Experiment 5

Shashwat Shah
60004220126
BE comps C22

Aim : Image Negation, grey level slicing and thresholding.

Theory : These image processing techniques help enhance and extract features from image.

→ Image Negation

In image negation, each pixel value is subtracted from the maximum pixel value, resulting in an inverted image.

→ Gray level slicing - Gray level scaling is a technique used to highlight a specific range of intensities in an image.

→ Thresholding

Thresholding is a techniques which converts a grayscale image into a binary image by turning all pixels above a certain intensity value white (1) and all other black (0).

Conclusion : In this experiment we applied image negative grey scale level slicing and thresholding techniques. Image negative inverted pixel values, grey level slicing highlighted specific intensity ranges and thresholding converted images to binary for object detection. These techniques are essential for image manipulation and analysis in field like computer vision and image enhancement.

NAME: Shashwat Shah SAP ID: 60004220126
DIV/BATCH:C22DATE: 14/10/24
23/09/24

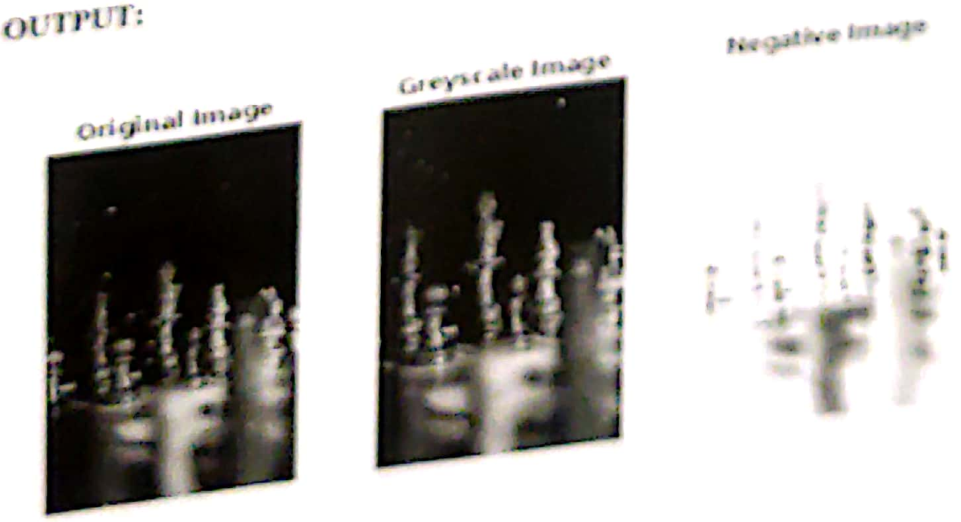
DIGITAL SIGNAL PROCESSING (DSP)
EXPERIMENT 06

AIM: To Image Negative, Grey Level Slicing & Thresholding

CODE:
IMAGE NEGATIVE

```
image_path = 'image1.jpg'; image =
imread(image_path);

% Convert to grayscale if
size(image, 3) == 3
        grey_image = rgb2gray(image);
end
% Number of intensity levels L = 256;

% Convert to negative
negative_image = L - 1 - grey_image; output_path =
'negative_image.jpg'; imwrite(negative_image,
output_path);

figure; subplot(1, 3, 1);
imshow(image);
title('Original Image');

subplot(1, 3, 2); imshow(grey_image);
title('Greyscale Image');

subplot(1, 3, 3); imshow(negative_image);
title('Negative Image');
```

OUTPUT:



