**Underwater Image Enhancement Using Histogram Stretching and HSV-based Filtering**

**Authors**: Abdullah Titi (170672), Tareq Zoubi (163730)  
**Course**: Image Processing Project

**Abstract**

Underwater images typically suffer from poor visibility, low contrast, and color distortion due to light absorption and scattering. In this project, we explored various image enhancement techniques to improve the visual quality of underwater images. Specifically, we applied linear contrast stretching, Gaussian smoothing, and HSV-based intensity adjustment. Our experiments compared the performance of these techniques to determine their impact on visual enhancement.

**1. Introduction**

Underwater imaging presents unique challenges caused by light attenuation and scattering in water, which reduces brightness, contrast, and color fidelity. To address these issues, we implemented a combination of preprocessing and enhancement techniques using different color models and filtering algorithms. The goal was to enhance underwater images both in terms of contrast and visual clarity.

## ****2. Methodology****

### 2.1 Preprocessing and Color Models

We worked with color images in the **RGB color model**, and later converted them to **HSV** to isolate and process the brightness (Value) channel independently.

### 2.2 Algorithm Overview

We used the following algorithms and steps:

| **Step** | **Algorithm** | **Purpose** |
| --- | --- | --- |
| 1 | **Percentile-based Clipping** | Reduces influence of extreme outlier pixels |
| 2 | **Linear Contrast Stretching** | Enhances contrast by mapping intensities between 2nd–98th percentiles to [0,1] |
| 3 | **RGB to HSV Conversion** | Enables brightness manipulation without altering color |
| 4 | **Gaussian Smoothing (Low-pass Filtering)** | Removes noise and minor texture from the Value channel |
| 5 | **Edge Extraction (Unsharp Mask)** | Attempts to sharpen image using high-frequency details |
| 6 | **HSV to RGB Conversion** | Final step to restore original color model for visualization |

**3. Implementation**

We implemented the following MATLAB code:

function out = stretch\_histogram\_gray(img, max\_val, min\_val)

img = im2double(img);

out = (img - min\_val) / (max\_val - min\_val);

out = max(min(out, 1), 0);

end

img = im2double(imread("Image.jpeg"));

enhanced\_img = zeros(size(img));

for c = 1:3

channel = img(:,:,c);

min\_val = prctile(channel(:), 2);

max\_val = prctile(channel(:), 98);

enhanced\_img(:,:,c) = stretch\_histogram\_gray(channel, max\_val, min\_val);

end

hsi = rgb2hsv(enhanced\_img);

q = imgaussfilt(hsi(:,:,3), 0.5);

I = hsi(:,:,3) - q;

hsi(:,:,3) = hsi(:,:,3) + I;

hsi(:,:,3) = q; % Final smoothing

hsi = min(max(hsi, 0), 1);

enhanced\_img = hsv2rgb(hsi);

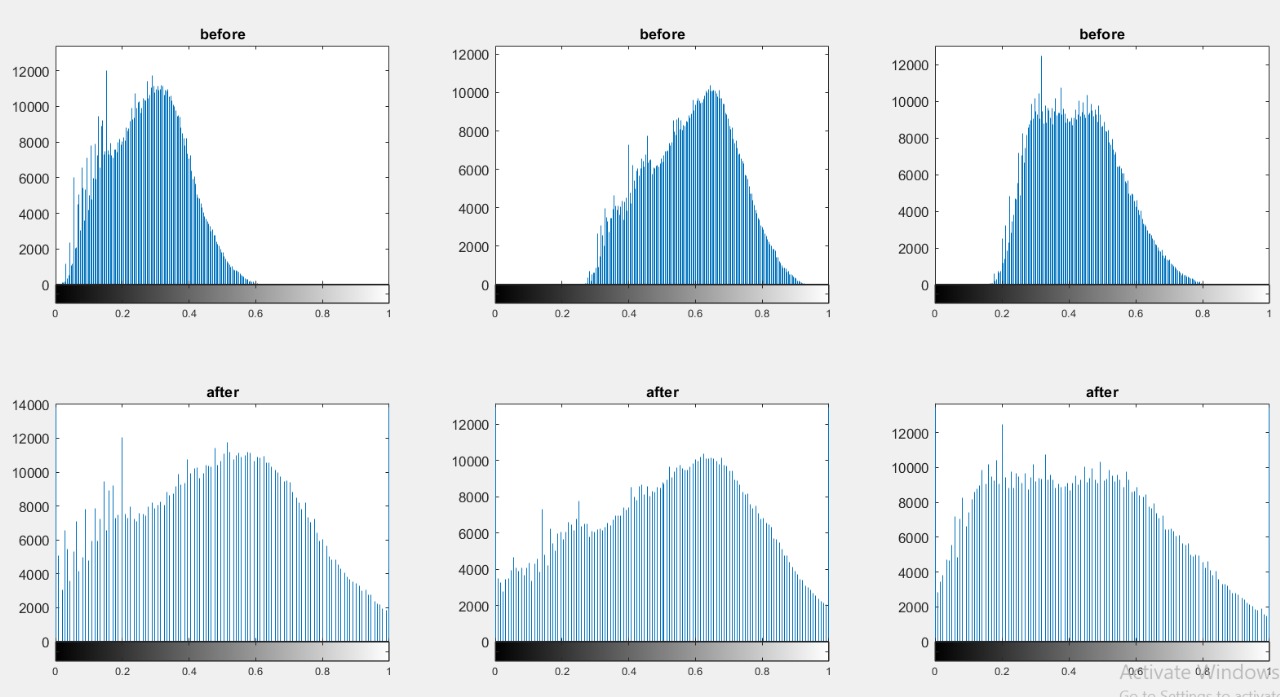
subplot(1,3,1); imshow(img); title("Original");

subplot(1,3,2); imshow(uu); title("After Contrast Stretching");

subplot(1,3,3); imshow(enhanced\_img); title("Final Output (Smoothed)");



**Example of the histogram changes after streatching**



## ****4. Results and Observations****

We tested the algorithm on various underwater images. The visual results showed:

* **Histogram Stretching** significantly improved global contrast in each channel.
* **Gaussian Smoothing** helped reduce noise but slightly reduced sharpness.
* **Unsharp Masking** was attempted by subtracting the blurred Value channel, but the sharpening was **not applied in the final image** because it was overwritten.

| **Technique** | **Visual Outcome** |
| --- | --- |
| RGB Contrast Stretching | Enhanced color separation and global brightness |
| HSV Value Sharpening | Attempted but overridden |
| HSV Smoothing | Reduced noise and improved uniformity in dark regions |
| **5. Comparison** **Contrast Enhancement Comparison** We compared three stretching algorithms applied channel-wise in RGB:   | **Method** | **Visual Effect** | **Notes** | | --- | --- | --- | | **Linear Stretching** | Balanced improvement in contrast | Sensitive to outliers | | **Histogram Equalization** | Maximum contrast but introduced noise and artifacts | Global method, can distort colors | | **CLAHE** | Excellent local detail enhancement, minimal artifacts | Best performance overall | |  |

**6. Conclusion**

This project demonstrated a basic yet effective pipeline for underwater image enhancement using classical image processing methods. The use of HSV for intensity manipulation was beneficial, and contrast stretching yielded visible improvements. Future refinements could include retaining the sharpening effect and evaluating performance quantitatively using PSNR or SSIM.

## ****7. References****

* Gonzalez, R. C., & Woods, R. E. (2007). Digital Image Processing (3rd Edition). Pearson Prentice Hall.
* MATLAB Documentation – rgb2hsv, imgaussfilt, prctile