**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**

Our web portal - ’GalacTech’ was developed in the Django framework. It is a Python-based free and open-source web framework that follows the model–template–views (MTV) architectural pattern. Django’s primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and pluggability of components, less code, low coupling, rapid development, and the principle of don’t repeat yourself. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Some key components of Django are: –

* A lightweight and standalone web server for development and testing .
* a form serialization and validation system that can translate between HTML forms and values suitable for storage in the database .
* A template system that utilizes the concept of inheritance borrowed from object-oriented programming .
* A caching framework that can use any of several cache methods.
* Support for middleware classes that can intervene at various stages of request processing and carry out custom functions .
* An internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signals .
* An internationalization system, including translations of Django’s own components into a variety of languages .
* A serialization system that can produce and read XML and/or JSON representations of Django model instances .
* A system for extending the capabilities of the template engine.
* An interface to Python’s built-in unit test framework.

**6.3 Algorithm Details**

**6.3.1 Convolutional Neural Networks**

The CNN algorithm is a powerful image processing algorithm. As far as automated image processing algorithms are concerned, these algorithms are currently the most effective tools that are available to us. The algorithms are used by many companies to figure out what objects are present in a picture, and to identify the objects.

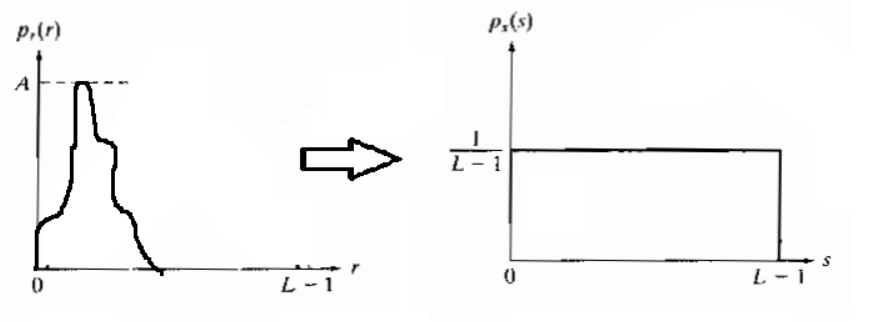
Data contained in images is a combination of RGB colors. A file can be imported into memory from a file using Matplotlib and an image can be plotted from it. The computer doesn’t see an image, all it sees is an array of numbers. It's important to note that the computer doesn't see an image at all, it only sees an array of numbers. A color image is stored in a 3-dimensional array that can be manipulated in three dimensions. The first two measurements are the image's height and width (measured in pixels). The red, green, and blue hues found in each pixel are represented by the final dimension..

Convolutional Neural Networks with a focus on image and video recognition applications. CNN is mostly utilized for image analysis applications like segmentation, object detection, and picture recognition. Convolutional Neural Networks have three different kinds of layers:

* Convolutional Layer: In a typical neural network each input neuron is connected to the next hidden layer. In CNN, only a small region of the input layer neurons connect to the neuron hidden layer.
* Pooling Layer: The pooling layer is used to reduce the dimensionality of the feature map. There will be multiple activation & pooling layers inside the hidden layer of the CNN.
* Fully-Connected layer: Fully Connected Layers form the last few layers in the network. The input to the fully connected layer is the output from the final Pooling or Convolutional Layer, which is flattened and then fed into the fully connected layer.

**6.3.2 Histogram Equalization**

Histogram equalization is used to enhance contrast. It is not necessary that contrast will always be increased in this. There may be some cases where histogram equalization can be worse. In that case the contrast is decreased. Most of the pixels have intensity values between 0 to 50. The little part of the image that the human eye can recognize as white is also black (Not completely). In other words, the black part of the image is black and white part of the image is not white but it is “less black”. What if we can represent the white part with higher intensity while not affecting the black part (much). By doing so we can achieve more contrast in the image. This technique is nothing but Histogram Equalization.



Histogram Equalization

**Software Testing**

**7.1 Types of Testing**

Following are the testing types Performed on IoT model

**7.1.1 Functional Testing**

* **Unit Testing -** It tests each module or component of an application . The IoT development team usually performs this task.
* **Integration Testing -** when modules are integrated , it is essential to see how they work together.
* **End-to-End testing -** this type involves running tests for the entire software product.
* **Smoke testing -** This kind of testing enables the assessment of the software's stability. .
* **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.
* **InterfaceTesting -** Testers verify the GUI meets the Specified Requirements and specifications.