Original Image

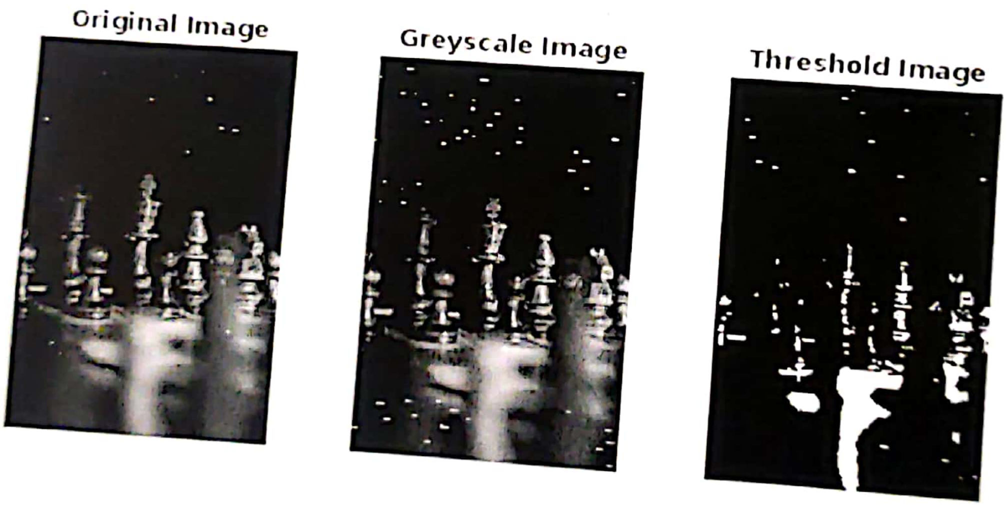
Greyscale Image

Negative Image

THRESHOLDING

```
image_path = 'image1.jpg'; image =
imread(image_path);

% Convert to grayscale if
size(image, 3) == 3
        grey_image = rgb2gray(image);
end

% Number of intensity levels L = 256;
thres = 127;

% Apply thresholding
threshold_image = grey_image >= thres; threshold_image =
uint8(threshold_image) * (L - 1);

output_path2 = 'threshold_image.jpg';
imwrite(threshold_image, output_path2);

figure; subplot(1, 3, 1);
imshow(image);
title('Original Image');

subplot(1, 3, 2); imshow(grey_image);
title('Greyscale Image');

subplot(1, 3, 3); imshow(threshold_image);
title('Threshold Image');
```

OUTPUT:

