



## **Lab Report 02**

**Submitted by:**

Name: Shaikat Hazra Pranto

ID: 2020-1-60-072

Course: CSE438

Section: 2

**Submitted To:**

Md Ashraful Haider Chowdhury

Lecturer

Department of Computer Science and Engineering

**Submission date:**

4<sup>th</sup> March, 2025

**Problem 01:**

```
clear all;
```

```
close all;
```

```
% Read the image
```

```
I = imread('Picture1.jpg');
```

```
stretched_img = imadjust(I, stretchlim(I, [0.01, 0.99]), []);
```

```
subplot(1,2,1); imshow(I); title('Original Image');
```

```
subplot(1,2,2); imshow(stretched_img); title('Contrast Stretched Image');
```

```
% ID: 2020-1-60-072
```

**Original Image**



**Contrast Stretched Image**



**Problem 02:**

```
clear all;
close all;
clc;

% Read the image
I = imread('Picture1.jpg');

% Convert to grayscale if the image is RGB
if size(I, 3) == 3
    I = rgb2gray(I);
end

% Get the correct size of the grayscale image
[rows, cols] = size(I);

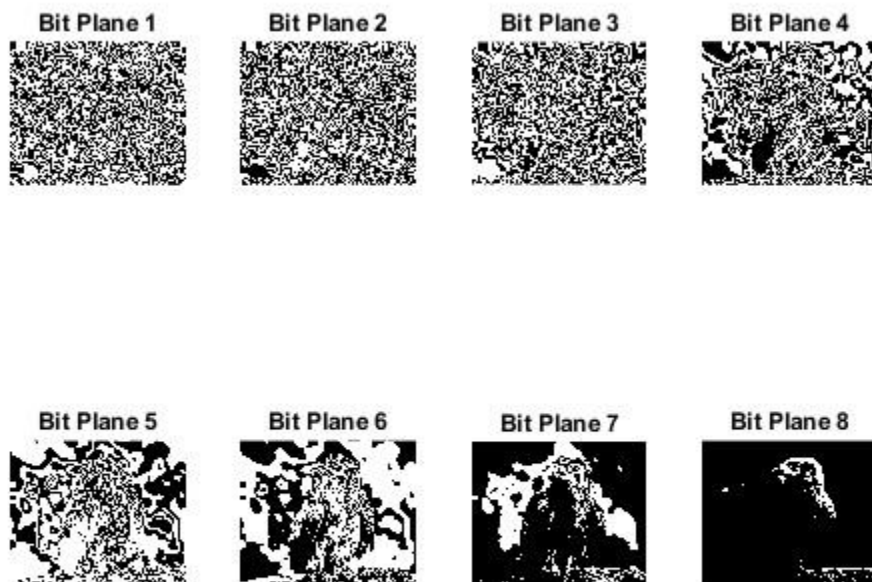
% Create an array to store bit planes
bit_planes = zeros(rows, cols, 8, 'uint8');

% Extract bit planes
for i = 1:8
    bit_planes(:, :, i) = uint8(bitget(I, i) * 255); % Extract i-th bit and scale to 0-255
end

% Display the bit planes
figure;
for i = 1:8
    subplot(2,4,i);
    imshow(bit_planes(:, :, i)); % Show bit plane
```

```
title(['Bit Plane ', num2str(i)]);  
end
```

% ID: 2020-1-60-072



### Problem 03:

```
clear all;
```

```
close all;
```

```
clc;
```

```
% Read the image
```

```
I = imread('Picture2.png');
```

```
% Convert to grayscale if RGB
```

```
if size(I, 3) == 3
```

```
    I = rgb2gray(I);
```

```
end
```

```
% Convert to double for transformations
```

```
I_double = double(I);
```

```
% Logarithmic Transformation
```

```
c_log = 255 / log(1 + double(max(I(:))))); % Compute scaling constant
```

```
I_log = c_log * log(1 + I_double); % Apply transformation
```

```
I_log = uint8(I_log); % Convert back to uint8
```

```
% Power-law (Gamma) Transformation
```

```
gamma = 0.5; % Experiment with values (e.g., 0.5, 1.5, 2.0)
```

```
I_gamma = 255 * ((I_double / 255) .^ gamma); % Apply power-law transformation
```

```
I_gamma = uint8(I_gamma); % Convert back to uint8
```

```
% Display Results
```

```
figure;
```

```
subplot(1,4,1);
```

```
imshow(I);
```

```
title('Original Image');
```

```
subplot(1,4,2);
```

```
imshow(I_log);
```

```
title('Logarithmic Transformation');
```

```
subplot(1,4,3);
```

```
imshow(I_gamma);
```

```
title(['Power-law Transformation']);
```