**7.1.2 Non-functional Requirements**

#### **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.

#### **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.

#### **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 |

**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 |

**3. Module ID: 03**

**Module to be tested:images Storage**

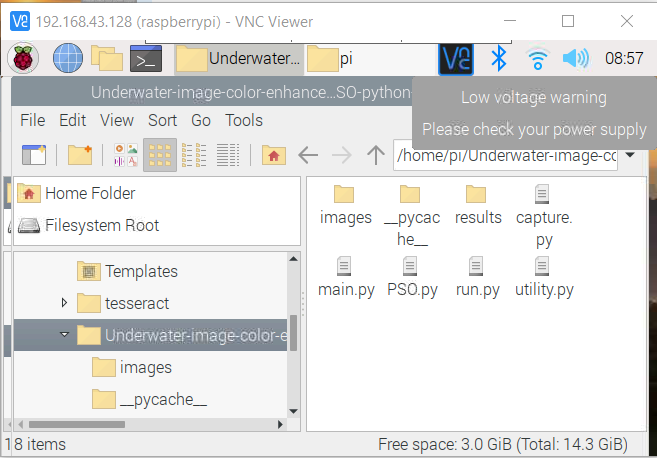
| **Test case Id** | **description** | **input** | **Actual output** | **Expected Output** | **Result** |
| --- | --- | --- | --- | --- | --- |
| 301 | Input images should be saved in input folder | 1.jpg | Saved in input folder | Should be saved in input folder | Pass |
| 302 | Processed images should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

**Results**

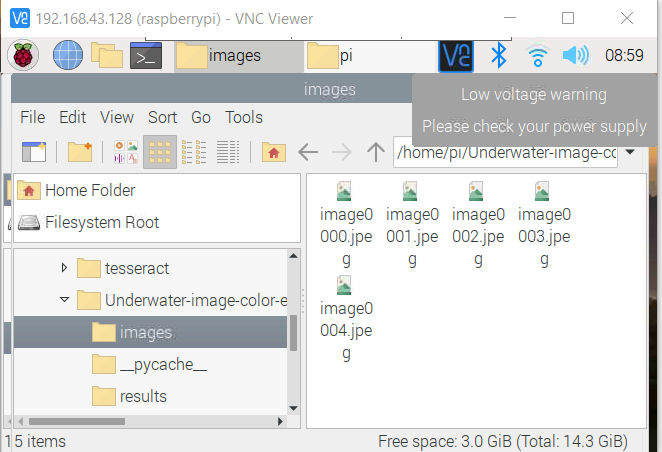
**8.1 Outcomes**

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 .

