requirements		16
		3.5.2	Software Requirements (Platform choice)		16
		3.5.3	Hardware Requirements		16
	3.6	Analysis Models: SDLC Model to be applied		17	
04	System Design		20		
	4.1	System Architecture		20	

	4.2	Mathematical Model	25
	4.3	Data Flow Diagram	25
	4.4	Entity Relationship Diagram	25
	4.5	UML Diagrams	27
	4.5.1	Use Case Diagram	27
	4.5.2	Sequence Diagram	28
	4.5.3	Activity Diagram	29
05	PROJECT PLAN		30
	5.1	Project Estimate	9
	5.1.1	Reconciled Estimates	29
	5.1.2	Project Resources	29
	5.2	Risk Management	9
	5.2.1	Risk Identification	29
	5.2.2	Risk Analysis	29
	5.3	Project Schedule	9
	5.3.1	Project Task Set	29
	5.3.2	Task Network	29
	5.3.3	Timeline Chart	29

	5.4	Team Organization	9
	5.4.1	Team Structure	29
06	Project Implementation		31
	6.1	Overview of project module	31
	6.2	Tools and Technologies use	32
	6.2.1	Raspbian OS	29
	6.2.2	VNC Viewer	29
	6.2.3	Python Thonny IDE	29
	6.2.4	Django	29
	6.3	Algorithm Details	32
	6.3.1	NUCE	29
	6.3.2	Histogram Equalization	29
07	Software Testing		31
	7.1	Types of Testing	32
	7.1.1	Functional Testing	29
	7.1.2	Non Functional Testing	29
	7.2	Test Cases and Test Results	32
08	Results		31
	8.1	Outcomes	32

	8.2	Screenshots	32
	8.3	Results by Comparing Pre existing models	32
09		Conclusions	31
	9.1	Conclusion	32
	9.2	Future Scope	32
	9.3	Applications	32
		Appendix A	33
		Appendix B	34
		Appendix C: Plagiarism Report	35
		References	36

LIST OF ABBREVIATIONS

ABBREVIATION	ILLUSTRATION
CFA	Color Filter Array
MR	McCann Retinex
WB	White Balance
DCP	Dark Channel Prior
UIEB	Underwater Image Enhancement Benchmark

LIST OF FIGURES

Figure No	Figure Name	Page No
1.1	Problem Definition	2
3.1	Raspberry Pi	11
3.2	Camera	12
3.3	HQ Camera Module	13
3.4	Camera Setup	14
3.5	Final Camera Module with Raspberry Pi	14
3.6	Incremental Model	17
3.7	Gantt Chart	19
4.1	System Architecture	20
4.2	Color Filter Array	21
4.3	Gamma Correction	22
4.4	White Balancing	23
4.5	Color Temperatures	24
4.6	0 Level DFD	25
4.7	1 Level DFD	26
4.8	Use Case Diagram	27
4.9	Sequence Diagram	28
4.10	Activity Diagram	29

1 INTRODUCTION

1.1 Overview

The problem of underwater image enhancement and colour correction is difficult and has become more important recently. The colour contrast of images captured underwater cannot be distinguished or noticed by the human eye. Using expensive equipment, we may capture underwater images that are bluish and greenish in colour because more depth prevents light from reaching its full strength. Since there is less light intensity at a certain depth, the camera is unable to capture images using all three RCB values, which results in bluish or greenish images. Underwater images suffer from significant colour distortion and the dominance of a heavy blue colour. In order to improve these images, we created a system using IOT (Raspberry Pi) in which images clicked under water are improved and saved in a new folder. When compared to previous underwater enhancement techniques, our technique produces more natural underwater images.

In our system, images are taken underwater and saved in the input image folder on the local device. When the Python script is executed, all of the captured images are processed simultaneously. We have used four methods to improve the images: 1. Neutralizing image cast 2. Histogram equalization 3. Swarm intelligence 4. Unsharp masking These techniques improve underwater images and result in a natural underwater image. The improved images are saved on the local device in the result folder.

As it concentrates on enhancing the contrast and colour of underwater images through contrast stretching, the results imply that it enhances the quality of damaged underwater images. It employs the sharpening technique to bring out minute details in an image or to increase detail that has been lost due to low contrast and high scattering of light. Therefore, images generated by our technology are natural underwater images.

1.2 Motivation

The ocean contains various abundant resources. However, it has not been effectively explored and exploited by humans, especially in the underwater world. Underwater image processing plays an indispensable role in underwater operations by humans or underwater robots. Underwater image enhancement is an active research problem that deals with correcting image distortions to recover true pixel intensities

1.3 Problem Definition and Objectives

Underwater image enhancement and colour correction are challenging tasks that have gained priority in recent years. The colour contrast of underwater images cannot be distinguished or detected with clarity by the human eye. We get bluish and greenish underwater images while using expensive equipment since the intensity of light cannot be attained at higher depths. In contrast to all previous methodologies, our proposed model makes use of a hardware-based Raspberry Pi model, and the camera model will be trained so that underwater image reconstruction is not required because it will clear the images as soon as the image is captured. Thus, our problem description reads: A method for using a Raspberry Pi to improve the colour of underwater images.

1.4 Project Scope and Limitations

The proposed model captures an image as the input and a sequence of operations such as naturalizing image cast, histogram equalization, swarm intelligence method, unsharp masking, etc. are performed on the input image. Finally, multi-scale images The fusion of the inputs is done to obtain the resultant enhanced image as an output. This will result in clicking images underwater which is believed to be a tedious task given the extreme absorption and scattering effects of the medium. Therefore, image improvement techniques are applied to get rid of image noise and improve overall colour correction of an image.

Cost of hardware components. The model requires a Raspberry Pi and camera module, which are expensive. As the model is placed under the water to capture images, hardware components need to be protected from water and pressure.

1.5 Methodologies of Problem Solving

Four Strategy we have followed in our model:

1.5.1 Superior underwater colour cast neutralization

This is the initial action. The first stage presents a novel strategy to remove the underwater colour cast. It introduces a new approach to neutralize underwater color cast. Before color cast neutralization color channels are decomposed.the Natural Underwater Colour Enhancement approach lessens the underwater colour cast. By considering the variations between superior and inferior colour channels, these gain factors are determined.

1.5.2 Fusion of dual-intensity pictures based on the mean and median values

Next the idea is to construct lower-stretched and upper-stretched histograms in the second stage using the dual-intensity images fused based on the average of mean and median values. The relationship between these histograms greatly enhances the visual contrast. Following the computation of the picture histogram's minimum, maximum, mean, and median values, the average point between the mean and median values is computed.

1.5.3 Swarm-intelligence based mean equalization

Then, a mean equalization based on swarm intelligence is suggested to enhance the output image's naturalness. The mean values of inferior colour channels are modified to be close to the mean value of superior colour channels through the fusion of swarm intelligence algorithms. Based on mean values, colour channels are classified as superior or inferior. Each pixel's fitness is assessed, and adjustments to its velocity and location are made until mean equality is achieved.

1.5.4 Unsharp masking

Lastly, the unsharp masking technique is applied to sharpen the overall image. Unsharp masking is the process of making a slightly fuzzy exposure of the original negative onto a brand-new roll of film. The unsharp mask filter method involves subtracting an unsharp mask from the sample image. An unsharp mask only produces a hazy image by spatially filtering the specimen image with a Gaussian low-pass filter.

2 LITERATURE SURVEY

2.1 Reference Papers

2.1.1 Color Correction Based on CFA and Enhancement Based on Retinex with Dense Pixels for Underwater Images

Author : Changli Li, Shiqiang Tang

Published Year : 2020

Platform : IEEE Access

This paper proposed two methods:

1. Underwater image color correction method based on color filter array (CFA) .
2. Underwater image enhancement method based on retinex theory.

This paper dealt with underwater image colour correction and underwater image enhancement methods. A colour distortion correction is made for the red channel from the other two channels (blue and green), as for any RGB image captured by a camera with a colour filter array, its RGB values are dependent. The linear function for adaptive histograms is used to improve the visual quality of the whole image. The result shows that the proposed method gives clearer, more uniform visual effects.

2.1.2 Fast Underwater Image Enhancement for Improved Visual Perception

Author : Md Jahidul Islam , Youya Xia

Published Year : 2020

In his paper, he proposed another methodology for improving underwater image quality based on the generative adversarial network (GAN) model. Using information about an image's global colour, local texture, and style, this approach creates a perceptual loss function. In addition to ensuring substantially faster inference times, it conducts extensive qualitative and quantitative assessments and user studies to demonstrate that the suggested model performs as well as and frequently better than state-of-the-art models. That shows how well it works to enhance the performance of underwater object detection and human body pose estimation.

2.1.3 Simultaneous Enhancement and Super-Resolution of Underwater Imagery for Improved Visual Perception

Author : Peigen Luo and Junaed Sattar

Published Year : 2020

In this paper, the simultaneous enhancement and super-resolution (SESR) problem is described and a productive learning-based underwater imaging solution is presented. It presents its comprehensive training process, accompanying loss functions, and thorough network architecture. It carries out a number of qualitative and quantitative experiments that indicate Deep SESR: (i) offers SOTA performance on underwater image enhancement and super-resolution; (ii) exhibits significantly better generalization performance on natural images than existing solutions; (iii) provides competitive results on terrestrial images; and (iv) achieves quick inference on single-board platforms. Deep SESR is appropriate for usage in close to real time by visually guided underwater robots due to its impressive performance, computational efficiency, and accessibility of application-specific design options.

2.1.4 An approach for underwater image enhancement based on color correction and de-hazing.

Author : Yue Zhang

Published Year : 2020

Platform : International Journal of Advanced Robotic Systems.

This paper has an improved approach for eliminating the local reddish effect and reducing image noise. As when light propagates in water, water medium ,water particles and scatter light will absorb. Because of this underwater images present defects such as color deviation, low contrast, and blurry details. This paper includes two algorithms for color correction of underwater images.

1) White Balance Algorithm (WB): The main problem of underwater images are greenish-blue appearance due to scattering of waves when depth is increasing. As higher wavelength waves get absorbed first. So red will be absorbed first. Red channel is degraded first when it passes through water and green channel is almost safe because of its shorter wavelength compare to red channel. So, this white balance method mainly focuses on restore the colors which are degraded due to absorption of white light propagate through water.

2) Dark Channel Prior Algorithm (DCP): Dark channel prior algorithm is mainly proposed for image dehazing. But directly applying the DCP algorithm to underwater images does not provide a good enhancement effect. The dark channel value obtained based on the minimization and operation is likely to be the red channel component in the dark channel calculation process, which leads to a dark image after restoration.

3 SOFTWARE REQUIREMENT SPECIFICATION

3.1 Assumptions and Dependencies

- Our model depends on underwater image enhancement techniques where we have obtained different training testing dataset from existing models.
- Our model also depends on raspberry pi for training the Machine Learning Model

3.2 Functional Requirements

3.2.1 Camera Interface

Camera is connected to the Raspberry pi and trained to capture and correct underwater images. Divers can capture the underwater images using camera interface.

3.2.2 Algorithms

We have used various algorithms such as NUCE algorithm and Histogram Equalization.

3.3 External Interface

3.3.1 User Hardware Interface

- Raspberry Pi

Raspberry Pi 4 Model B is the latest product in the popular Raspberry Pi range of computers. It offers ground-breaking increases in processor speed, multimedia performance, memory, and connectivity compared to the prior-generation Raspberry Pi 3 Model B+, while retaining backwards compatibility and similar power consumption. For the end user, Raspberry Pi 4 Model B provides desktop performance comparable to entry-level x86 PC systems.

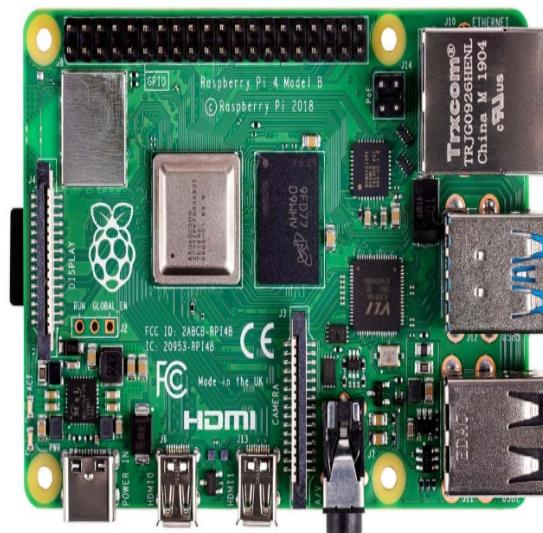


Figure 3.1: Raspberry Pi

- Camera

Pi Camera module is a camera which can be used to take pictures and high-definition video. Raspberry Pi Board has CSI (Camera Serial Interface) interface to which we can attach Pi-camera module directly. This Pi Camera module can attach to the Raspberry Pi's CSI port using 15-pin ribbon cable.



