# Comprehensive Framework for Advanced Underwater Image Enhancement Using Multi-Scale Fusion and Color Balancing

Aashi Jain, Aditi Agarwal, Rizul Gupta, Priyansh Trivedi
Department of Electronics and Communication Engineering
Indian Institute of Technology Roorkee, India
Email: aashi_j@ece.iitr.ac.in, aditi_a@ece.iitr.ac.in, rizul_g@ece.iitr.ac.in, priyansh_t@ece.iitr.ac.in
Enrollment No.: 22116001, 22116004, 22116080, 22116074

***Abstract*—Underwater imaging suffers from challenges like color distortion, haze, and low contrast caused by light scattering and absorption. This paper proposes a robust framework combining white balance correction, gamma adjustment, unsharp masking, and multi-scale fusion techniques. These processes address specific underwater image artifacts, significantly improving visibility and color fidelity. The pipeline is validated using Python-based implementations, showcasing applicability to underwater conditions.**

## I. Introduction

Underwater image processing addresses challenges such as poor visibility and unnatural hues caused by selective light absorption and scattering. These degradations impact applications in marine biology, archaeology, and robotics.

### A. Challenges in Underwater Imaging

The principal causes of image degradation underwater are:

1) **Light Absorption:** Red wavelengths are absorbed first, leaving images dominated by blue-green tones.
2) **Light Scattering:** Suspended particles scatter light, reducing contrast and introducing haze.
3) **Backscatter:** Particles reflect stray light, creating a veiled effect that obscures objects.

## II. Proposed Framework

The proposed framework, shown in Figure 1, consists of multiple enhancement techniques designed to address underwater imaging issues.

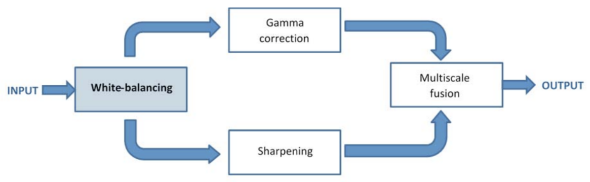


Fig. 1: Pipeline for underwater image enhancement. Major stages include preprocessing, gamma adjustment, unsharp masking, white balance correction, and multi-scale fusion.

### A. Preprocessing and White Balance Correction

Underwater images often suffer from excessive green and blue tones due to light absorption and scattering in water. A channel-wise correction algorithm is applied to mitigate this, modifying the red and blue channels based on the green channel, which is typically the least affected by attenuation. The white balance correction is formulated as follows:

$$I_{rc}(x) = I_r(x) + \alpha(\overline{I_g} - \overline{I_r})(1 - I_r(x))I_g(x), \quad (1)$$

$$I_{bc}(x) = I_b(x) + \alpha(\overline{I_g} - \overline{I_b})(1 - I_b(x))I_g(x), \quad (2)$$

where $I_{rc}(x)$ and $I_{bc}(x)$ are the corrected red and blue channels, $\alpha$ controls the correction intensity, and $\overline{I_g}$ represents the mean intensity of the green channel. This method assumes that the green channel retains the most natural color information in underwater environments, as it is less impacted by the selective absorption of light in water.

#### 1) Implementation Details

The algorithm works by computing the mean intensities of the red, green, and blue channels, then using the green channel's mean intensity to adjust the red and blue channels:

- **Red and Blue Channel Adjustment:** The red and blue channels are adjusted based on their mean values and the green channel's mean value. This correction helps restore the natural color balance in underwater images.
- **Alpha Control:** The correction intensity is controlled by the parameter $\alpha$, which is adjusted based on the image's quality and the severity of color distortion.

#### 2) Theoretical Justification

The green channel is less affected by light absorption and scattering in underwater environments, as it is less susceptible to attenuation than the red and blue channels. This preprocessing step ensures the underwater image has a more natural color balance, improving the overall visual quality. By adjusting the red and blue channels based on the green channel, this method compensates for the selective loss of color information at different depths in the water.

#### 3) Application in Underwater Image Enhancement

This white balance correction is a crucial first step in the enhancement pipeline. Restoring color balance prepares the

image for further processing, such as edge enhancement and contrast adjustment. The method is designed to be robust across various underwater imaging conditions, including varying water depths and lighting conditions, which may affect the image's color distribution.

### B. Gamma Adjustment for Illumination Correction

Gamma correction adjusts the intensity values of an image using a power-law function:

$$s = cr^{\gamma}, \tag{3}$$

where $r$ is the normalized input intensity ($0 \leq r \leq 1$), $s$ is the output intensity, $c$ is a scaling constant (typically set to 1), and $\gamma$ is the gamma exponent.