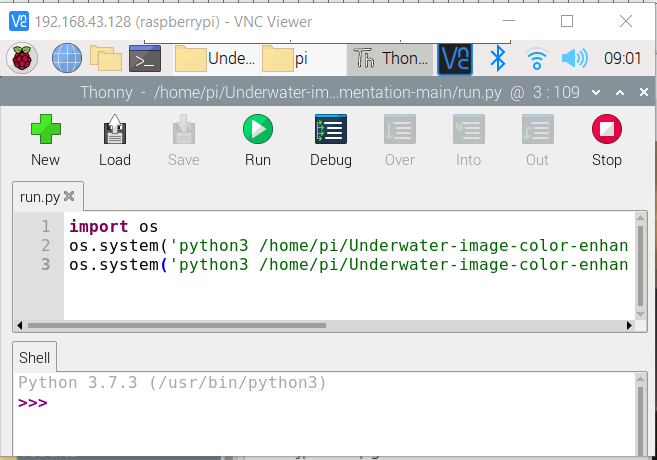
**8.2 Screenshots**

****

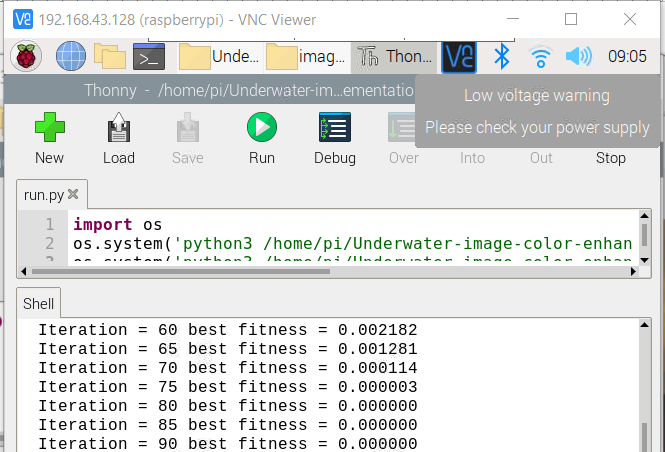
Project folder

****

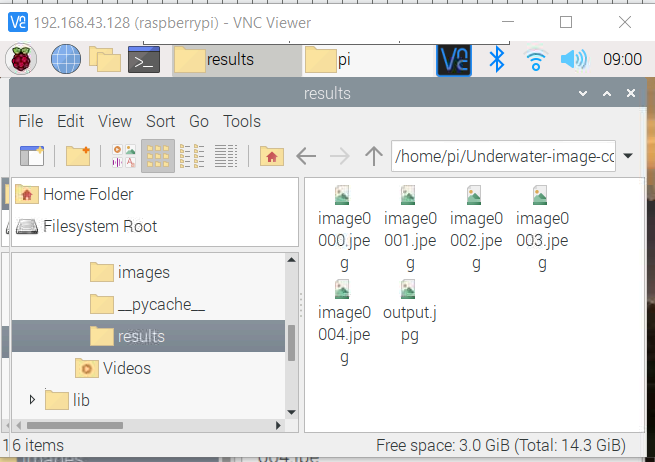
Input images folder

****

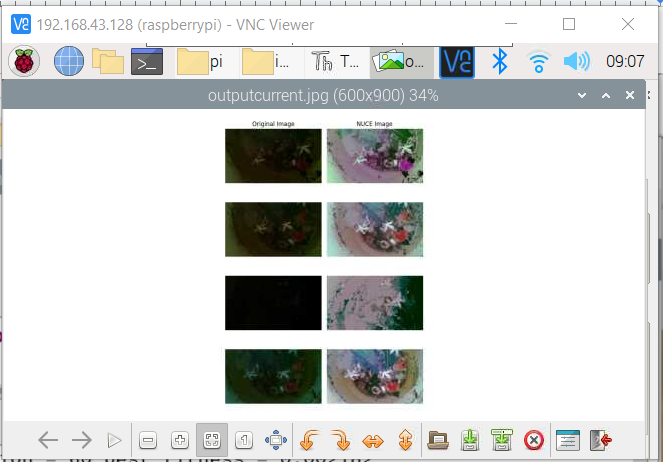
Run.py file

****

Iterations performed for processing

****

Result folder where output processed images are saved

****

Histogram for comparison

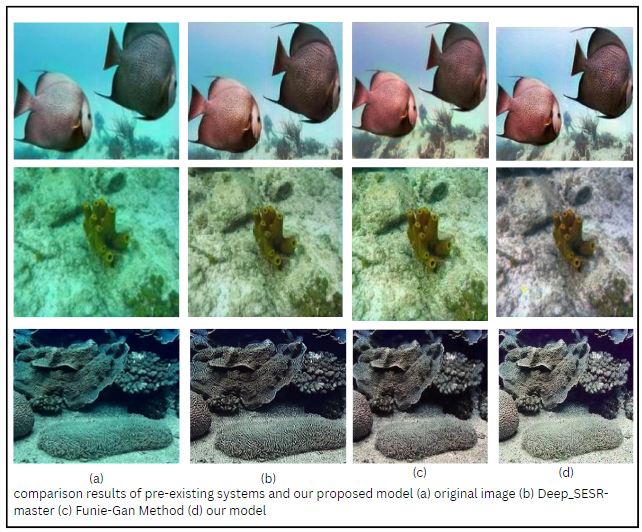
**8.3 Results by Comparing Pre existing models**

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 oversaturated due to the noise amplification process that occurs.The color and texture recovery are generally poor even though the hue rectification is generally accurate.



**Conclusion**

**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 Advantages**

1) Our Proposed model will be a hardware-based model that captures underwater images and improves the quality of degraded images under the water itself.

2) Resultant images after processing are produced as natural images using Swarm Intelligence

3) Model is focusing on reducing human efforts and time.

**9.3 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 techniques. 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.

**9.4 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. 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 of 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.