Figure 3.2: Camera

- 4k HD Camera module

4K Action Camera with 4X Zoom: Professional 4K/30FPS, 2.7K/30fps, 1080P/60FPS video and 16MP photo resolution enables you to capture exciting moment for you. This camera also features zoom range from 1.0X to 4.0X.6mm CS mount lenses and 16mm with C-mount are examples of all the compatible ones that exist. The high-quality camera offers an alternative to the camera module v2. 4k HD action camera features multiple functions including driving mode, image rotation, time-lapse, burst photo, loop recording, slow motion, dramashot, white balance, etc. Bringing this camera to more applicable conditions beyond your expectation.



Figure 3.3: 4K HD Camera Module

- Camera setup

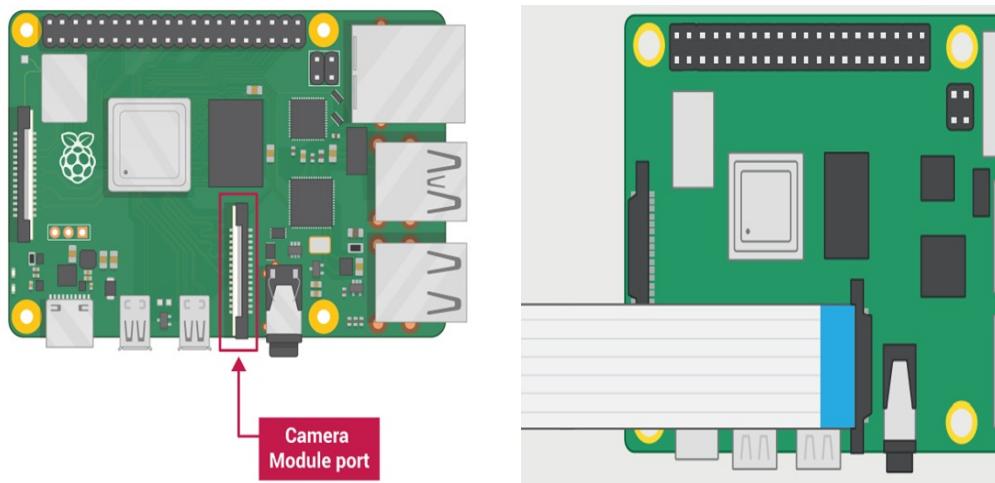


Figure 3.4: Camera Setup

- Final Camera Module with Raspberry Pi



Figure 3.5: Final Camera Module with Raspberry Pi

3.4 Nonfunctional Requirements

3.4.1 Performance Requirements

Our project will need a Raspberry pi that is connected with camera module on where our proposed methodologies performance the execution one phase at a time all these methods are deployed on raspberry pi. This is necessary in order to click underwater image to get corrected image.

3.4.2 Security Requirements

Our module is compatible with all types of cameras such as pi camera, waterproof cameras and mobile cameras. User will be able to click underwater image with different mega pixel cameras. Our methods are compatible for all types of camera specifications.

3.4.3 Software Quality Attributes

Our model design in such a way that it will be helpful for underwater researchers and scuba divers for clicking underwater images and get corrected image under the water itself.

3.5 System Requirements

3.5.1 Database Requirements

We do not require any database, the output/resultant images are stored locally in the folder

3.5.2 Software Requirements (Platform choice)

- Operating System: Windows / Linux
- IDE: Jupyter Notebook, VS Code
- ML tools: TensorFlow, Keras, scikit-learn
- Data Handling and Visualization: Pandas

3.5.3 Hardware Requirements

- Processor: Intel Core i5 or i7 (recommended)
- RAM: Minimum 8 GB (recommended)
- GPU: An NVIDIA GPU with CUDA support and at least 12 GB of VRAM
- Raspberry Pi 3 Model B
- Camera 12.3 Megapixel or more
- SD card (32 GB min.)

3.6 Analysis Models: SDLC Model to be applied

Incremental Model

Incremental Model is a software development procedure where requirements are split up into several independent software development cycle modules. Each module in this model undergoes the phases of requirements, design, implementation, and testing. The module's functionality is increased with each new release. Up till the full system is achieved, the process is continued.

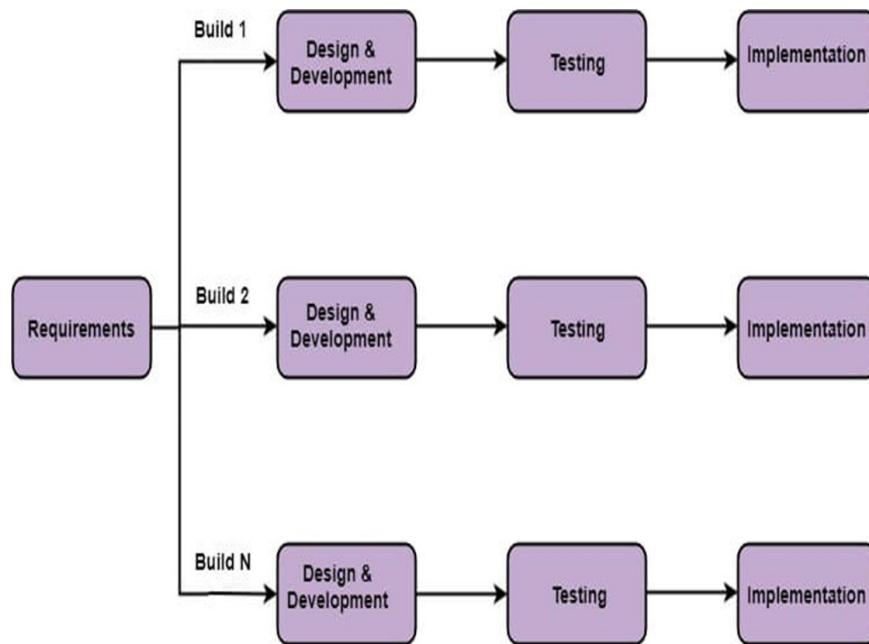


Figure 3.6: Incremental Model

The incremental model's many phases are as follows:

1. Requirement analysis: The product analysis expertise identifies the requirements in the first phase of the incremental model. The requirement analysis team is also aware of the functional needs for the system. This phase is extremely important for the incremental paradigm of software development.
2. Design Development: The design of the system functionality and the development methodology are successfully completed in this phase of the incremental model of SDLC. The incremental model

uses style and development phase when software acquires new practicality.

3. Testing: In the incremental model, the testing phase evaluates the effectiveness of every function that already exists as well as newly added functionality. The numerous techniques are utilised to test each task's behaviour during the testing process.

4. Implementation: The development system's coding phase is made possible by the implementation phase. In the planning and developing phase, it involves the final coding, and in the testing phase, it involves functional testing. Following the conclusion of this phase, the number of working products is updated and improved all the way up to the finished system product.

Advantages:

- 1) Errors are easy to be recognized
- 2) Easier to test and debug
- 3) More flexible
- 4) The Client gets important functionality early

Disadvantages:

- 1) Need for good planning
- 2) Total Cost is high
- 3) Well defined module interfaces are needed

4 SYSTEM DESIGN

4.1 System Architecture

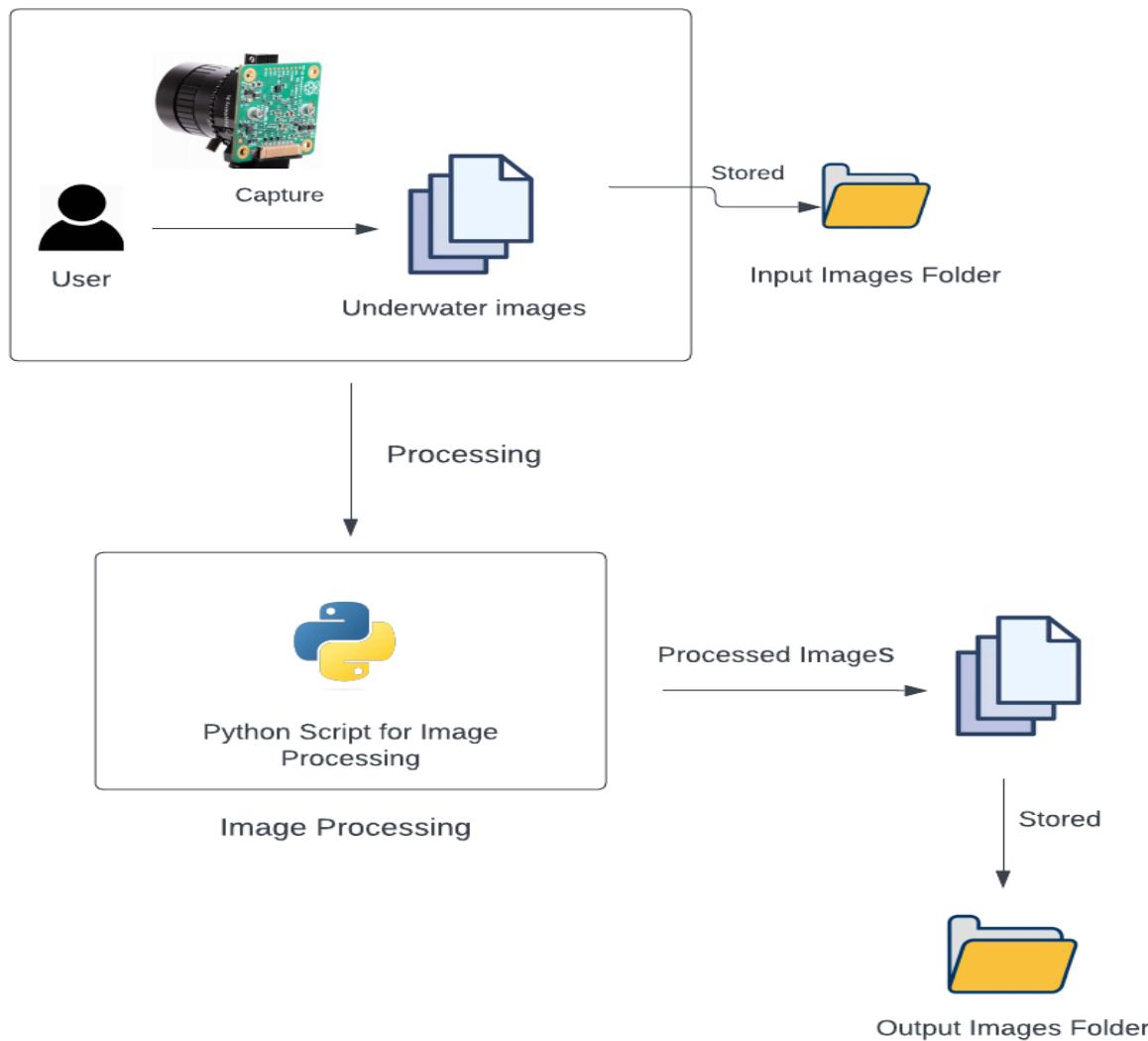


Figure 4.1: System Architecture

4.2 Mathematical Model

4.3 Data Flow Diagram

A data flow diagram demonstrates the path that information takes through a system or process. It contains the numerous sub processes the data flows through, data inputs and outputs, and data stores. Standardised symbols and terminology are used to construct DFDs, which define numerous entities and their relationships.

1. DFD Level 0 –

Context Diagrams are another name for DFD Level 0. It provides a general summary of the entire system or process that is being studied or modelled. The system is displayed as a single, high-level process, together with its relationship to external entities, in what is intended to be an overview view.