The transformation modifies brightness based on $\gamma$: $\gamma < 1$ brightens the image, while $\gamma > 1$ darkens it. For $\gamma = 1$, the image remains unchanged. The corrected intensity values are computed as follows:

$$I_{\text{gamma}} = 255 \cdot \left(\frac{I}{255}\right)^{\gamma}, \tag{4}$$

ensuring accurate reproduction of the image by compensating for the non-linear response of imaging devices.

Gamma correction is essential in image processing because many imaging devices (e.g., monitors and cameras) have a non-linear response to intensity. Without correction, the displayed image may appear either too dark or too bright, failing to represent the true appearance of the scene. The intensity response is linearized by applying gamma correction, ensuring accurate reproduction of the original image. This correction maintains perceptual consistency across different devices.

### C. Unsharp Masking for Edge Enhancement

Unsharp masking sharpens details by amplifying high-frequency components, making it particularly effective for enhancing edges in underwater images degraded by scattering and absorption. This technique can be implemented in both spatial and frequency domains.

#### 1) Spatial Domain Implementation

In the spatial domain, the sharpened image is computed as:

$$I_{\text{sharp}} = I + \beta(I - G * I), \tag{5}$$

where $G * I$ is the Gaussian-blurred version of the image, and $\beta$ controls the sharpening intensity. A linearly normalized variation (histogram-equalized) of this formula ensures consistent sharpening without oversaturation:

$$I_{\text{sharp}} = \frac{I + \text{hist\_eq}(I - G * I)}{2}. \tag{6}$$

This approach reduces over-enhancement in regions with already high-frequency details, as highlighted in the study.

#### 2) Frequency Domain Implementation

Unsharp masking in the frequency domain enhances high-frequency components by subtracting a smoothed version of the image. The mask is computed as:

$$g_{\text{mask}}(x, y) = f(x, y) - f_{\text{LP}}(x, y), \tag{7}$$

where $f(x, y)$ is the original image and $f_{\text{LP}}(x, y)$ is the lowpass-filtered image.

Using the discrete Fourier transform (DFT), the lowpass-filtered image is:

$$f_{\text{LP}}(x, y) = \mathcal{F}^{-1}\{H_{\text{LP}}(u, v)F(u, v)\}, \tag{8}$$

where $H_{\text{LP}}(u, v)$ is the transfer function of the lowpass filter, $F(u, v)$ is the DFT of the input image, and $\mathcal{F}^{-1}$ denotes the inverse DFT.

The sharpened image in the frequency domain is:

$$G(u, v) = [1 + (1 - H_{\text{LP}}(u, v))]\, F(u, v), \tag{9}$$

where $(1 - H_{\text{LP}}(u, v))$ acts as a highpass filter. The choice of $H_{\text{LP}}(u, v)$ influences the smoothing and sharpening effects.

#### 3) Histogram Equalization for Improved Masking

Histogram equalization is applied to enhance the contrast of the mask $g_{\text{mask}}$ before blending it with the input image. This step ensures better edge definition and prevents regions from appearing overly flat:

- **Purpose:** To enhance the contrast of the high-frequency mask, making edges more prominent.
- **Method:** Each mask channel is equalized independently to normalize the intensity distribution and prevent color distortions.

As observed in the research paper, the histogram equalization step also complements the unsharp masking process by reducing artifacts caused by uneven lighting.

#### 4) Implementation Details

The Python implementation adopts the following approach:

- A Gaussian blur is applied to compute a smoothed version of the image.
- The high-frequency mask is enhanced using histogram equalization.
- The sharpened image is computed as:

$$I_{\text{sharp}} = \frac{I + g_{\text{mask}}}{2}, \tag{10}$$

  where $g_{\text{mask}}$ is scaled for optimal visibility.
- Parameters such as blur kernel size, Gaussian sigma, sharpening amount, and threshold are fine-tuned for underwater images.

#### 5) Integration with Fusion-Based Enhancement

The unsharp masking technique is an integral component of the fusion-based enhancement strategy described in the research paper. It helps preserve fine details by:

- Enhancing edges lost due to scattering.
- Suppressing noise while boosting significant features.

The sharpening process is particularly effective when combined with gamma correction and weight map blending in the multi-scale fusion process.

#### 6) Theoretical Justification

The research highlights that combining unsharp masking with histogram equalization improves the visibility of edges while maintaining the natural appearance of underwater scenes. By compensating for the scattering and absorption effects, this method ensures artifact-free sharpening that enhances the final reconstructed image.

