**T.Y .BTech (Electronics) – Image Processing**

**Expt No -6: Image Preprocessing**

**Problem Statement** –

* + 1. Write a program to

(a) Implement Gray level slicing (intensity level slicing) in to read any gray level image.

(b) Read an 8 bit image and to see the effect of each bit on the image.

(c) Read an image and to extract 8 different planes i.e. ‘bit plane slicing.”

* **Objective:** To analyze the effect on image by image preprocessing.
* **Tools Required:** MATLAB
* **Theory:** *Brief describe in short*
* **Specify the syntax used in MATLAB for following above operation**.

.

* **Conclusion**:
* **Discussion** –

1. ***Program 1:*** same problem statements

% Gray Level Slicing, Bit Plane Slicing, and Viewing Bit Effect

clc;

% Task (a): Gray Level Slicing

% Read a grayscale image

img = imread('Image.png'); % replace with your image path

if size(img, 3) == 3

img = rgb2gray(img); % Convert to grayscale if the image is RGB

end

% Define gray level range for slicing

lower\_limit = 100; % Lower intensity limit (adjust as needed)

upper\_limit = 150; % Upper intensity limit (adjust as needed)

% Gray Level Slicing - Create a binary mask for the selected range

sliced\_img = img;

sliced\_img(img < lower\_limit | img > upper\_limit) = 0; % Set other pixels to 0

sliced\_img(img >= lower\_limit & img <= upper\_limit) = 255; % Set pixels in range to white

% Show the result

figure;

subplot(1, 2, 1);

imshow(img);

title('Original Image');

subplot(1, 2, 2);

imshow(sliced\_img);

title('Gray Level Slicing Image');

% Task (b): Show the effect of each bit on the image (8-bit image)

% For simplicity, use the original image

bit\_effect\_img = img;

% Show the effect of each bit (from most significant to least significant bit)

figure;

for bit = 7:-1:0

% Mask to isolate the current bit

bit\_plane = bitget(bit\_effect\_img, bit+1); % bitget returns the specific bit

subplot(2, 4, 8-bit);

imshow(bit\_plane);

title(['Bit ', num2str(bit)]);

end

% Task (c): Bit Plane Slicing - Extract 8 bit planes

% Extract 8 bit planes from the image

bit\_planes = zeros(size(img, 1), size(img, 2), 8);

% Create a new figure for displaying all 8 bit planes

figure;

for i = 1:8

% Extract i-th bit plane using bitget function

bit\_planes(:,:,i) = bitget(img, i);

% Display each bit plane in a subplot

subplot(2, 4, i);

imshow(bit\_planes(:,:,i));

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

end