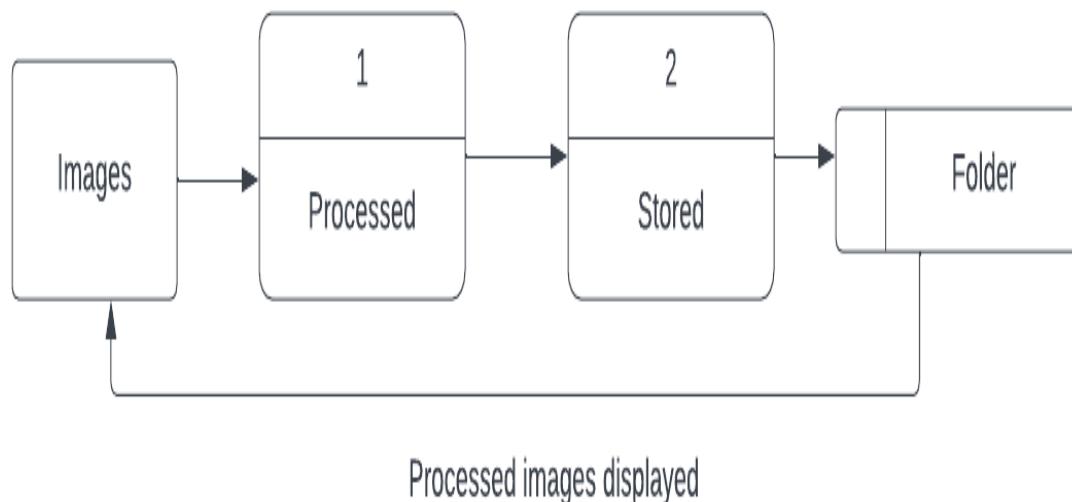


Figure 4.2: 0 Level DFD

2. DFD Level 1 –

While still providing a broad overview, Level 1 DFDs delve into greater detail than a context diagram. The single process node from the context diagram is divided into sub-processes in level 1 DFD. The diagram will require more data flows and data stores when new processes are introduced to connect them.

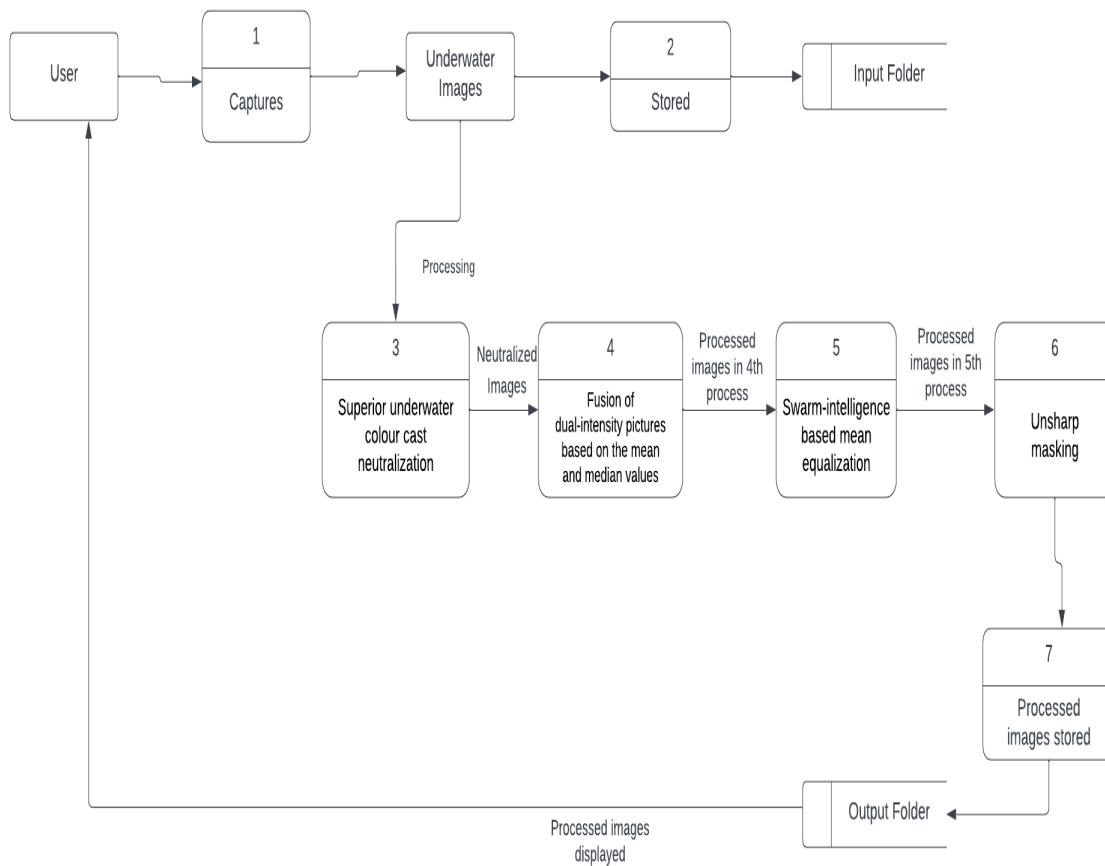


Figure 4.3: 1 Level DFD

4.4 Entity Relationship Diagram

Entity-Relationship model is referred to as an ER model. This data model is on a high level. The data items and relationships for a given system are defined using this model. It creates the database's conceptual design. Additionally, it creates a very straightforward and straightforward data view.

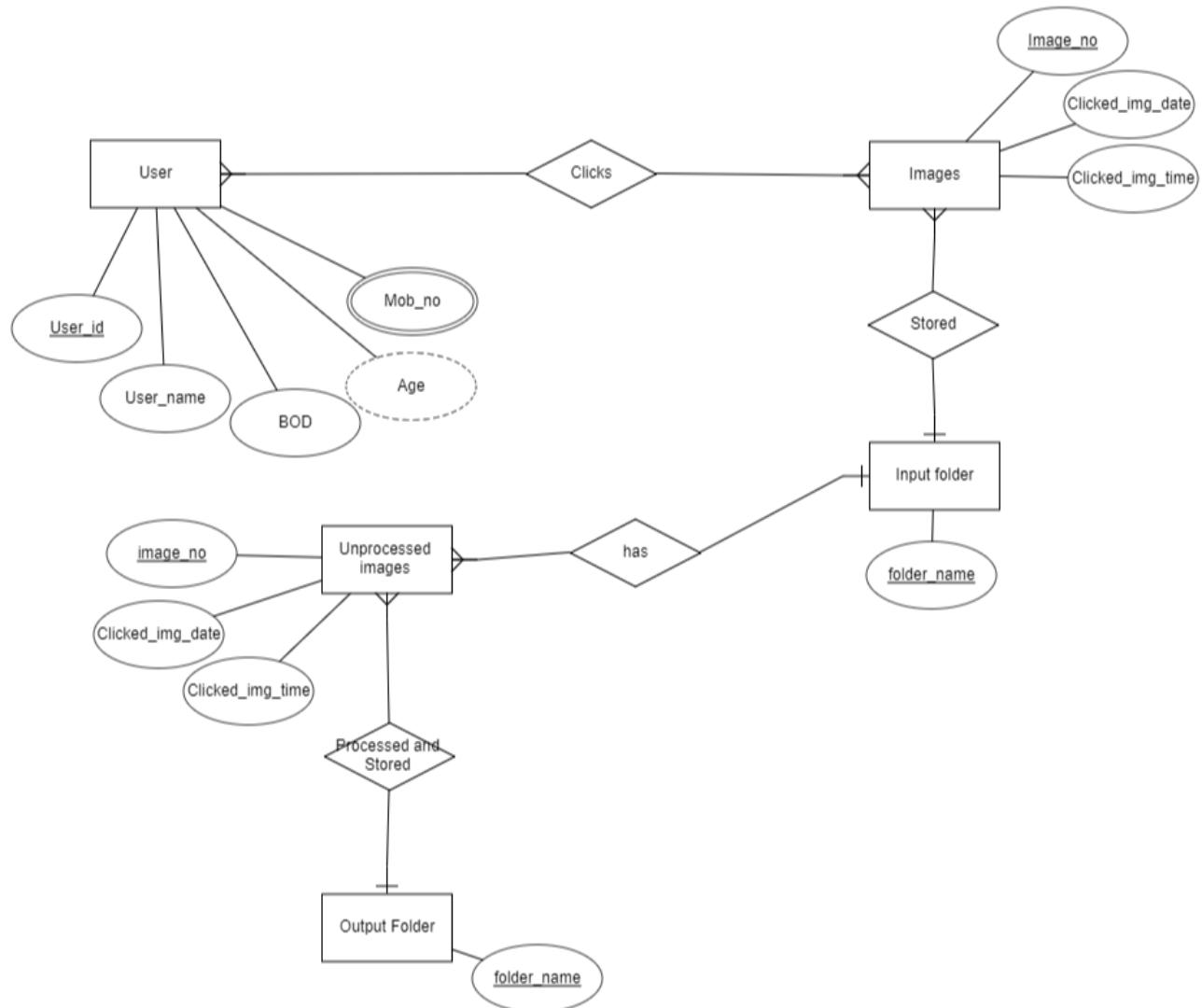


Figure 4.4: ER Diagram

4.5 UML Diagrams

For the development of software, the software industry uses various well-defined process models. The Software Engineering Process (SEP) or Software Development Process (SDP) is the name of this model. The Unified Software Development Process (USDP) is a widely used and commonly applied SDP in the industry. It is also known as the Unified Process (UP) in short. Unified Process (UP) deals with the process side of software development, while Unified Modelling Language (UML), a virtual language part of the system or project, coexists. The standard modelling notation and language for object-oriented analysis and design is UML. These notations, which are widely acknowledged by numerous software development organizations, are used to visualize system architecture.

4.5.1 Use Case Diagram

Use-case diagrams describe how a user interacts with a system. As a series of interactions between the actor and the system, it is depicted. Both written and visual descriptions of use cases are provided. The system is presented as the user would see it. Use cases do not emphasize the system's internal operation or processing. Three system components are displayed in the use case diagram:

- **Actors:** Human-shaped sticks are used to represent actors. It might not always be a representation of the user. It can be used to represent a user, another system, or a device that communicates with a specific high-level feature (use case) of the system.
- **Use case:** Use cases essentially represent the system's top-level functions. An ellipse is used to symbolize a use case.
- **System boundary:** The system boundary is a container that holds use cases. Actors are beyond the queue. System boundaries are used to illustrate how a system's functionality is contained.

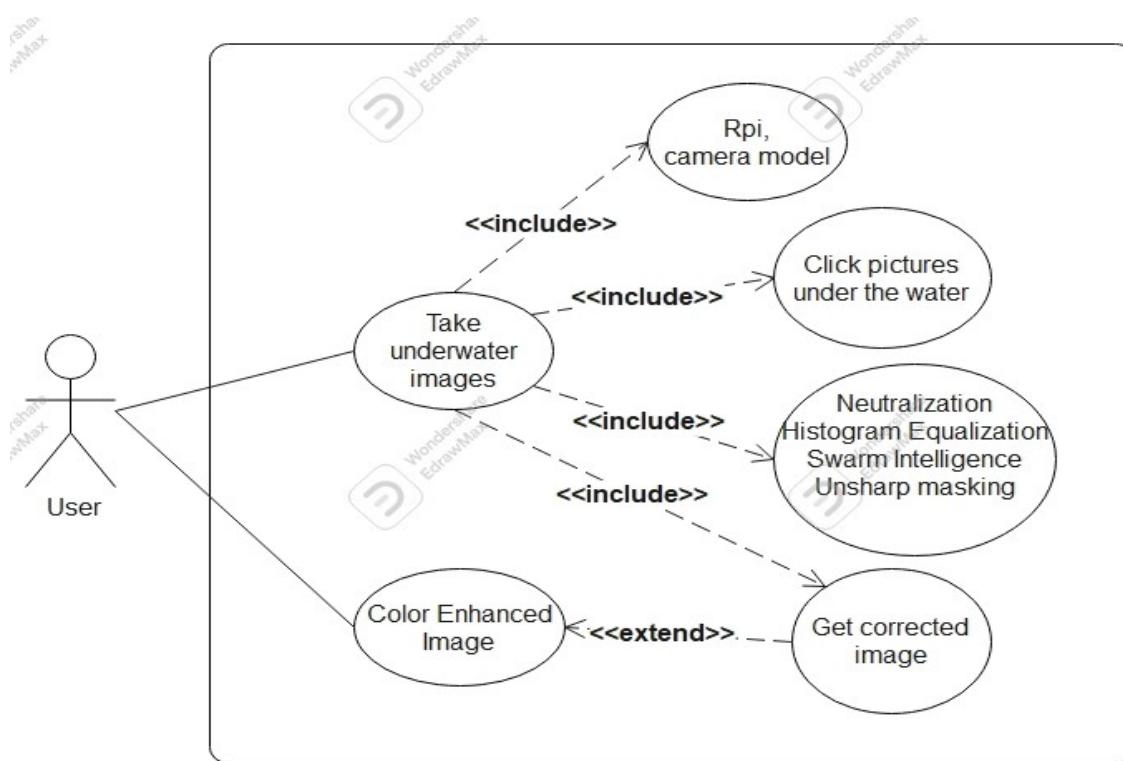


Figure 4.5: Use Case Diagram

4.5.2 Sequence Diagram

Sequence diagrams are used to display how objects interact once a system is operating. The order of events is the sequence diagram's main concern. The timeline shows events as they happen from top to bottom in time. The item lifeline, a very distinctive element in the diagram, is used to represent how an object interacts with other objects in relation to its lifeline. The dashed line below the object instance is used to represent the instance lifeline. The point of control in a sequence diagram switches from one object to another as one object activates the other. The duration of the instance across time is indicated by the complete line.