### D. Multi-Scale Fusion Process

The multi-scale fusion process combines weight maps derived from gamma-adjusted and unsharp-masked images, designed to preserve edges and enhance color contrast while addressing underwater image degradations. The process is implemented using Gaussian and Laplacian pyramids for multi-scale blending.

- **Weight Maps:** Edge sharpness and saturation weight maps are computed for gamma-adjusted and sharpened images. The edge map is calculated using a Laplacian operator, and the saturation map measures pixel-wise deviation from the mean channel value:

$$W_{\text{sat}}(x) = \sqrt{\frac{1}{3}\sum_{c}\left(I_c(x) - \overline{I}(x)\right)^2}, \tag{11}$$

  where $I_c(x)$ is the intensity of channel $c$ at pixel $x$, and $\overline{I}(x)$ is the mean intensity across all channels.
  Combined weight maps for sharp and gamma images are normalized as:

$$W_{\text{sharp}}(x) = \frac{W_{\text{sharp,edge}}(x) + W_{\text{sharp,sat}}(x) + \delta}{\text{denominator}}, \tag{12}$$

$$W_{\text{gamma}}(x) = \frac{W_{\text{gamma,edge}}(x) + W_{\text{gamma,sat}}(x) + \delta}{\text{denominator}}, \tag{13}$$

  where $\delta$ is a small regularization term, and the denominator is the sum of sharp and gamma weight maps.
- **Pyramid Construction:** Gaussian pyramids are built for weight maps using iterative down-sampling, while Laplacian pyramids are constructed for each color channel by subtracting successive Gaussian levels. The Laplacian pyramid for image $I$ at level $l$ is:

$$L_l\{I(x)\} = G_l\{I(x)\} - G_{l+1}\{I(x)\}, \tag{14}$$

  where $G_l$ denotes Gaussian filtering at level $l$.
- **Blending:** At each pyramid level, inputs are fused using normalized weights:

$$R_l(x) = \sum_{k=1}^{K} G_l\{\overline{W}_k(x)\} L_l\{I_k(x)\}, \tag{15}$$

  where $G_l$ and $L_l$ represent Gaussian and Laplacian components, and $R_l(x)$ is the reconstructed image at level $l$.
- **Reconstruction:** The final enhanced image is reconstructed by iteratively up-sampling and summing Laplacian pyramid layers:

$$I_{\text{final}} = \sum_{l=1}^{N} R_l(x), \tag{16}$$

  ensuring smooth blending and minimal artifacts.

#### 1) Implementation

Key steps include:

- **Edge and Saturation Maps:** The edge map is computed using the Laplacian operator, and the saturation map is derived by measuring pixel-wise deviation from the mean channel intensity.
- **Weight Map Calculation:** Combined weight maps for both sharp and gamma-adjusted images are computed by summing the edge and saturation weight maps, followed by normalization.
- **Pyramid Construction:** Gaussian and Laplacian pyramids are constructed for the image and weight maps to decompose the input images and facilitate multi-scale fusion.
- **Blending and Reconstruction:** The Laplacian pyramid layers are blended at each level using the normalized weight maps, and the final enhanced image is obtained by iteratively reconstructing the image from these layers.

With minimal manual parameter tuning, this computational framework ensures robust contrast enhancement, edge sharpness, and color fidelity in underwater images.

## III. Experimental Results

To validate it, we tested the proposed framework on a diverse set of underwater images suffering from varying degrees of degradation. The results demonstrate significant color balance, contrast, and overall visibility improvements.

Figure 2 illustrates the results for eight underwater images. We provide the original input, intermediate outputs (after white balance correction and unsharp masking ), and the final enhanced output for each image. This comprehensive display helps visualize the contributions of each stage in the enhancement pipeline.

Due to space constraints, the detailed results and intermediate stages are presented on the following page.

## IV. Conclusion

This paper presents an integrated framework for underwater image enhancement. Combining white balance correction, gamma adjustment, unsharp masking, and multi-scale fusion significantly improves clarity and color fidelity. The Python-based implementation demonstrates the pipeline's practical utility, paving the way for real-time and adaptive enhancements.

## V. References


1) J. Y. Chiang and Y. C. Chen, "Underwater Image Enhancement by Wavelength Compensation and Dehazing," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1756–1769, 2012.
2) C. O. Ancuti et al., "Color Balance and Fusion for Underwater Image Enhancement," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 379–393, 2018.


