Underwater Image Quality Enhancement by using L2UWE Framework

Yunfei(Alex) Tian
Electrical Engineering Department
University of Victoria
Victoria, British Columbia, Canada
yunfeitian@uvic

Abstract— Underwater image enhancement poses a challenge due to non-uniform lighting, water absorption, and scattering, which can obscure important features such as edges and textures. This report uses a framework, L2UWE which proposed by Tunai Porto Marques and Alexandra Branzan Albu, for enhancing low-light underwater image quality using the dark channel prior, contrast code image, and multi-scale fusion techniques. The proposed method combines a finely-detailed model and a brighter model of the original image after pre-processing tasks, and the merged result is used to generate the final enhanced image. The report provides a detailed discussion of the pre-processing and multi-scale fusion procedures, with the final enhanced result presented at the end.

I. INTRODUCTION

Underwater image generally presents a poor quality due to several reasons. Underwater non-uniform lighting reduce the image quality. It is hard to detect features such as edges, and textures. In additional, water absorption reduces the light propagation. And water scattering is also another reason that reduce the underwater image quality, which lead to the captured images have dark region, low contrast, color distortion, reduced sharpness, and lazy appearance. An example of such an image is shown in Figure 1, which is a random sample from the USR-248 database[1].

Figure 1. The input testing image from USR-248 database

We could observe that the image exhibits low-light effects, with a dark and unclear background, deteriorated colors, and unclear edges of objects. To address this, the L2UWE framework developed by Tunai Porto Marques and Alexandra Branzan Albu, can enhance the image quality. This framework is developed for three areas: underwater image enhancement, aerial image dehazing and low-light image enhancement. However, we would only apply this framework to enhance the underwater image quality. This framework combines few advanced image processing techniques including dark channel prior-based dehazing of single images, contrast-guided approach for the enhancement of low-light images, and multi-scale fusion for image enhancement. These algorithms were proposed by different authors. However, Framework combines all these techniques and perform image decomposition, local contrast enhancement, multi-scale fusion ,and image reconstruction. The result outperforms the compared methods, as evaluated by various standards such as UIQM, PCQI, GCF, e- and r-scores, FADE, and SURE. This report presents the implementation details of the L2UWE framework. The final result will be demonstrated efficiently to enhance the low-light underwater images at the end of the report.

II. METHODOLOGY AND TECHNICAL REVIEW

Before delving into the main algorithm, it's important to understand how hazy images are formed. According to Tunai *et al*. [2], a hazy image $I(x)$ is the sum of direct attenuation and airlight. In other words:

$$I(x) = J(x)t(x) + A\infty(1 - t(x)) \qquad (1)$$

Here, J is the haze-less version of image that we are trying to obtain in this project. The parameter t represents the transmission map, which is a grayscale image that encodes the amount of light that passes through different parts of an image. The direct attenuation, $D(x) = J(x)t(x)$, is the result of multiplying the haze-less image and the transmission map. It indicates the the attenuation suffered by the scene radiance due to properties of the medium. $A\infty$ is the estimation of global atmospheric lighting. Airlight $V(x)$ can be calculated using the relation of $A\infty(1 - t(x))$.

By understanding how hazy images are formed, we can see that we can obtain the haze-free image by simply converting Equation1, as long as we calculate the estimated atmospheric light and transmission map. The question is, how can we find all the unknown values using only the captured hazy image $I(x)$? Fortunately, He *et al*. [3] discovered the *Dark Channel* and *Dark Channel Prior* technique to estimate the atmospheric lighting $A\infty$. Moreover, the Dark Channel Prior technique can also be used to derive the transmission map t.

A. *Dark Channel Prior-based dehazing of single images*

The dark channel prior is a simple yet powerful method for removing haze from images. By estimating the atmospheric light and transmission map, the method is able to remove the haze from the image and reveal the haze-free version. While the method is generally effective in many scenarios, there may be some limitations in extremely low-light or high-contrast images. Nevertheless, the charm of simplicity in achieving remarkable results makes the dark channel prior approach an enduring and effective technique for haze removal.