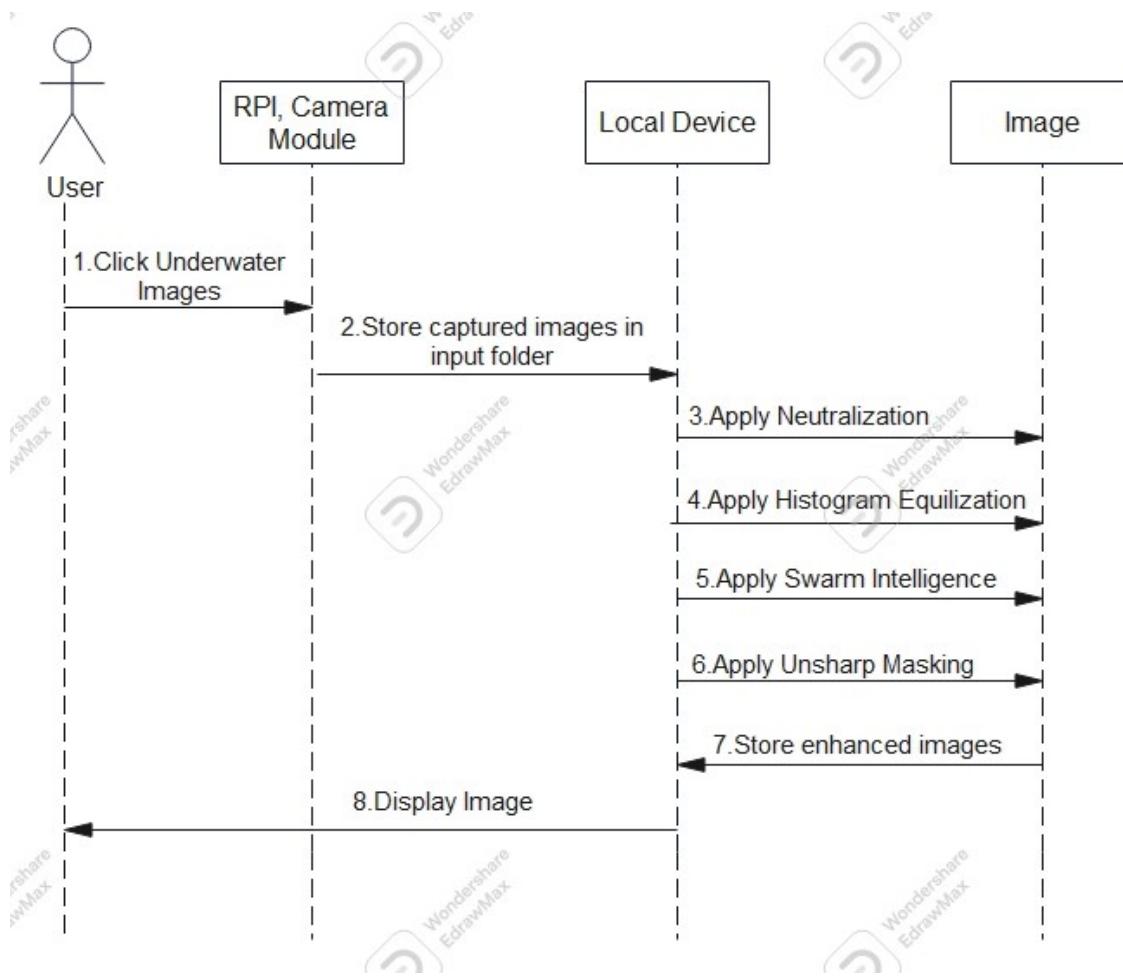


Figure 4.6: Sequence Diagram

4.5.3 Activity Diagram

Activity diagrams are used to represent flow of different activities in the system. It shows the flow of control and sequencing among different possible activities. Like use case, activity diagram also captures the user's perspective of the system. Activity diagrams do not explore system design or system's processing logic, instead focuses on flow of activities that a user of the system can experience. Activity diagram are mostly used for work flow modelling in Web Services and SOA (Service-Oriented Architecture) applications.

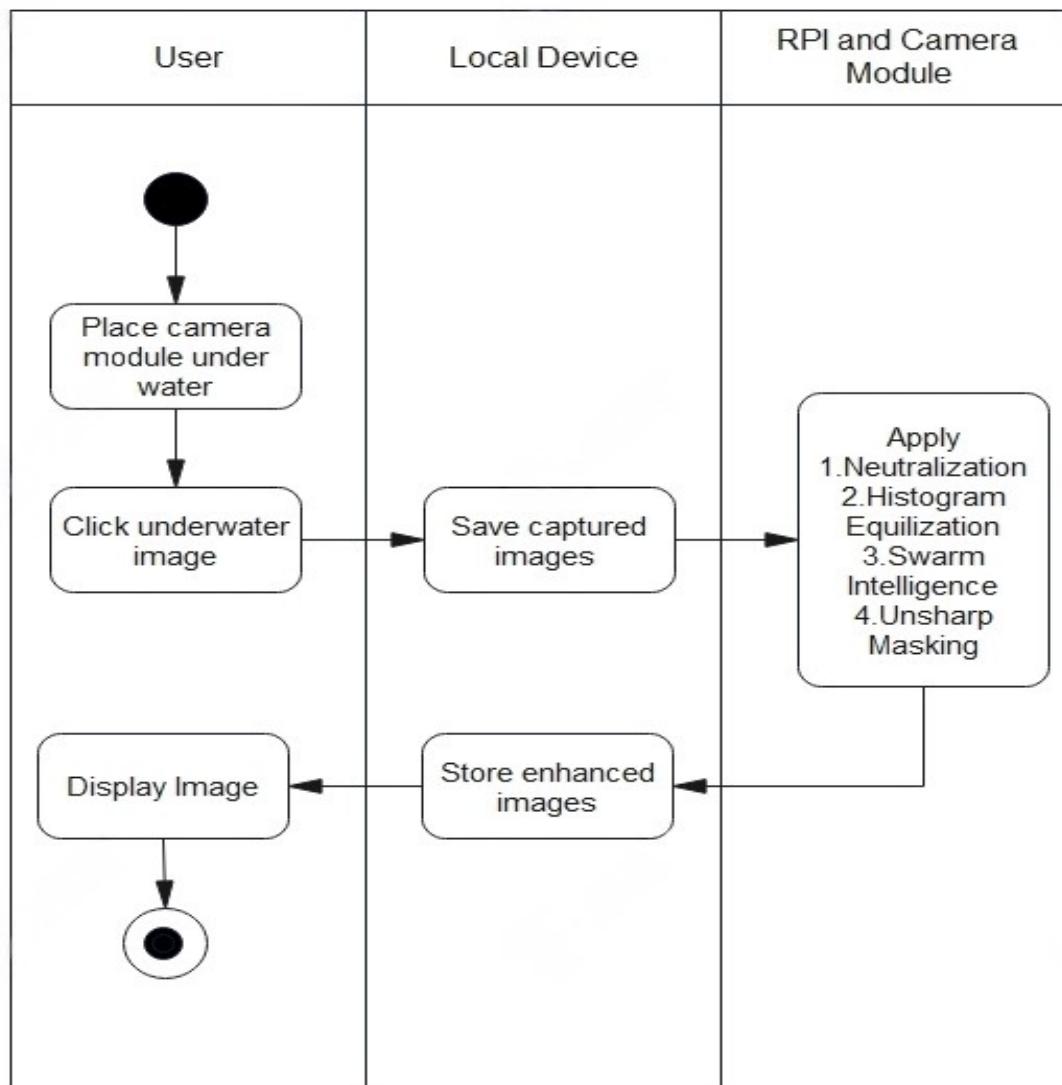


Figure 4.7: Activity Diagram

5 PROJECT PLAN

5.1 Project Estimate

Sr. No	Software & Hardware Components	Cost
1	Raspberry Pi 3B or higher version	5000+/-
2	Camera	- Pi camera (5 MP) 400-500 - Waterproof Camera (4K 16 MP) 2000/-
3	HDMI Cable	180/-
4	Sd Card	500/-
5	Our Proposed software cost	6000/- (approx.)
6	Paper Publishing	8K-10K(approx.)
	Total =	7680 (excluding software cost & paper publishing cost)

Table 5.1: Project Estimation

5.1.1 Reconciled Estimates

- Cost Estimate

There was no expenditure done for the project. The required data is made freely available by using EUVP Dataset. The entire technology stack used by us is free and open source. Training of the Machine Learning models was done using free GPU access provided by Raspbian OS.

- Time Estimate

The time required to complete the project is dominated by the data preprocessing step. For researching existing systems, it took about 2 months. It required about 5-6 months of work

and the modelling phase required 2-3 months. And for the UI development around 5-10 days. Total we developed whole system in 10-11 months from initial stage to working model.

5.1.2 Project Resources

- Hardware Resources
 - Storage: 4 GB (min.)
 - Memory: 8 GB (min.)
- Software Resources
 - Linux Operating System
 - Python 3.8 Programming Language
 - Google Chrome Web Browser
 - Python Thonny IDE

5.2 Risk Management

5.2.1 Risk Identification

The project risks were identified in the design as well as review phase. The answers to the following questions helped in this process:

1) Are the end-users enthusiastically committed to the system built?

Yes, this project caters to a very niche audience –divers researchers who wish to study underwater or marine life and are looking to automate their workflow. This project helps them achieve the same.

2) Are requirements fully understood by the software development team?

Yes, the software development team has understood all the requirements.

3) Are the project requirements stable?

Yes, the project requirements were identified completely and finalized in the requirements phase.

4) Is the number of people on the project adequate to do the job?

Yes, according to the project scope, the number of people on the development team are adequate to develop the software

5.2.2 Risk Analysis

Technical Risks

1) User data might leak.

Mitigation: User data such as uploaded spectra is not stored at all.

2) Data must be uploaded in any image format.

Mitigation: Both the leading data formats jpeg/jpg and png are supported.

3) A large dataset might not fit in memory during the preprocessing and modelling phase.

Mitigation: Data processing and training is performed in batches

4) Model results might be inaccurate.

Mitigation: Extensive modelling has been done to ensure that the classification results are near-perfect. Accuracy can be identified by analysing the histogram of images for comparing the results.

5.3 Project Schedule

5.3.1 Project Task Set

Major Tasks in the Project stages are:

- Task 1: Requirement Analysis.
- Task 2: Learning Astronomy.
- Task 3: Technology Study and Design.
- Task 4: Data Preprocessing.
- Task 5: Coding and Implementation.
- Task 6: Testing and Documentation.

