

**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:**

4th March, 2025

**Problem 01:**

clear all;

close all;

% Read the image

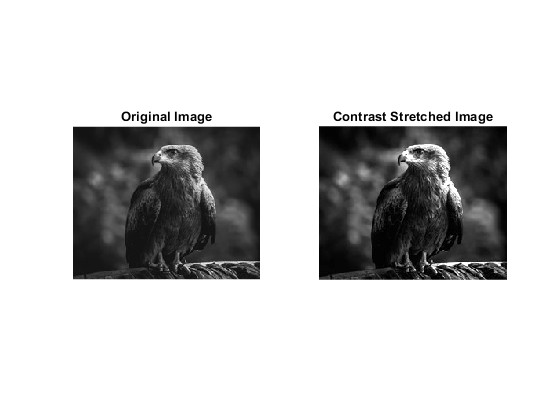
I = imread('Picture1.jpg');

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



**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

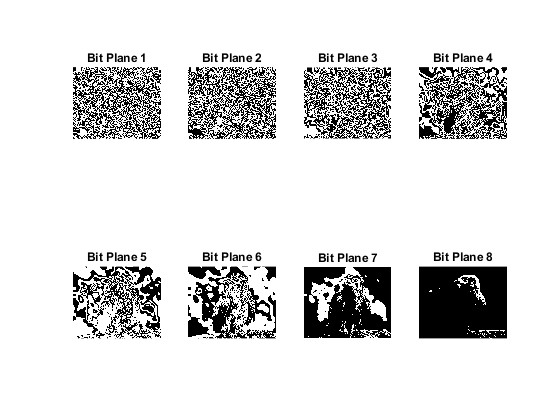
subplot(2,4,i);

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);

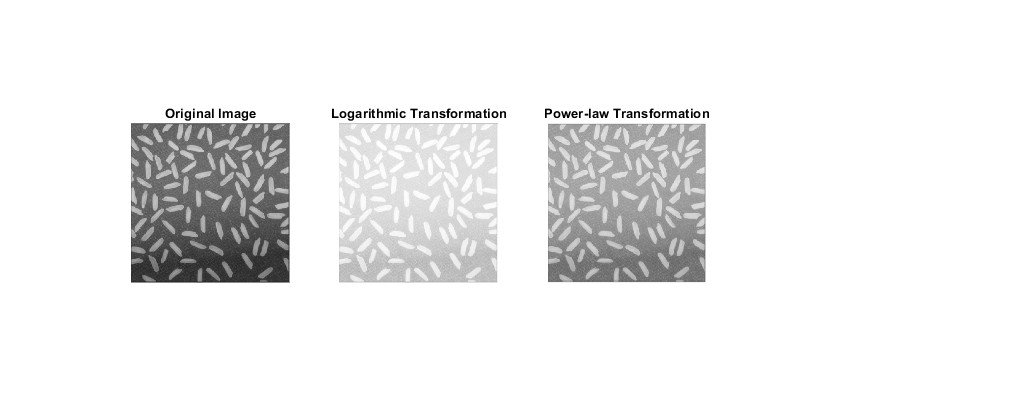
title('Logarithmic Transformation');

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);

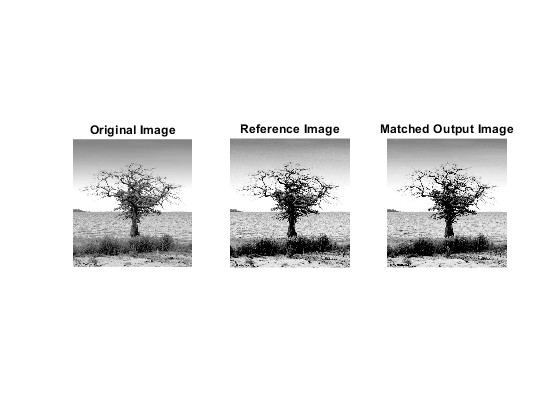
title('Histogram of Reference Image');

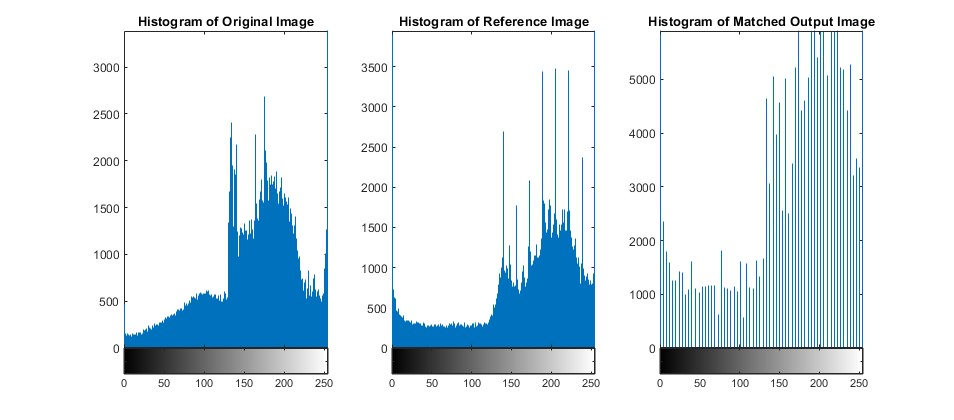
subplot(1,3,3);

imhist(I\_matched);

title('Histogram of Matched Output Image');

% ID: 2020-1-60-072

****

****

**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

A close-up of a chest x-ray

AI-generated content may be incorrect.

**A comparison of a diagram

AI-generated content may be incorrect.**