% ID: 2020-1-60-072



**Problem 04:**

```
clear all;
close all;
clc;

% Read the images
I = imread('Picture3.jpg');
Ref = imread('Picture4.png');

% Convert images to grayscale if they are RGB
if size(I, 3) == 3
    I = rgb2gray(I);
end
if size(Ref, 3) == 3
    Ref = rgb2gray(Ref);
end

% Perform histogram matching
I_matched = imhistmatch(I, Ref);

% Display Images
figure;
subplot(1,3,1);
imshow(I);
title('Original Image');

subplot(1,3,2);
imshow(Ref);
title('Reference Image');
```

```
subplot(1,3,3);  
imshow(I_matched);  
title('Matched Output Image');
```

```
% Display Histograms
```

```
figure;  
subplot(1,3,1);  
imhist(I);  
title('Histogram of Original Image');
```

```
subplot(1,3,2);  
imhist(Ref);  
title('Histogram of Reference Image');
```

```
subplot(1,3,3);  
imhist(I_matched);  
title('Histogram of Matched Output Image');
```

```
% ID: 2020-1-60-072
```



**Original Image**



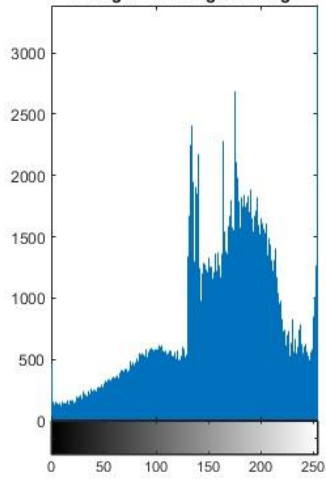
**Reference Image**



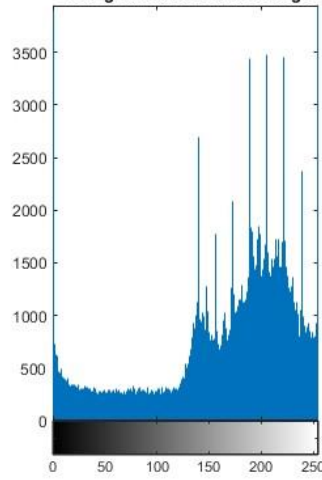
**Matched Output Image**



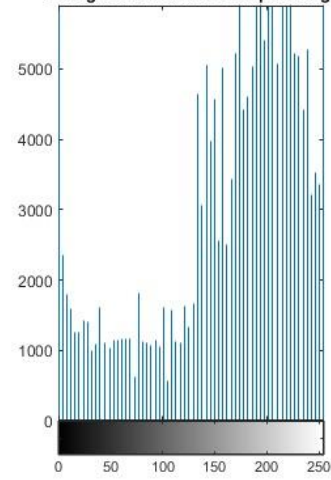
**Histogram of Original Image**



**Histogram of Reference Image**



**Histogram of Matched Output Image**



**Problem 05:**

```
clear all;
close all;
clc;

% Read the image
I = imread('Picture5.jpg'); % Input image

% Convert to grayscale if it is RGB
if size(I, 3) == 3
    I = rgb2gray(I);
end

% Apply Histogram Equalization
I_eq = histeq(I);

% Display the images
figure;
subplot(1,2,1);
imshow(I);
title('Original Image');

subplot(1,2,2);
imshow(I_eq);
title('Histogram Equalized Image');

% Display Histograms
figure;
subplot(1,2,1);
```

```
imhist(I);  
title('Histogram of Original Image');  
  
subplot(1,2,2);  
imhist(I_eq);  
title('Histogram of Equalized Image');  
  
% ID: 2020-1-60-072
```