## VI. Contributions

- **Aashi Jain:** Implemented Histogram Equalization and Unsharp Masking techniques to enhance image clarity and detail.
- **Aditi Agarwal:** Led the preprocessing pipeline, including White Balance Correction, Gamma Correction and developed one final pyramid of the image
- **Rizul Gupta:** Designed the Gaussian Blur function for Unsharp Masking, calculated the Weight Maps, and reconstructed the final enhanced image from the computed pyramid.
- **Priyansh Trivedi:** Developed Gaussian and Laplacian Pyramid functions, resolved integration issues and debugged the overall code to ensure seamless functionality.

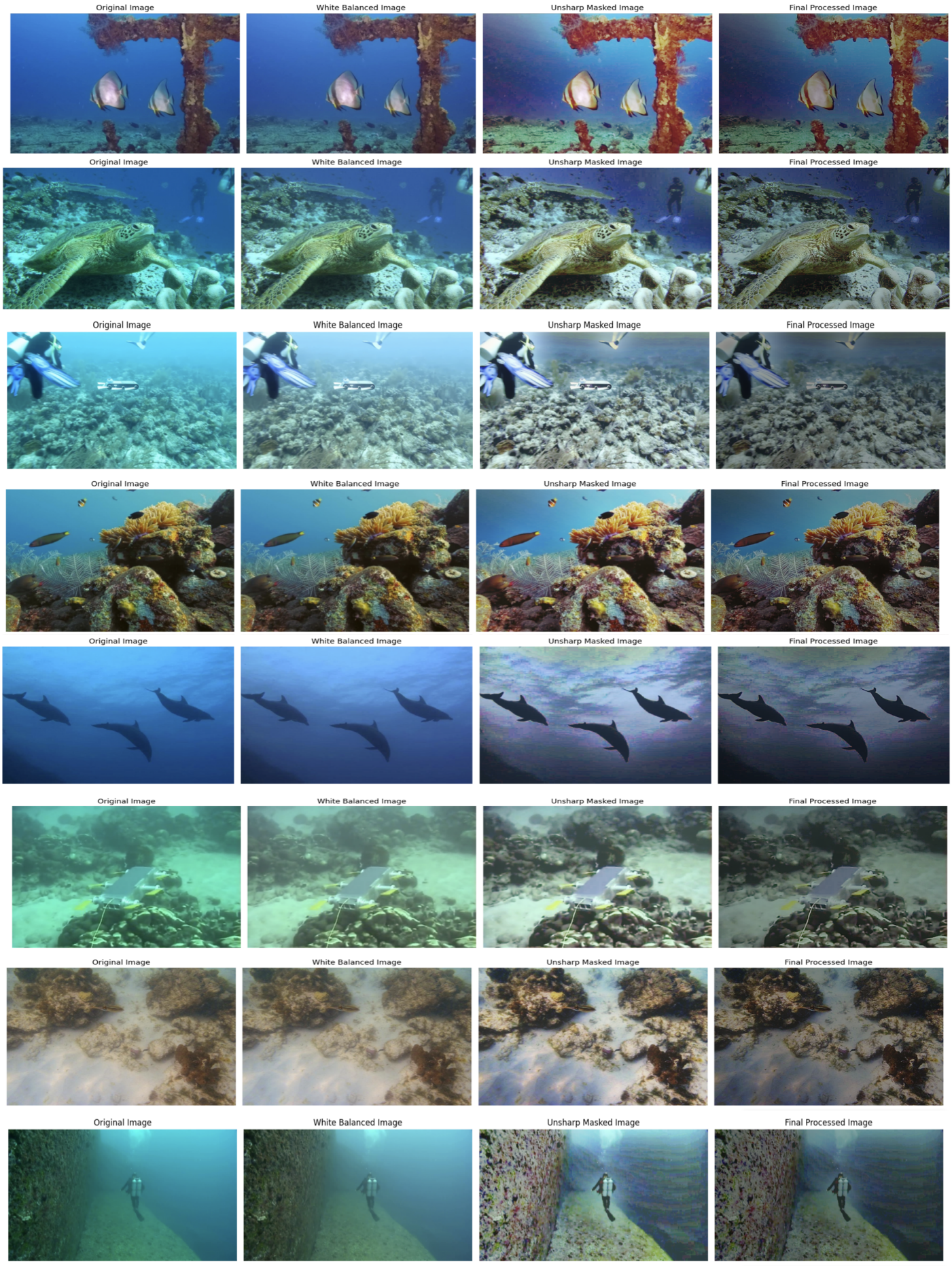


Fig. 2: Results