5.3.2 Task Network

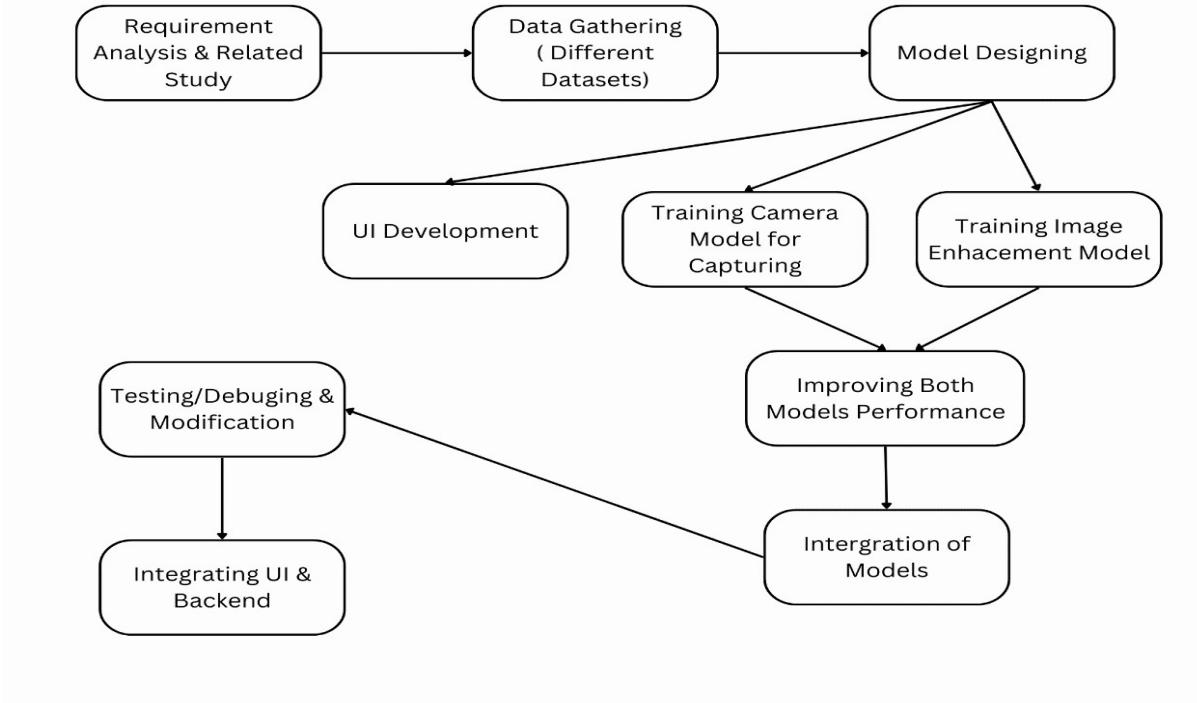


Figure 5.1: Task Network

5.3.3 TimeLine Chart

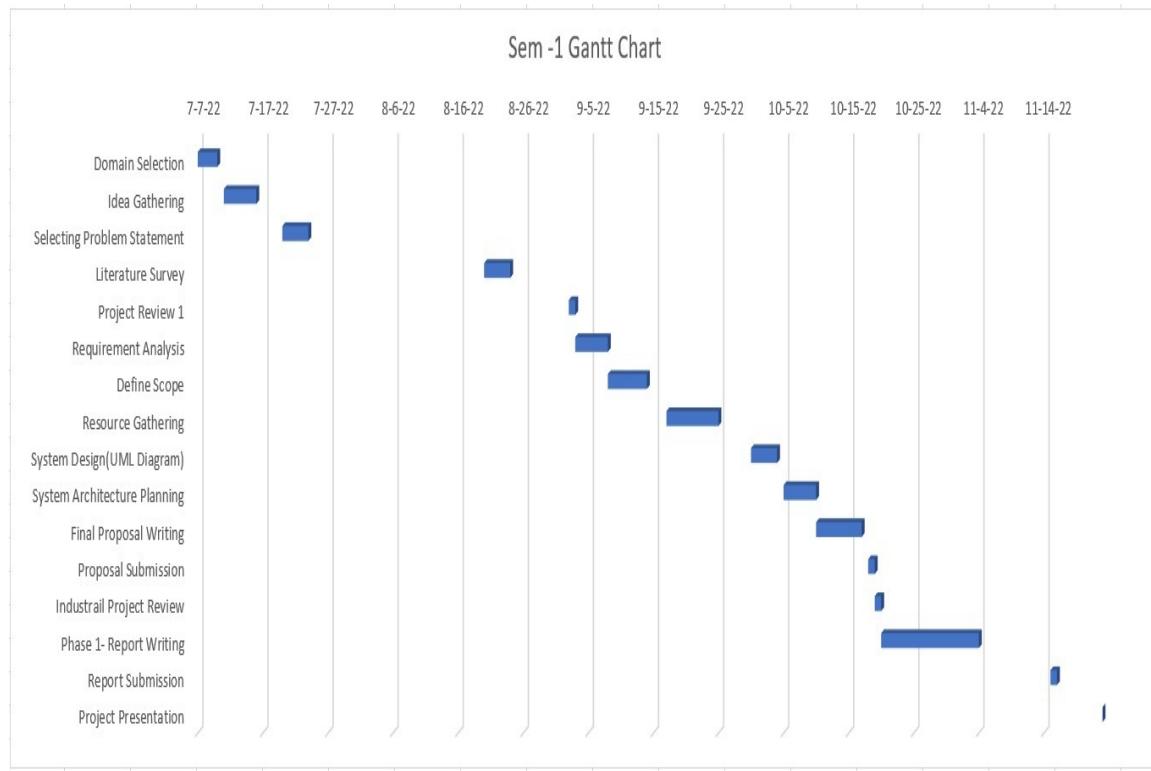


Figure 5.2: Gantt Chart for Sem 1

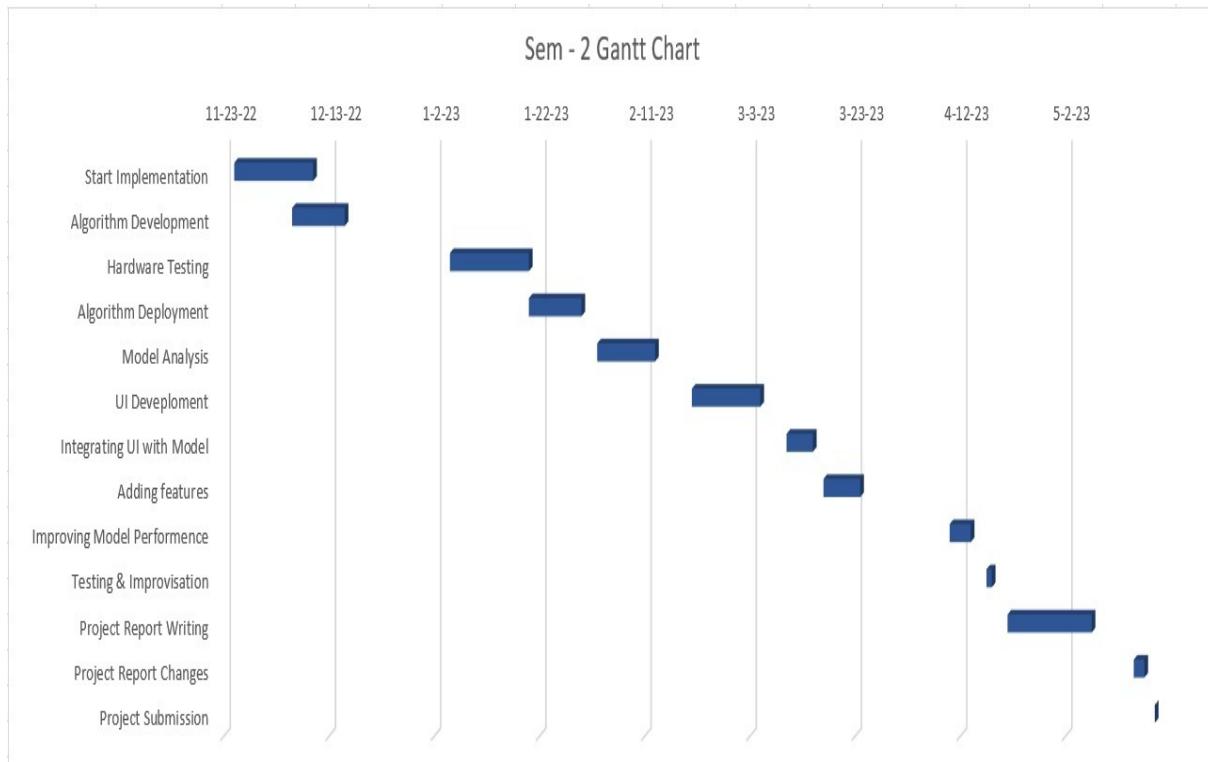


Figure 5.3: Gantt Chart for Sem 2

5.4 Team Organization

5.4.1 Team Structure

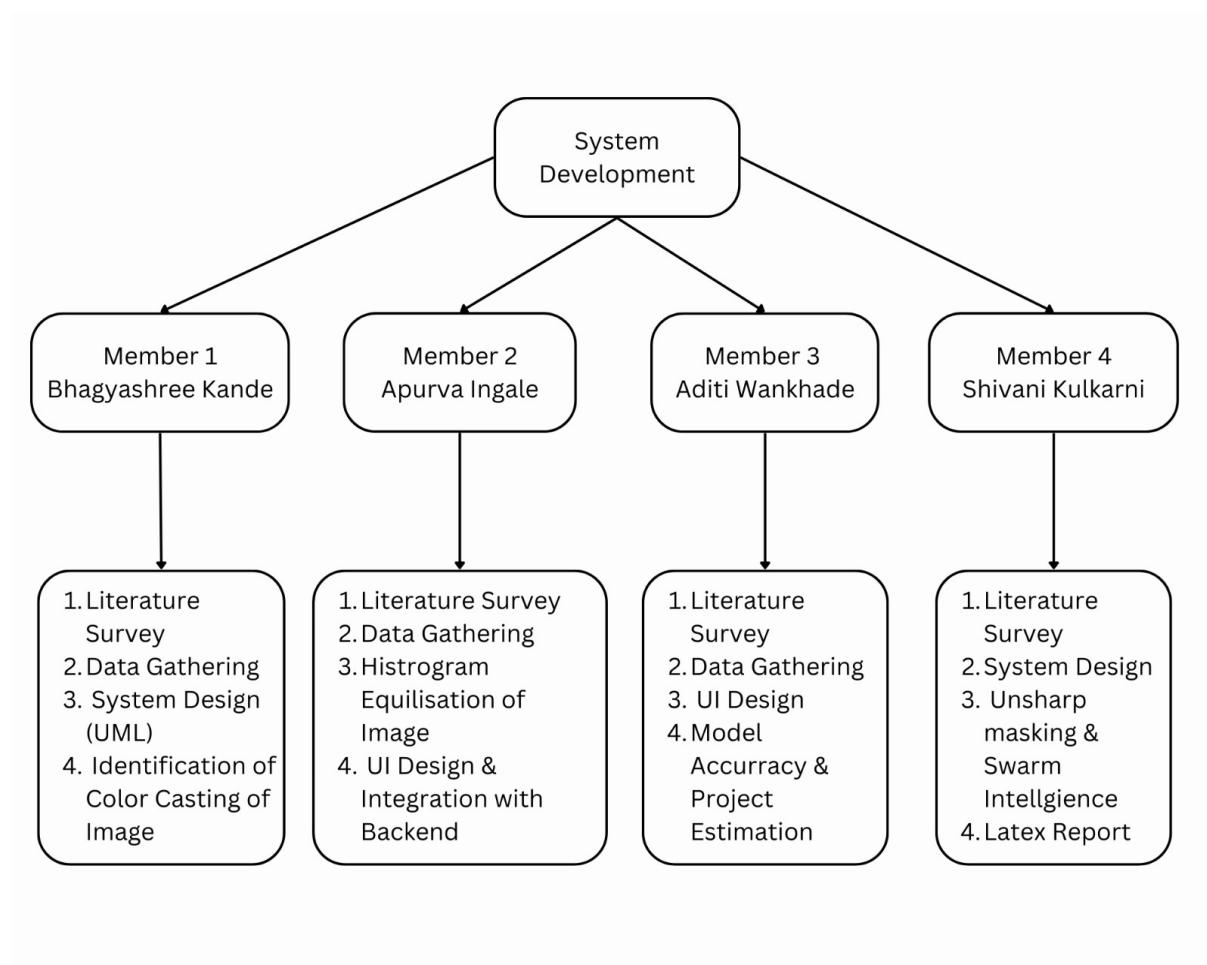


Figure 5.4: Team Structure

6 PROJECT IMPLEMENTATION

6.1 Overview of project module

We have developed a hardware based model using Raspberry Pi where the camera module is connected with raspberry pi and is used to capture underwater images . Python code is deployed in raspberry pi which is used to process and enhance the captured underwater images simultaneously while capturing . The input underwater image will undergo processing to enhance the image. There are four algorithms which will run sequentially to produce the output image. .The input captured images will be stored in one folder and the enhanced resultant images will be stored in the results folder . After processing the images a histogram of original input images and resultant images will be produced for comparing the difference.