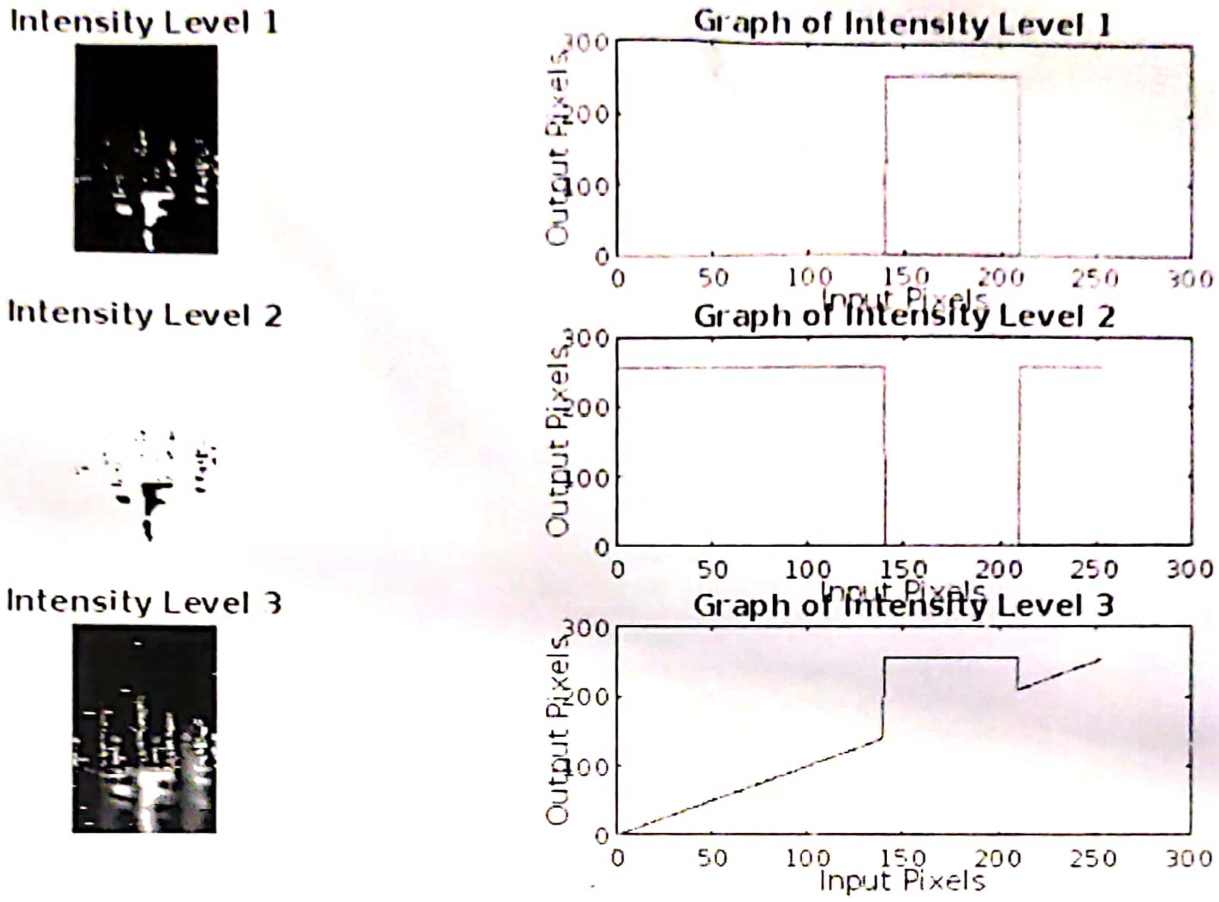


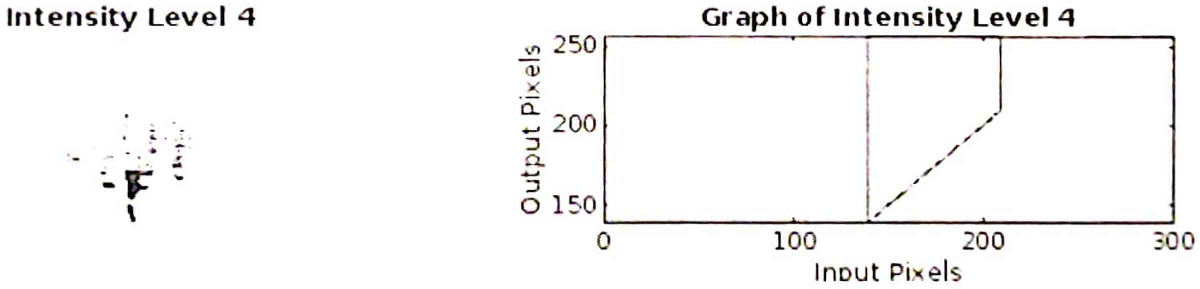
GREY LEVEL SLICING

```
function plot_transformations_in_batches(input_img, x_values, transformations, batch_size)
        num_transforms = size(transformations, 1);
        num_batches = ceil(num_transforms / batch_size);  % Calculate how many batches are needed

        % Convert input image to double for consistent arithmetic input_img =
        double(input_img);

        for batch = 1:num_batches
                figure('Position', [100, 100, 1400, 800]);  % Adjust size for space start_idx = (batch - 1) *
                batch_size + 1;
                end_idx = min(batch * batch_size, num_transforms);

                for i = start_idx:end_idx
                        transform_func = transformations{i, 1}; transform_title =
                        transformations{i, 2};

                        % Apply the transformation to the image transformed_img =
                        arrayfun(transform_func, input_img); y_values =
                        arrayfun(transform_func, x_values);

                        % Plot the transformed image subplot(batch_size, 2, 2*(i -
                        start_idx) + 1); imshow(transformed_img, []);
                        title(transform_title);

                        % Plot the transformation graph subplot(batch_size, 2, 2*(i -
                        start_idx) + 2); plot(x_values, y_values);
                        xlabel('Input Pixels'); ylabel('Output
                        Pixels'); grid on;
                        title(['Graph of ' transform_title]);

                end
        end

end

% Load and process image img =
imread('image1.jpg'); gray_image =
rgb2gray(img);

% Define transformations transformations = {
        @(r) (r > 140 & r < 210) * 255, 'Intensity Level 1';
        @(r) ~(r > 140 & r < 210) * 255, 'Intensity Level 2';
        @(r) (r > 140 & r < 210) * 255 + (r <= 140 | r >= 210) .* double(r), 'Intensity Level 3';
        @(r) (r > 140 & r < 210) .* double(r) + (r <= 140 | r >= 210) * 255, 'Intensity Level 4';
        @(r) (r < 150) * 0 + (r >= 150) * 255, 'Threshold at 150';
        @(r) (r < 60) * 0 + (r >= 60) * 255, 'Threshold at 60';
};

x_values = linspace(0, 255, 500);  % For plotting transformation curves

% Plot transformations in batches of 3 to avoid overloading batch_size = 3;
```

```
plot_transformations_in_batches(gray_image, x_values, transformations, batch_size).
```

OUTPUT:


Intensity Level 1
Graph of Intensity Level 1
Output Pixels
Input Pixels
Intensity Level 2
Graph of Intensity Level 2
Output Pixels
Input Pixels
Intensity Level 3
Graph of Intensity Level 3
Output Pixels
Input Pixels


Intensity Level 4
Graph of Intensity Level 4
Output Pixels
Input Pixels