The dark channel is a channel in a RGB color image where some pixel values are very low, close to 0, especially for some of the most common colors in RGB format. For exmaple, red (255, 0, 0), orange (255, 165, 0), yellow (255, 255, 0), and green (0, 255, 0). He *et al*. [3] analyzed over 5000 haze-free images and found that the grayscale values of the dark channels of haze-free outdoor images are concentrated in the first bin, which ranges from 0 to 15. This natural law can be expressed as a histogram in Figure 2:

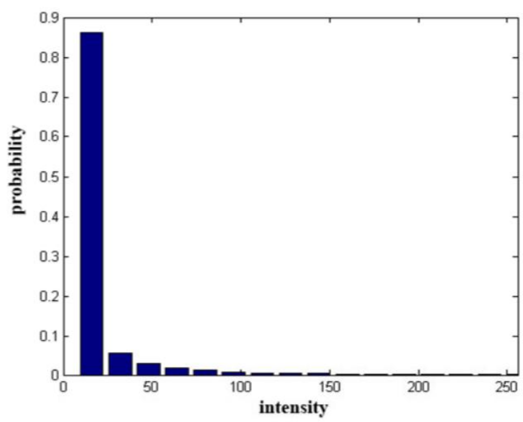


Figure 2. histogram of the intensity of the pixels in all of the 5000 dark channels

To demonstrate this principle, "Single Image Haze Removal Using Dark Channel Prior"[3] provides a haze-free image database with dark channels. Figure 3 shows haze-free example images and their corresponding dark channels. It can be observed that in haze-free images, even in non-sky regions with good exposure, there are usually some pixels in one of the three RGB channels that have very low values, close to 0. In contrast, in hazy images, Figure 4 shows the dark channel values are much larger than 0, appearing as gray-white (gray value 128) even in red, blue, and green regions, although not very obvious.

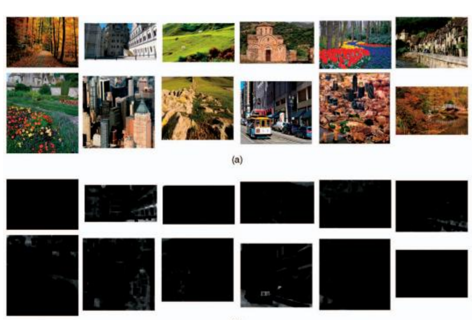


Figure 3. (a)Haze-free example images. (b) the corresponding dark channel

Figure 4. (a) Input hazy image. (b) The corresponding dark channel

To compute the dark channel of a hazy image, we can use Equation 2, which involves finding the minimum value among the three color channels at pixel y in the image *I(x)*, and then finding the minimum value within the patch Ω for the resulting single-channel image, which is the minimum intensity of each channel in a local window.

$$J^{dark}(x) = \min_{y\in\Omega(x)} \left(\min_{c\in\{r,g,b\}} I^c(y) \right) \qquad (2)$$

The dark channel, which is a channel in a RGB color image where some pixel values are very low, plays a crucial role in this method, and can be computed by finding the minimum value among the three color channels at pixel y in the image *I(x)*, and then finding the minimum value within the patch Ω for the resulting single-channel image. It is the minimum intensity of each channel in a local window. The dark channel prior is a simple and effective method for removing haze from images. Most importantly, obtaining the Dark Channel Prior simplifies the process of calculating the atmospheric light and transmission map.

The estimation of global atmospheric light vector A for a single three-dimensional image can be calculated by analyzing the 0.1% or 0.2% brightest pixels in the dark channel. Tunai *et al*. [2] suggest that a single global value may not accurately represent the illumination in low-light scenes, so they propose using equation 3 to estimate the local atmospheric light intensity within patches.

$$A^c_{L\infty}(x) = \max_{y\in\Psi(x)} \left(\min_{z\in\Omega(y)} (I^c(z)) \right) \qquad (3)$$

Furthermore, the Dark Channel Prior can be directly used to derive the transmission map *t*. He *et al.* [3] suggest to compute the dark channel of the input image by taking the minimum intensity value over a window of a certain size, normalizing the dark channel by dividing it by the maximum value over the entire image, and then compute the transmission map by taking the complement of the normalized dark channel, as shown in Equation 4.