6.2 Tools and Technologies used

6.2.1 Raspbian OS

The Raspberry Pi OS provides a desktop environment called PIXEL, which stands for Pi Improved Xwindows Environment, Lightweight, which is based on LXDE, which has a similar look and feel to several common operating systems, such as macOS and Microsoft Windows. There is a background image on the desktop of the computer. In the top portion of the screen is a menu bar that contains shortcuts to a web browser (Chromium), a file manager, and a terminal that can be accessed from a menu bar. It is also possible to access the Bluetooth menu, Wi-Fi menu, volume control, and clock on the other end of the menu bar. Besides that, the desktop appearance can also be changed from the default one, for example, by repositioning the menu bar on the screen.

6.2.2 VNC Viewer

In the context of Virtual Network Computing (VNC), Remote FrameBuffer Protocol (RFB) is a method that allows you to control another computer through a graphical desktop sharing system. VNC is an application which allows you to connect to your Raspberry Pi's graphical desktop remotely using your keyboard and mouse input. It allows you to pass on keyboard and mouse input

from one computer to another over the network, relaying graphical updates. Getting started with VNC is pretty simple, but you will usually only be able to connect to your Raspberry Pi from another computer that is on the same network as your Raspberry Pi when you set it up.

6.2.3 Python Thonny IDE

A free Python Integrated Development Environment (IDE) called Thonny was created specifically with the novice Pythonista in mind. It contains a built-in debugger that might be useful when you encounter ugly errors and, among other fantastic capabilities, it allows you to perform step through expression evaluation. Coding tools called Integrated Development Environments (IDEs) make authoring, debugging, and testing code simpler. Numerous offer beneficial features, like code completion, syntax highlighting, debugging tools, variable explorers, visualization tools, and many others.

6.2.4 Django

Django's main objective is to make it simpler to create intricate, database-driven websites. The ideas of don't repeat yourself, fewer code, less coupling, and component reusability and pluggability are strongly emphasized by the framework. Python is used everywhere, including in settings, files, and data models. Additionally, Django has a dynamically constructed administrative creation, read, update, and delete interface that is customisable.

Some key components of Django are: –

- 1) A small, independent web server for testing and development.
- 2) A framework for form serialization and validation that can convert between HTML forms and values appropriate for database storage.
- 3) A template system that makes use of the object-oriented programming idea of inheritance.
- 4) A framework for caching that supports several cache technologies.
- 5) Support for middleware classes that can perform custom functions and interfere at different points during the processing of requests.
- 6) An internal dispatcher system that enables events to be communicated between application components using pre-defined signals.

- 7) A method for internationalization that includes language translations of Django's own components.
- 8) A serialization system that can create and read representations of Django model objects in XML and/or JSON.
- 9) A system for increasing the template engine's functionality.
- 10) An interface for the unit testing framework that comes with Python.

6.3 Algorithm Details

6.3.1 NUCE

NUCE refers to natural-based underwater image color enhancement used to enhance underwater images.

- In order to reduce the underwater color cast, the proposed NUCE method enhances the inferior color channels by utilizing gain factors in order to enhance the inferior color channels. As a result of recognizing the differences between superior and inferior color channels, these gain factors can be calculated.
- Using dual-intensity images for fusion with each other, the proposed method improves the contrast of the image by averaging averages of mean and median values for the two intensity images. In order to produce lower-stretched and upper-stretched histograms, the average point is determined and selected as the point of separation for each of the two types of histograms.
- The third step of this method is to propose the use of swarm-intelligence based mean histogram equalization as an important step towards improving the naturalness of the output image. The mean values of inferior color channels are adjusted through the fusion of the swarm intelligence algorithm to be close to the mean values of superior color channels through the use of the swarm intelligence algorithm.

6.3.2 Histogram Equalization

We use histogram equalization to improve contrast. Contrast does not necessarily need to be raised in this. Histogram equalization may sometimes be detrimental in certain circumstances. The contrast is lessened in that instance. The majority of the pixels' intensities range from 0 to 50. Even the tiny portion of the image that the human eye can distinguish as white is black (albeit not entirely). In other words, the image's white portion is "less black" than its black portion, which is black. What if we could intensify the representation of the white portion without significantly altering the black portion? We can increase the contrast in the image by doing this. This technique is nothing but Histogram Equalization.

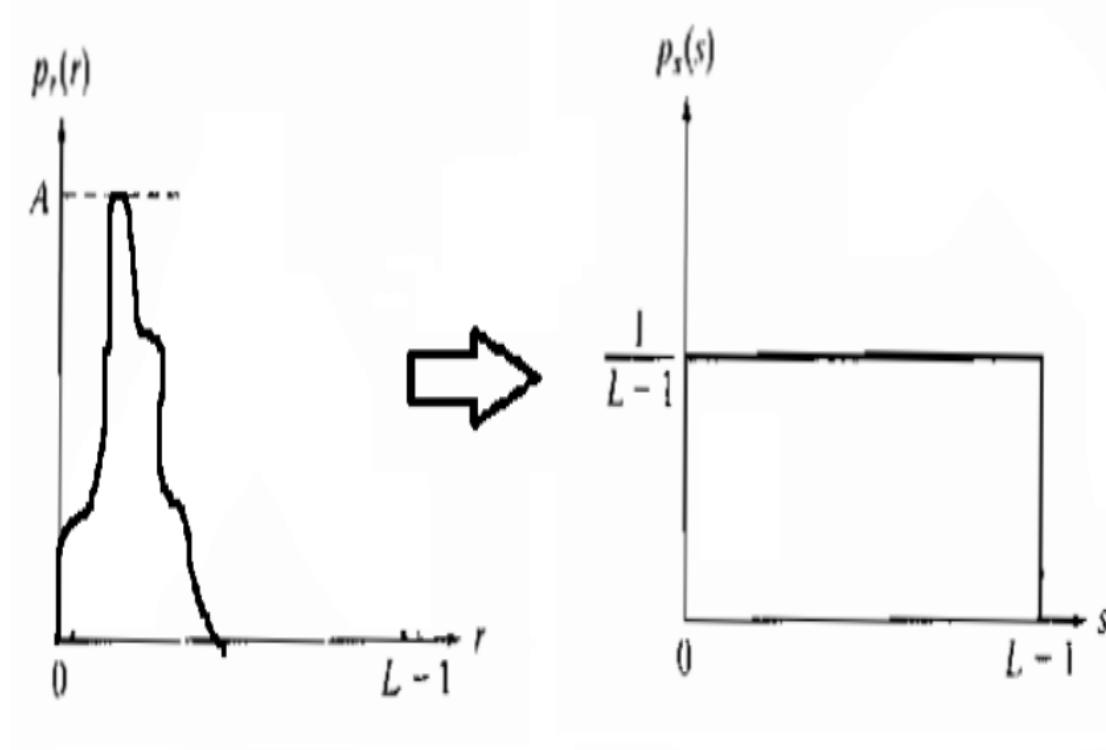


Figure 6.1: Histogram Equalization

7 Software Testing

7.1 Types of Testing

Following are the testing types Performed on IoT model:

7.1.1 Functional Testing

- 1) Unit Testing - It tests each module or component of an application. The IoT development team usually performs this task.
- 2) Integration Testing - when modules are integrated, it is essential to see how they work together.
- 3) End-to-End testing - this type involves running tests for the entire software product.
- 4) Smoke testing - This kind of testing enables the assessment of the software's stability.
- 5) Regression testing - Programme adjustments result from each new module that is added. If the IoT device's firmware needs to be updated, this could result in system changes as well. After each update, it is imperative to confirm that all components are still functioning properly.
- 6) Interface Testing - Testers verify the GUI meets the Specified Requirements and specifications.

7.1.2 Non Functional Testing

- 1) Compatibility testing

Compatibility tests should be done first since IoT systems are created using a variety of hardware and software configurations. This is because this is the most important stage of IoT testing. To ensure maximum compatibility, this process often entails testing multiple hardware, browsers, operating systems, and communication methods. It establishes compatibility between a target product that implements a standard specification and other products that implement the specification exactly. Each piece of software must be able to recognise input from other programmes, manage the workload necessary for it to play its part in the architecture, and produce outputs that are useful and accessible.

2) Performance testing

The second phase of IoT software testing begins with validating the performance-related implementation. Some of the key factors that the performance testing process typically deals with include:

- Performance under maximum load or data.
- Testing system for several devices at the same time.
- Communication tests between devices.
- System usability such as RAM load, battery usage, power consumption.
- Testing the device under various network conditions and environment factors.

3) Connection testing

The third testing stage ensures uninterrupted connectivity even when users cannot have a complete set of data. The stability of the IoT system depends on how well the devices and the hub are connected. After all, if it loses the connection for at least one second, it can cause data inaccuracy and system instability. Flawless connectivity, besides data recovery, is one of two critical features of connectivity testing.

Test Results: All the test cases mentioned above passed successfully and no defects/errors were encountered in the process.

7.2 Test Cases and Test Results

1. Module ID: 01

Module to be tested: hardware Testing

Test Case Id	Description	Result
101	Raspberry Pi compatibility testing	Verified and correct
102	Camera module connection with raspberry Pi	Verified and connected

Table 7.1: Test Cases for Hardware Testing

2. Module ID: 02

Module to be tested:Image Enhancement at different depth

Test case Id	Description	Input	Output	Result
201	Image captured at depth of 2.8-5 m			Pass
202	Image captured at depth of 3-8 m			Pass
203	Image captured at depth of 5-10 m			Pass

Table 7.2: Test Cases for image enhancement at different depth

3. Module ID: 03

Module to be tested: Images Storage

Test case Id	description	Input	Actual output	Expected Output	Result
301	Input images should be saved in input folder	1.jpg	Saved in input folder	Should be saved in input folder	Pass
302	Processed images should be saved in result folder	1.jpg	Saved in result folder	Should be saved in result folder	Pass

Table 7.3: Test Cases for Images Storage

8 RESULTS

8.1 Outcomes

In this project , we have developed a hardware based model using Raspberry Pi where the camera module is connected with raspberry pi and is used to capture underwater images . Python code is deployed in raspberry pi which is used to process and enhance the captured underwater images simultaneously while capturing .The input captured images will be stored in one folder and the enhanced resultant images will be stored in the results folder . After processing the images a histogram of original input images and resultant images will be produced for comparing the difference.