$$t(x) = 1 - \omega \min_{y\in\Omega(x)} \left(\min_{c\in\{r,g,b\}} \frac{I^c(y)}{A^c_\infty} \right) \qquad (4)$$

After obtaining the atmospheric light intensity and transmission map, it is straightforward to calculate the haze-free image by a simple conversion, as shown in Equation 5.

$$J^c(x) = \frac{I^c(x) - A^c_\infty}{\max(t(x), t_0)} + A^c_\infty \qquad (5)$$

However, this output may not be satisfactory due to the use of a single-sized patch. Using a single-sized patch can result in oversaturation for small patch sizes and undesired halos for large patch sizes. A single patch size is typically not optimal for images of varying scales. Fortunately, Marques *et al*. [4] introduced the contrast-coded image algorithm, which provides a dynamic patch size and solves the issue of using a single-sized patch.

B. *Contrast-guided approach for the enhancement of low-light images*

In simple terms, the CCI algorithm is based on the prior knowledge of dark channel and local color variation properties, and is used to remove haze from images. For each pixel, the CCI algorithm calculates the standard deviation of the surrounding patch for each pixel and selects the patch with the smallest standard deviation as its corresponding patch. Note the value will be place in the center. The algorithm introduces a variable i to control the size of the patch. Typically, i is selected from {1,2,3,...,7}. By adjusting the value of i, the balance between haze removal effect and preservation of image details can be achieved. Equation 6 presents how to obtain the contrast code image.

$$CCI(x) = \arg\min_i [\sigma(\Omega_i(x))] \qquad (6)$$

Marques *et al*. [4] explain the algorithm in detail in their paper, and the final CCI result during the local window is represented by the *c* value as Figure 5 shows.

The dynamic *c* value will be used to calculate a dynamic patch size. According to Tunai *et al*. [2], they proposed a model to calculate the dynamic patch size. Equation 7 shows the detail.

$$S_\Upsilon(m,c) = 3m - [\frac{m}{3}(c-1)] \qquad (7)$$

Formula 7 provides a formal expression of this reasoning, reflecting the relationship between the CCI code c and the size of the lighting square patch used in the local atmospheric lighting model calculation.

```
Data: Hazy image
Result: Contrast code image
CCI ← zeros
Sigmas ← zeros
while not all pixels (x,y) in hazy image are visited do
    c ← 7
    while c >= 1 do
        s ← (c * 2) + 1
        std ← standard deviation inside s×s square window centered at pixel (x,y)
        if c = 7 then
            CCI(x,y) ← 7
            Sigmas(x,y) ← std
        else
            decay ← 1 − (t/100)
            if std < (Sigmas(x,y) * decay^(c−1)) then
                CCI(x,y) ← c
            end
        end
        c ← c − 1
    end
end
```

Figure 5. Calculation of the contrast code image (CCI).

Now, instead of using Equation 3, we can use a better and more accurate model, Equation 8, to estimate the atmospheric light intensity, and the adjusted equation is shown below:

$$A^c_{LCG\infty}(x,m) = \max_{y\in\Upsilon(x,m)} \left(\min_{z\in\Omega(y)} (I^c(z)) \right) \qquad (8)$$

However, determining the optimal value of *m* for calculating the patch size depends on the multi-scale fusion process and requires further discussion, which will be addressed in the next section.

C. *Multi-scale fusion for image enhancement*

At this point, we have been able to develop a new atmospheric light intensity model that can be applied to obtain a haze-free image. However, we still need to determine the patch size by selecting the appropriate value of m from Equation 7. To address this, Tunai *et al*. [2] suggest deriving two patches with m = 5 and m = 30. The smaller patch captures finer details of the original image, while the larger patch creates a brighter model. Once the two different models are generated, multi-scale fusion can be used to combine them and produce a final result, which is a novel technique that enhances the clarity of hazy images.

Multi-scale fusion works by utilizing two inputs derived from the original hazy image and then using a multi-scale fusion process to blend them. First, three weight maps are calculated for each input image based on luminance, chromaticity, and saliency, which are used to integrate salient features, contrast, and exposedness, resulting in accurate images. According to the article "Under Water Image Enhancement by Fusion" [5], the chromatic weight map controls saturation gain, the luminance weight map balances brightness by controlling luminance gain, and the saliency weight map identifies the property with respect to neighborhood regions and reflects the distinction between a particular region and its neighboring areas. The saliency map generation method used in this approach yields higher precision and better recall rates than traditional methods.

Next, a Gaussian pyramid is generated for the normalized weight map of each input, and both input images are decomposed into a Laplacian pyramid, which is a sequence of images representing the difference between an image and its smoothed version. The two Laplacian pyramids are independently fused using an appropriate equation, producing a final image that is free of haze. It is worth noting that the multi-scale fusion approach offers flexibility and versatility in its strategies for the multi-scale fusion process, as highlighted in other works. Overall, the multi-scale fusion approach represents a promising technique for dehazing images and can lead to improvements in several computer vision applications.

Ancuti et al. [6] provided Equations 9, 10, and 11 to calculate the three weight maps for luminance, chromaticity, and saliency, respectively.

$$\mathcal{W}^k_L = \sqrt{1/3\left[(R^k - L^k)^2 + (G^k - L^k)^2 + (B^k - L^k)^2\right]}. \qquad (9)$$

$$\mathcal{W}^k_C(x) = exp\left(-\frac{(S^k(x) - S^k_{max})^2}{2\sigma^2}\right) \qquad (10)$$

Identify applicable funding agency here. If none, delete this text box.

$$\mathcal{W}_S^k(x) = \left\| I_k^{\omega_{hc}}(x) - I_k^{\mu} \right\| \tag{11}$$

Note that the saliency weight map is computed by subtracting a Gaussian-smoothed version of the input from the mean intensity value of the input, using ([1, 4, 6, 4, 1])/16 as Gaussian Kernel. The luminance weight map is calculated based on the saturation of the colors present in the image. The local contrast weight map is computed by applying a Laplacian kernel ([[-1,-1,-1],[-1,8,-1],[-1,-1,-1]])/8, provided by Tunai *et al*. [2], on the mean intensity of the image. The resulting weight maps are used to highlight regions of the image that are more salient or have more local intensity variation. Finally, we can combine the weight maps and normalize the result from which a 5-level Gaussian pyramid is derived as shown in Equation 12 below.

$$\mathcal{F}_l(x) = \sum_k G_l\{\bar{\mathcal{W}}^k(x)\} L_l\{\mathcal{I}_k(x)\} \tag{12}$$

The final haze-less image is obtained by summing the contribution of the resulting inputs, which are the levels of the pyramid, as shown in Equation 13.

$$\mathcal{J}(x) = \sum_l \mathcal{F}_l(x) \uparrow^d \tag{13}$$

III. IMPLEMENTATION

The implementation of the L2UWE framework is based on the pipeline provided by Tunai *et al*. [2], as shown in detail in Figure 6.

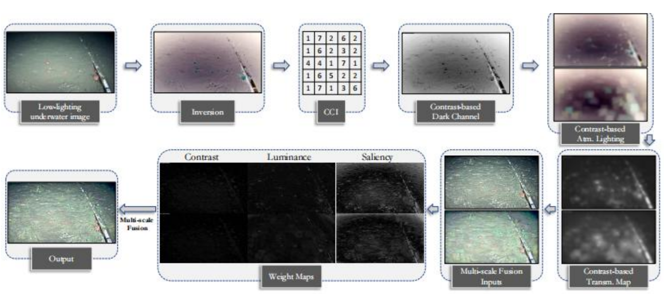


Figure 6. The pipeline of L2UWE framework.

All technical aspects have been reviewed in the previous section, and the implementation procedure is straightforward. The CCI is calculated in the inverted version of the original image, and the contrast-based dark channel is used to produce two atmospheric light models with *m* values of 5 and 30, as previously discussed. Two transmission maps are then calculated based on different atmospheric light models, and two haze-free images are produced. Finally, three weight maps are calculated as inputs to the multi-scale fusion Gaussian pyramid, and the images are combined to generate a high-quality image.