8.2 Screenshots

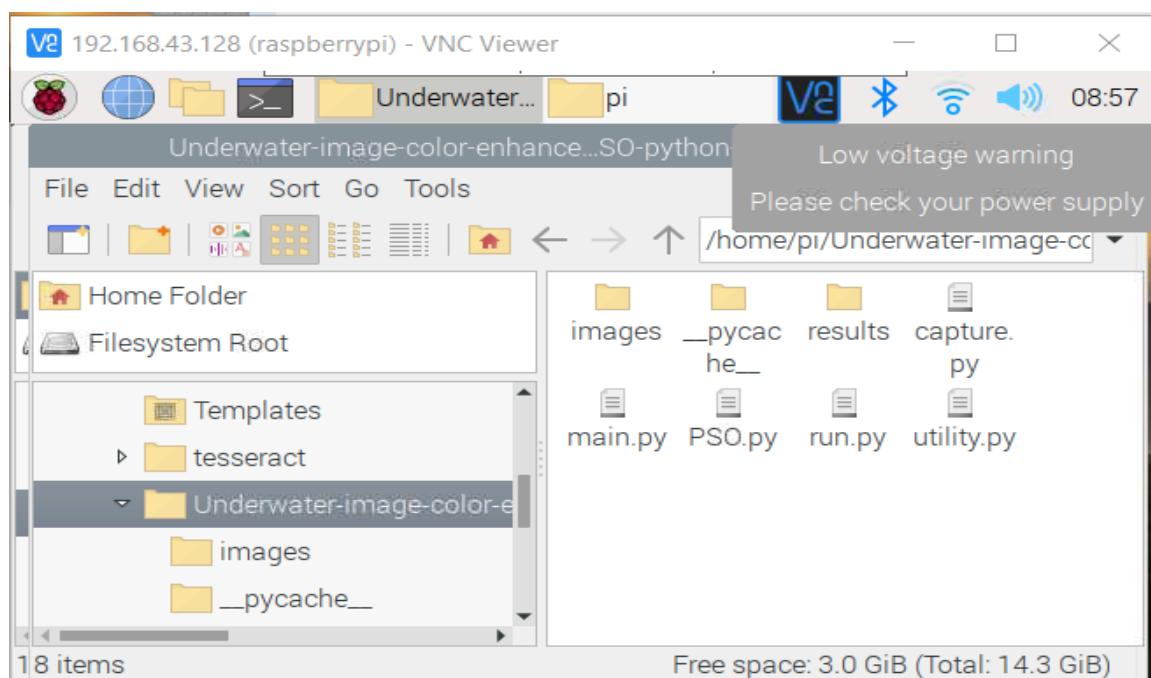


Figure 8.1: Project folder

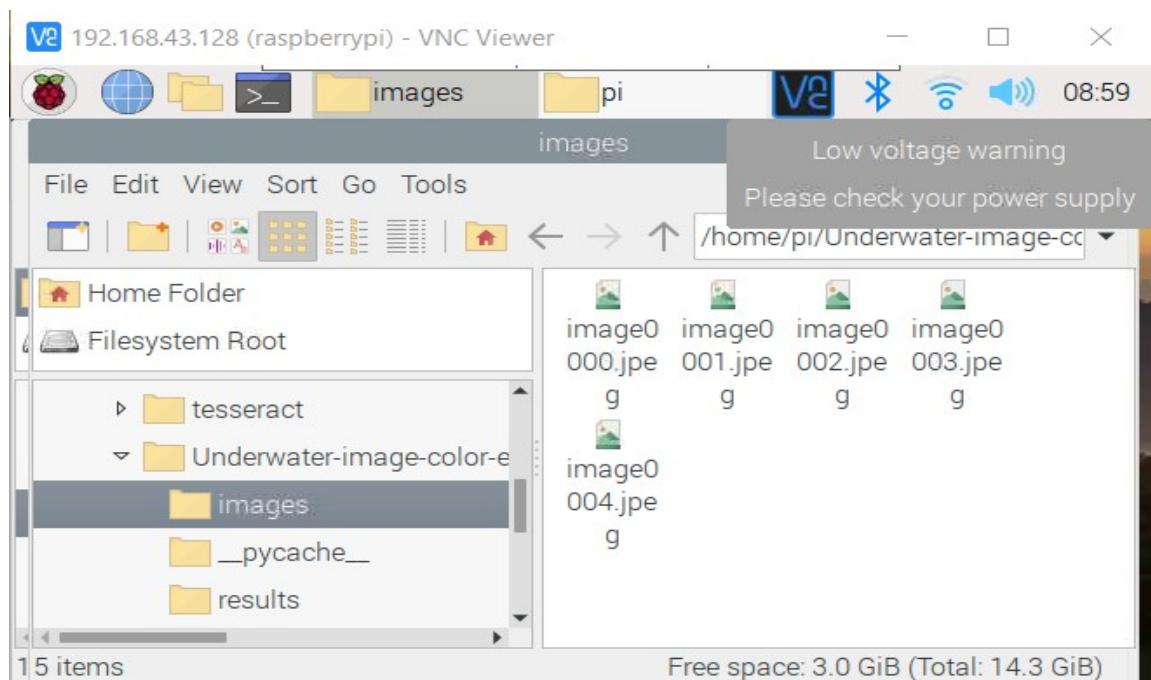


Figure 8.2: Input images folder

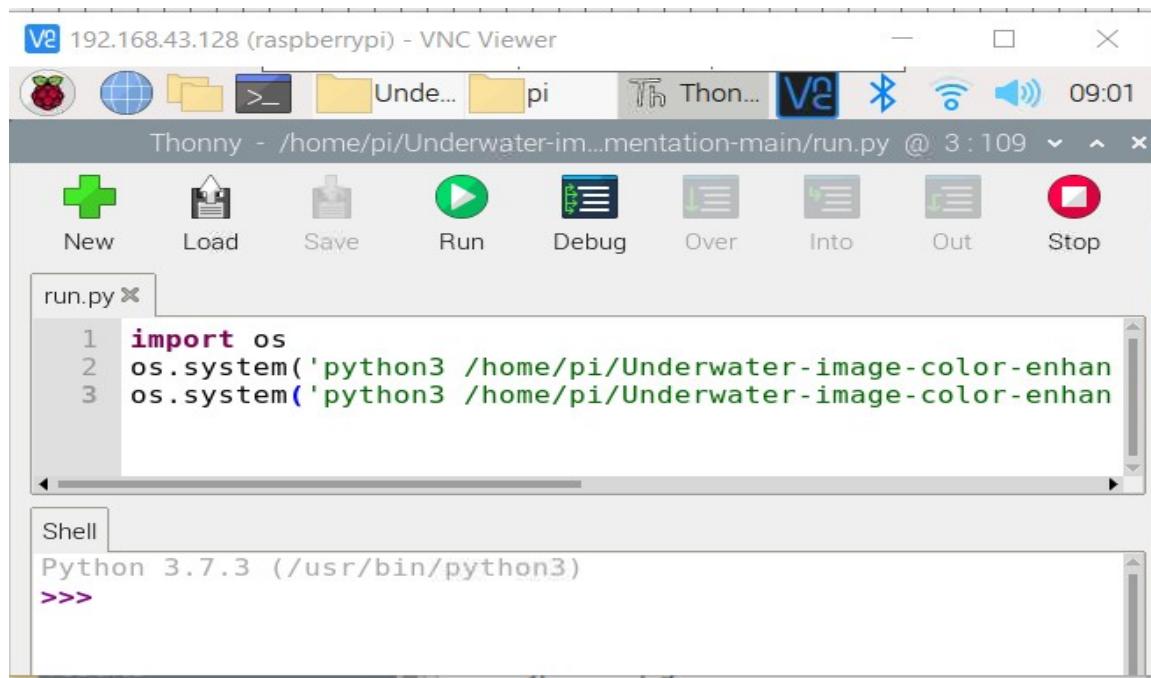


Figure 8.3: Run.py file

The screenshot shows the Thonny IDE interface on a Raspberry Pi via VNC. The top bar displays the connection information "192.168.43.128 (raspberrypi) - VNC Viewer" and the date/time "09:05". A "Low voltage warning" message is present. The toolbar includes icons for New, Load, Save, Run, Debug, and various serial port controls (Over, Into, Out, Stop). A status message "Please check your power supply" is shown. The code editor window contains a file named "run.py" with the following content:

```
1 import os
2 os.system('python3 /home/pi/Underwater-image-color-enhancement/color_enhancement.py')
3 os.system('python3 /home/pi/Underwater-image-color-enhancement/edge_enhancement.py')
```

The terminal window below shows the output of the script, indicating iterations and best fitness values:

```
Iteration = 60 best fitness = 0.002182
Iteration = 65 best fitness = 0.001281
Iteration = 70 best fitness = 0.000114
Iteration = 75 best fitness = 0.000003
Iteration = 80 best fitness = 0.000000
Iteration = 85 best fitness = 0.000000
Iteration = 90 best fitness = 0.000000
```

Figure 8.4: Iterations performed for processing

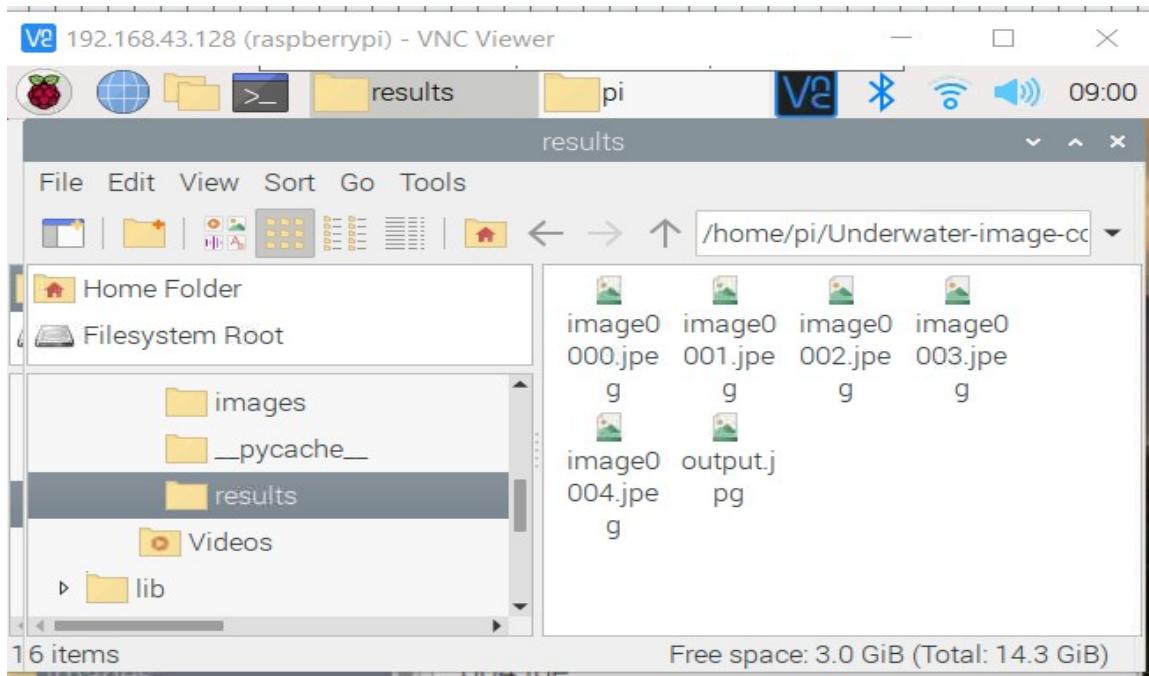


Figure 8.5: Result folder where output processed images are saved

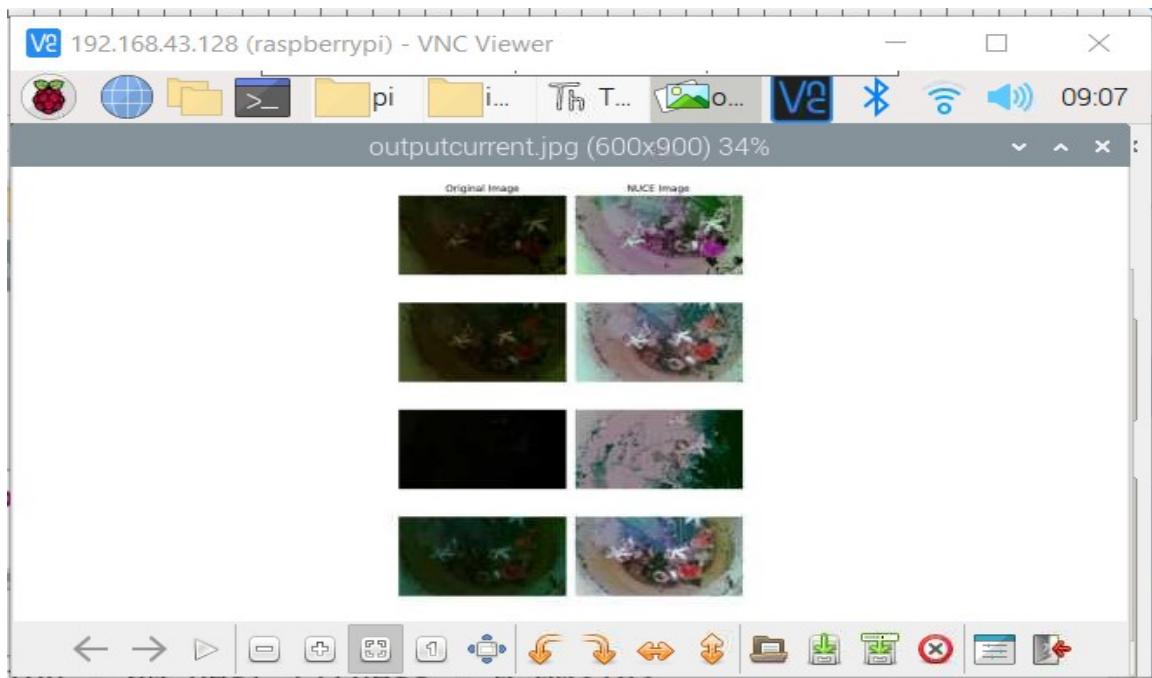


Figure 8.6: Histogram for comparison

8.3 Results by Comparing Pre existing models

For the purposes of testing the effectiveness of our method, a large number of raw underwater images were used, primarily from the Underwater Image Enhancement Benchmark (UIEB) and Enhancing Underwater Visual Perception (EUVP) dataset, to verify that it was effective.A comparison was made between the proposed method and several Pre-Existing methods for improving and enhancing underwater images that have recently been proposed, including Deep SESR-Master and Funie-Gan Method.