A. *Development Environment*

The L2UWE framework was implemented and tested using Python in the Google Colab environment. Python is a popular language for image processing and artificial intelligence due to its ease of use and flexible, efficient libraries. Google Colab is a free cloud-based IDE that facilitates collaboration and eliminates the need for local environment setup. It is also a high-performance computing platform that provides access to powerful hardware for computing tasks. The implementation used several libraries, including sys, os, openCV2, numpy, matplotlib, scipy.ndimage, and math, which were pre-installed and made the entire development process more straightforward.

B. *Dataset*

There are several datasets available online for underwater image processing, including UVP, UIEBD, SQUID, U-45, RUIE benchmark, Jamaica Port Royal, Virtual Periscope, TURBID, OceanDark, and USR-248. Among these, the EUVP dataset is one of the largest and most commonly used for underwater video processing, with over 2000 hours of footage captured at various locations in Europe. Other datasets differ in location, size, underwater scenes, and lighting conditions. Tunai *et al*. [2] used the OceanDark dataset in their paper, and this dataset was also used in the development of this project for comparison purposes. However, the actual testing and demonstration of the algorithm will be done using the USR-248 dataset, which offers a wider range of underwater scenes with different illumination levels and image qualities. This dataset was captured using a high-quality underwater camera and offers images of different sizes which makes testing easier. Additionally, the USR-248 dataset is a readily available resource that can be easily downloaded for free. In contrast, OceanDark requires an online request and waiting time for image resources. During the development and actual testing of this project, online OceanDark and USR-248 datasets were utilized due to time constraints. It is recommended to conduct further tests using a variety of datasets if time permits. However, this project has already completed testing with over 100 input images, resulting in convincing outcomes. The use of a large amount of input images contributed to the accuracy of the results.

C. *Pre-processing*

The L2UWE framework computes the local contrast map and CCI values by utilizing the standard deviation of the image pixels in a local neighborhood. This statistical measure provides a reliable assessment of the amount of variation or spread in a set of values, making it a suitable indicator for the local contrast in the image. To create the local contrast map, the input image is first converted to grayscale and inverted. Next, the inverted image is divided into overlapping patches, and the standard deviation of each patch is calculated. The standard deviation values are then used to construct the local contrast map, which has the same dimensions as the input image. The L2UWE framework uses the local contrast map to enhance low-light underwater images through a multi-scale fusion process. By combining the original input image with the local contrast map at different scales, the framework is able to capture both global and local contrast information in the image, and maintain the overall structure while enhancing the contrast. Overall, the use of standard deviation in the L2UWE framework provides an effective way to compute the local contrast map and enhance low-light underwater images. By utilizing the standard deviation of the image pixels in a local neighborhood, the framework can accurately capture the amount of variation in the pixel values and employ this information to enhance the contrast in the image. Figure 7 presents the original and pre-processed images.

Figure 7. Original input image(left).Inverted image(middle). Grayscale of the inverted image(right).

D. *Contrast Code Image*

In their work, Tunai et al. [2] proposed using CCI matrices to store integer codes c ($1 \leqslant c \leqslant 7$) that are initialized to 7 before modification. The size of each patch is determined by the expression (2c + 1) x (2c + 1) instead of fixed-sized patches where c represents the CCI value at a given pixel location. his dynamic patch sizing allows for more accurate calculation of the transmission map and dark channel. To achieve greater accuracy, the local standard deviation is computed using different window sizes, {15,13,11,9,7,5,3}(ranging from 15x15 to 3x3), that represent various patch sizes. The local standard deviation is a measure of local contrast and is computed over a local neighborhood of pixels, rather than the entire image. The computation involves moving a sliding window over the image and computing the standard deviation of pixel values within the window at each location, with the final result stored at the center of the location. The tolerance parameter *t* is introduced to represent the percentage by which a subsequent standard deviation has to be smaller than the previous one to trigger a new code *c* to be stored in the CCI. A 5% tolerance is recommended by Marques *et al*. [4]. The algorithm was tested on Figure 1, and the resulting dynamic c values were obtained and used to produce the CCI result shown in Figure 8. The proposed approach models areas of high contrast by considering larger regions, while areas of low contrast are illuminated uniformly with smaller patches studied to model local illumination.

```
[ ] CCI = np.zeros((x,y))
    for height in range(y):
        for width in range(x):
            # #init CCI value to 7
            # CCI[width][height] = 7

            #find the smallest std. dev.
            smallest_std_index = np.argmin(multiplication[width,height,:])
            # CCI[width][height] = c[smallest_std_index]
            CCI[width][height] = smallest_std_index+1
            #CCI[width][height] = psize[smallest_std_index]
    print(CCI)

    [[6. 7. 7. ... 7. 7. 7.]
     [6. 7. 7. ... 6. 7. 7.]
     [6. 6. 7. ... 7. 7. 7.]
     ...
     [7. 7. 7. ... 3. 2. 1.]
     [7. 7. 6. ... 3. 2. 1.]
     [7. 7. 6. ... 3. 2. 1.]]
```

Figure 8. The actual CCI result of the tested image(Figure 1).

E. *The Haze-less Image*

As highlighted in the technical review section, accurate calculation of the atmospheric lighting model and transmission map is essential to obtain a haze-free image. Equation 7 describes the relationship between patch size, contrast code *c*, and the multiplier *m*, which enables us to compute the atmospheric light for each color channel based on local contrast. The multiplier value of 5 and 30 produces two distinct models.

Once the atmospheric lighting model is derived, it must be normalized before computing the transmission map. Normalization is crucial to ensure that the transmission map accurately reflects the scene's transmission properties, without being influenced by the atmospheric lighting. The atmospheric lighting model is influenced by atmospheric conditions such as haze or fog, and can vary considerably. Accounting for such variations is vital when calculating the transmission map using the atmospheric lighting model to obtain an accurate estimate of the scene's transmission properties. Normalizing the atmospheric lighting model involves dividing the raw lighting model by a global maximum value, resulting in a normalized lighting model with a maximum value of 1. The resulting output image will have a range of 0 to 1, rather than 0 to 255. This normalization step eliminates any variations in the atmospheric lighting model that are unrelated to the scene's properties and ensures that the transmission map is not influenced by these variations. A normalized atmospheric lighting model leads to an accurate transmission map, reflecting the true attenuation of light due to scene objects. This is particularly important in image enhancement applications, where the transmission map is used to enhance image contrast. A more accurate transmission map results in improved contrast enhancement, which enhances the image's visual quality and facilitates subsequent image analysis tasks.

Figure 8 illustrates the two processed transmission maps. Using a larger patch size can cause the transmission map to be more blurred, as it captures the transmission properties of a larger image region. This blurring effect is due to the averaging of the transmission values over a larger area, which leads to a loss of detail and a smoothing of the transmission map. In contrast, using a smaller patch size can produce a more detailed transmission map, with sharper edges and finer-grained variations in the transmission values. The smaller patch size captures more local details and texture variations in the image, leading to a more precise estimation of the transmission properties. This effect was previously discussed in the "Contrast-guided approach for the enhancement of low-light images" section.

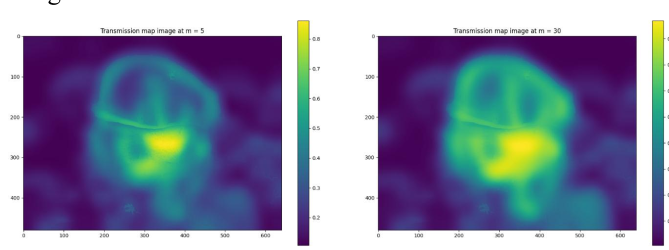


Figure 8. Transmission map when m = 5 (left). Transmission map when m = 30 (right).

Obtaining a haze-free image has become more manageable. To recover the radiance, it is essential to ensure that the input image values are in the range of 0 to 1. If the input image falls outside of this range, it is still necessary to rescale the intensities of each pixel in the three channels. To recover the radiance, we need to iterate through the image and subtract the original pixel value by the atmospheric light. Then, we divide the result by the maximum value of the transmission map among all channels, and finally, add the result to the atmospheric light. We have obtained two haze-less images in Figure 9, processed with small and large patch sizes. The image processed with a smaller patch size provides more details of objects, while the one processed with a larger patch size is brighter and has a clear background, but it misses the sharp edges of the objects.