In figure —There are several raw underwater images in the dataset which were captured under different underwater scenes, which are compared with their processed counterparts which have been processed using different methods.Using Figure—(a), it can be seen that there is a lot of variation in the colors of the raw images and some are turbid, while others are showing plenty of fog on the surfaces of the objects. In Fig. -(b), the Deep SESR approach may effectively increase contrast and brightness, but the enhancement effect is ineffective and results in color distortion..

The method described in fig (c) that is Funie-GAN has outstanding effectiveness when dealing with different types of underwater distorted images, and it can successfully fix the distortions seen in Fig. -(b) but, considering the color difference, the photos after processing appear a touch dark. On the contrary,in fig(d) the method we use can not only enhance contrast and correct color deviation, but also standardize brightness distribution, which will ultimately end up in an improved visual effect. The Deep SESR model can be used to recover colors and textures from unseen natural images, as well as producing HR images and saliency maps based upon these images. As far as this model is concerned, although it claims to formulate a multi-modal objective function which evaluates the degree to which chrominance-specific color degradation, sharpness loss, and low-level feature representation can be achieved, it does not manage to overcome accurately the color distortion that occurs in underwater images. Funie-GAN has been observed to have some challenging cases, which have been illustrated by the example of an image . First of all, we observe that Funie-GAN is not very effective at enhancing severely degraded and texture-less images.In such cases, the generated images are often over saturated due to the noise amplification process that occurs.The color and texture recovery are generally poor even though the hue rectification is generally accurate.

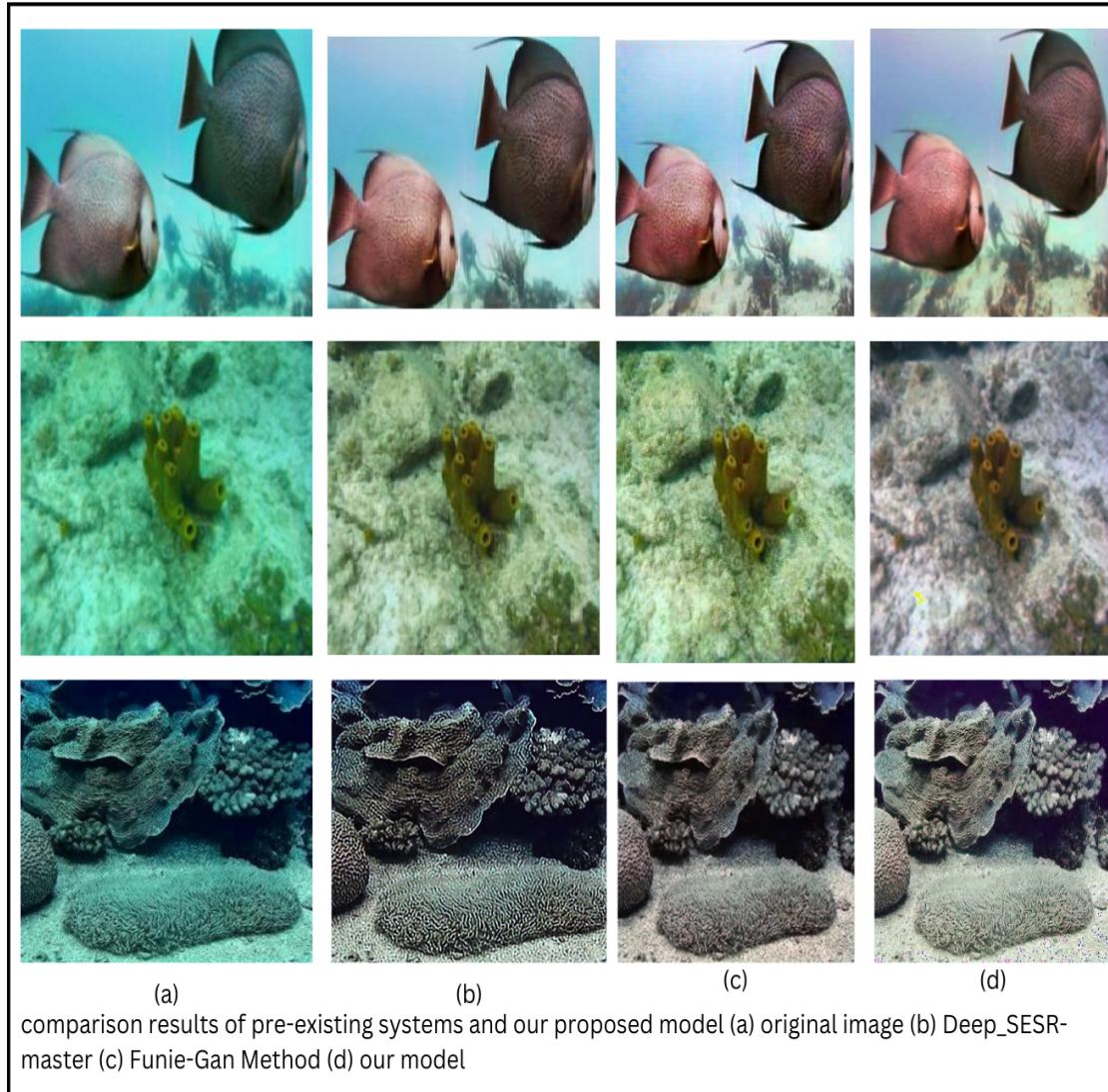


Figure 8.7: Results

9 CONCLUSIONS

9.1 Conclusion

Considering the characteristics of underwater imaging and the limitation of directly processing underwater images, In this paper, an improved and cost effective approach for underwater image enhancement is proposed . It is a hardware based model where the camera module is trained to improve quality of underwater image for better visibility using python code which is deployed in raspberry pi and enhanced image is stored in a folder. As evidenced by the results, the proposed method has the ability to enhance diverse underwater scenes with high clarity in addition to significantly minimizing the underwater color cast . Color Cast is reduced by regulating inferior and superior color channels . In order to improve the naturalness of the image in the output, a measure of average histogram equalization based on swarm-intelligence is put forward.

Existing methods have not been able to deal with the problems arising due to light scattering and light attenuation effectively, particularly when the images are captured deep beneath the surface of the ocean.By comparing various pre existing models we can obtain better and more precise results without reducing the originality of the real captured image.

9.2 Future Scope

As part of the future scope of our model, we intend to train it to work as a mobile application for easy accessibility in the future.It is intended that the application would be accessible and downloadable on public application stores such as the Google Play Store and/or the Apple App Store.By enhancing the testing methods more accuracy could be acquired. As opposed to visual methods, testing could be done by numerical methods, by doing research on the subject and finding a numerical figure that can be used as an accurate and representative indicator of the results, rather than visual methods.

Further models could be trained for object detection using machine learning or deep learning tech-

niques. This could be used for locating instances of underwater objects in images. Recognizing and identifying names of underwater bodies of interest and exploring the underwater world. An underwater clear video capturing feature should also be considered for future scope to make underwater film shooting a painless task.

9.3 Applications

- 1) Underwater scuba divers capture images of underwater bodies and explore underwater environments such as lakes , rivers, quarries , kelp forest and coral reefs.
- 2) Underwater researchers perform submarine experiments and scientists need to examine objects on the seafloor over time.
- 3) Underground mining images clearing used to identify and extract ore from below the surface.

APPENDIX A

Feasibility

1) Operational Feasibility

- Operational feasibility is a measure of how well a proposed system solves the identified problems, and takes advantage of the opportunities identified in the scope of research. The system should also satisfy the requirements identified in the requirement analysis phase of system development.
- The system provides liberty to the user for choosing the best fit algorithm to be implemented on captured image for correcting and processing the image.

2) Technical Feasibility

- The feasibility study is an influencing factor that contributes to the analysis of system implementation. The software components used are open source and freely available to reuse
- Technical feasibility study is carried out to determine whether the proposed system has the capability, in terms of software, hardware, personnel, and expertise to handle the completion of a task.
- The hardware components being used are of latest version and consists of all the upgraded technologies.

3) Economic Feasibility

- The financial resources to build this project are feasible as the product is going to be made with the help of open source software. Hardware components are expensive. The project only consists production cost and implementation cost are almost negligible.
- There is a need in the market for such a product as no such product exists and therefore there will be a demand for this product.

Problem Type

According to our research and analysis, the system will give an output in polynomial time Therefore the problem statement is of P Type.

APPENDIX B

- [1] Li, Changli Tang, Shiqiang Kwan, Hon Yan, Jingwen Zhou, Teng. (2020). Color Correction Based on CFA and Enhancement Based on Retinex With Dense Pixels for Underwater Images. IEEE Access. 8. 10.1109/ACCESS.2020.3019354.
- [2] M. J. Islam, Y. Xia and J. Sattar, "Fast Underwater Image Enhancement for Improved Visual Perception," in IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 3227-3234, April 2020, doi: 10.1109/LRA.2020.2974710.
- [3] Yunfeng Zhang, Qun Jiang, Peide Liu, Shanshan Gao, Xiao Pan, Caiming Zhang, "Underwater Image Enhancement Using Deep Transfer Learning Based on a Color Restoration Model", IEEE Journal of Oceanic Engineering, vol.48, no.2, pp.489-514, 2023.
- [4] Dadios, Elmer Almero, Vincent Jan Concepcion II, Ronnie Vicerra, Ryan Bandala, Argel Sybingco, Edwin. (2022). Low-Cost Underwater Camera: Design and Development. Journal of Advanced Computational Intelligence and Intelligent Informatics. 26. 851-858. 10.20965/jaciii.2022.p0851.
- [5] H. Li, P. Zhuang, W. Wei and J. Li, "Underwater Image Enhancement Based on Dehazing and Color Correction," 2019 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom), Xiamen, China, 2019, pp. 1365-1370, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00196.
- [6] Islam, Md Jahidul Luo, Peigen Sattar, Junaed. (2020). Simultaneous Enhancement and Super-Resolution of Underwater Imagery for Improved Visual Perception. 10.15607/RSS.2020.XVI.018.
- [7] C. Li et al., "An Underwater Image Enhancement Benchmark Dataset and Beyond," in IEEE Transactions on Image Processing, vol. 29, pp. 4376-4389, 2020, doi: 10.1109/TIP.2019.2955241.

- [8] Sharma, Prasen Bisht, Ira Sur, Arijit. (2021). Wavelength-based Attributed Deep Neural Network for Underwater Image Restoration.
- [9] C. Ancuti, C. O. Ancuti, T. Haber and P. Bekaert, "Enhancing underwater images and videos by fusion," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 81-88, doi: 10.1109/CVPR.2012.6247661.
- [10] C. Li, C. Guo, W. Ren, R. Cong, J. Hou, S. Kwong, D. Tao, "An Underwater Image Enhancement Benchmark Dataset and Beyond," IEEE Trans. Image Process., vol. 29, pp.4376-4389, 2019.

APPENDIX C: PLAGIARISM REPORT

- Page Count: 34
- Word Count: 5887
- Character Count: 43056
- Unique: 92.7 %
- Plagiarism: 7.3 %

REFERENCES

1. C. Li, S. Tang, H. K. Kwan, J. Yan and T. Zhou, "Color Correction Based on CFA and Enhancement Based on Retinex With Dense Pixels for Underwater Images," in IEEE Access, vol. 8, pp. 155732-155741, 2020, doi: 10.1109/ACCESS.2020.3019354.
2. H. Li, P. Zhuang, W. Wei and J. Li, "Underwater Image Enhancement Based on Dehazing and Color Correction," 2019 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom), 2019, pp. 1365-1370, doi10.1109/ISPABCDCloud-SustainCom-SocialCom48970.2019.00196.
3. Yang, H., Tian, F., Qi, Q., Wu, Q.M.J., Li, K.: Underwater image enhancement with latent consistency learning-based color transfer. IET Image Process. 16, 1594–1612 (2022).
<https://doi.org/10.1049/ipr2.12433>.
4. Sarma, Kaushik Pandi, Vigneshwaran. (2022). Underwater Image Enhancement Using Deep Learning. 10.1007/978-3-030-84760-938.
5. Kaur, Ramandeep. "Image Enhancement of Underwater Digital Images by Utilizing L*A*B* Color Space on Gradient and CLAHE Based Smoothing." Communications on Applied Electronics (CAE), Foundation of Computer Science FCS, New York, USA, – ISSN : 2394-4714 4.9 (2016): 22–30. Web.