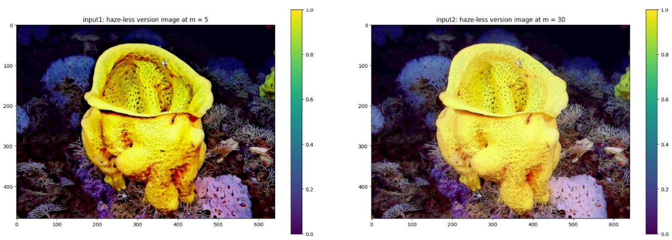
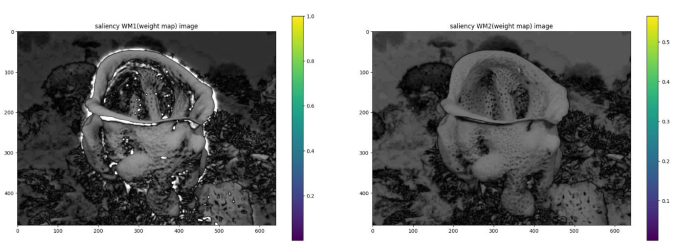


Figure 9. Haze-less image when m = 5 (left). Haze-less image when m = 30 (right).

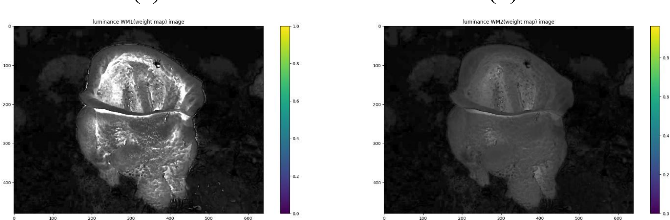
F. *Image Reconstruction by Combining the Generated Haze-less Images*

To enhance an image effectively, it is essential to capture its most significant features. In the L2UWE framework, this is achieved by fusing the enhanced base and detail components at multiple scales using a weighted average method. The multi-scale fusion method plays a significant role in combining the input images and improving the overall image quality.

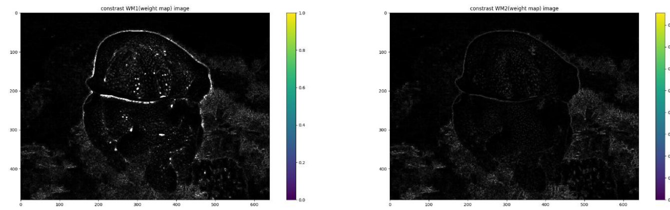
To determine the weights used in the weighted average method, the multi-scale fusion approach employs saliency, luminance, and contrast weight maps. These maps are generated by calculating the saliency weight using the Achantay method and the contrast weight by taking the absolute value of a Laplacian filtering on the luminance. All the relevant equations are discussed in the Multi-scale Fusion for Image Enhancement section of the framework. Figure 10 in the paper provides a visual representation of the output weight maps at different patch sizes.



(a) (b)



(b) (d)



(e) (f)

Figure 10. Saliency weight map at m = 5 (a). Saliency weight map at m = 30 (b). Luminance weight map at m = 5 (c). Luminance weight map at m = 30 (d).Contrast weight map at m = 5 (e). Contrast weight map at m = 30 (f).

Once the weight maps are generated, they are normalized before performing the Gaussian pyramid. The fused pyramids are then calculated using the normalized weight map, with the Gaussian pyramid being one-channel, while the Laplacian pyramid of the original image derived inputs has three channels. To create the final enhanced image, each value in the Gaussian pyramid is multiplied by the three values in a given location of the Laplacian pyramid. The L2UWE framework's use of weighted averaging and multi-scale fusion is a significant advancement in underwater image enhancement technology. By incorporating saliency, luminance, and contrast weight maps, the framework can identify the most significant regions of the image and enhance them effectively. The resulting image retains its original structure while being visually more appealing and suitable for analysis. The final output, along with a comparison image, is presented in Figure 11. A noticeable difference can be seen between the original and enhanced images. The enhancement process brings out all the details that were not visible in the original image. The original image appears extremely dark, making it difficult to identify the background objects. However, the enhanced image is much brighter, and the objects can be observed clearly without losing their features. The image enhancement process successfully improves the visual quality of the image, making it more useful for subsequent analysis tasks.

Figure 11.Original image (left). Enhanced image (right).

IV. EVALUATION METRICS

In order to assess the effectiveness of the L2UWE framework, Tunai et al. [2] employed several evaluation standards, such as UIQM, PCQI, GCF, e- and r-scores, FADE, and SURE. However, due to time constraints, it may be challenging to implement and verify these standards using Matlab. Fortunately, the Queen Mary University of London [7] has provided an online platform for evaluating underwater image quality that employs UCIQE, UIQM, and CCF. UCIQE is a metric that utilizes a set of color correction and image enhancement algorithms to simulate the visual effects of underwater imaging, such as color shift, haze, and contrast reduction. It quantifies the color distortion and overall quality of enhanced underwater images. Meanwhile, UIQM and CCF evaluate image quality based on factors such as sharpness, brightness, colorfulness, and contrast. Figure 12 depicts the screenshot result after Figure 1 has been enhanced and processed. The improvements can be observed from the indicators, which represent enhancements in all aspects, such as contrast, color, and fidelity.

Image	UCIQE	UIQM	CCF
	0.67765	0.88648	43.11512
	0.71728	1.07452	65.01080

Figure 12.Original image (top). Enhanced image (bottom).

Figure 13 displays a collection of before-and-after processed images. It is noticeable that the enhanced results are exceptionally remarkable when the background is generally dark. However, when the background is more greenish, the enhanced image may present a more greenish tint. This suggests that color correction can be further improved in future work. Unfortunately, the underlying reasons that cause this effect are not apparent and require further investigation. Due to the time restriction, the reason can be further investigated in the future.

Figure 13.Original image (left). Enhanced image (right).

V. CONCLUSION

This project investigated the effectiveness of the L2UWE framework in enhancing low-light underwater images efficiently. The L2UWE framework employs local contrast enhancement and multi-scale fusion techniques to produce high-quality images with better color and detail preservation. Although there was no comparison with other state-of-the-art methods, the evaluation results of UCIQE, UIQM, and CCF demonstrate that the L2UWE framework significantly enhances the quality of the underwater images. Therefore, we can conclude that this project successfully applied the L2UWE framework to enhance underwater images and achieved satisfying results that meet the project's objectives. The greenish tint issue remains unsolved, but given that this project was conducted by an individual with no prior experience in Python, the result is acceptable and can be improved in future work.

REFERENCES

[1] “USR-248 dataset,” *USR-248 Dataset | Minnesota Interactive Robotics and Vision Laboratory*. [Online]. Available: https://irvlab.cs.umn.edu/resources/usr-248-dataset.

[2] T. P. Marques and A. B. Albu, “L^2UWE: A framework for the efficient enhancement of low-light underwater images using local contrast and multi- scale fusion,” arXiv.org, 05-Nov-2020. [Online]. Available: https://arxiv.org/abs/2005.13736.

[3] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. IEEE transactions on pattern analysis and machine intelligence, 33(12):2341–2353, 2011. 2, 3, 4

[4] Tunai Porto Marques, Alexandra Branzan Albu, and Maia Hoeberechts. A contrast-guided approach for the enhancement of low-lighting underwater images. Journal of Imaging, 5(10):79, 2019. 1, 2, 3, 4, 5, 7, 8

[5] “Under Water Image Enhancement by Fusion,” UGC Approved Journal. [Online]. Available: http://www.ijmer.com/pages/Vol.4-Iss.4(Version-4).html.

[6] Codruta Orniana Ancuti and Cosmin Ancuti. Single image dehazing by multi-scale fusion. IEEE Transactions on Image Processing, 22(8):3271–3282, 2013. 1, 2, 4, 6

[7] “Platform for underwater image quality evaluation,” PUIQE. [Online]. Available: https://puiqe.eecs.qmul.ac.uk/.