EAST WEST UNIVERSITY

Department of Computer Science and Engineering

Lab-report:01

Course Name: Digital Image Processing

Course Code: CSE438

Section No: 03

Submitted To:

Prof. Dr. Ahmed Wasif Reza

Department of Computer Science and Engineering

East West University

Submitted by:

Student's ID: 2020-3-60-012

Student's Name: Sadia Islam Prova

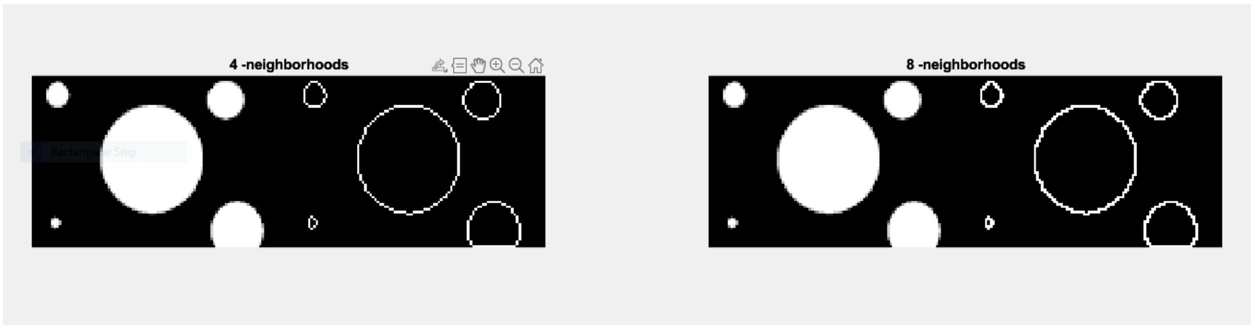
Date of submission: 13-2-24

Problem1: Determine the perimeter of an object by using 4 connected neighborhoods and 8 connected neighborhoods.

Code:

```
img = imread("img1.png");
imshow (img);

n4=bwperim(img,4);
n8=bwperim(img,8);

subplot(1,2,1) , imshowpair(img,n4,'montage');title("4 -neighborhoods");
subplot(1,2,2) , imshowpair(img,n8,'montage');title("8 -neighborhoods");
```

Output:

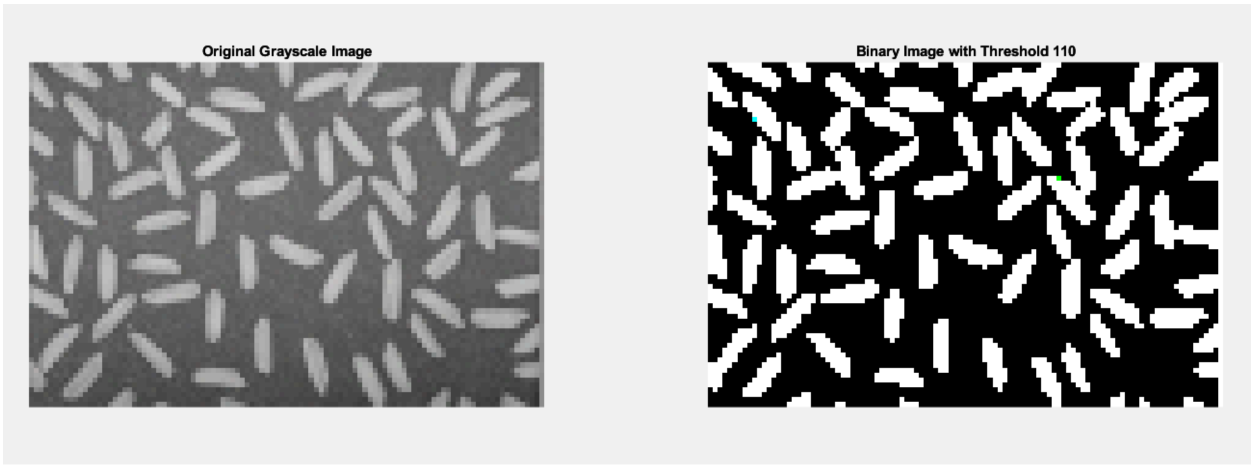


Problem2: Create a binary image using a threshold.

Code:

```
gray_img = imread('img2.png');
threshold_value = 130;
binary_img = gray_img > threshold_value;
binary_img_8bit = uint8(binary_img) * 255;
figure;
subplot(1, 2, 1);
imshow(gray_img);
title('Original Grayscale Image');
subplot(1, 2, 2);
imshow(binary_img_8bit);
title('Binary Image with Threshold 110');
```

Output:

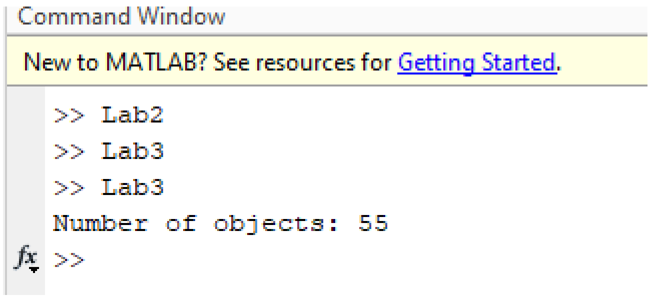


Problem3: Determine the number of objects in the binary image generated in Question 2 using the concept of connectivity.

Code:

```
img = imread("img2.png");
imshow (img);

if size(img,3) > 1
    img_gray= rgb2gray(img);

else
    img_gray=img;

end
threshold_value=150;
binary_img=img_gray>threshold;
cc=bwconncomp(binary_img);
num_objects=cc.NumObjects;
fprintf('Number of objects: %d\n',num_objects);
```

Output:



Command Window

New to MATLAB? See resources for Getting Started.

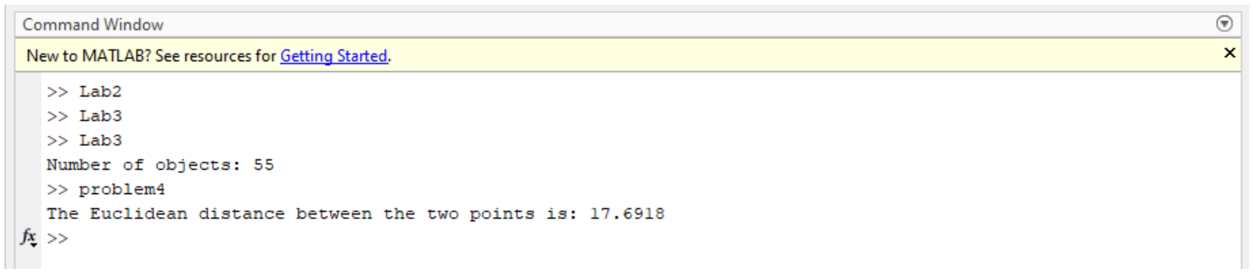
```
>> Lab2
>> Lab3
>> Lab3
Number of objects: 55
>>
```

Problem4: Find the Euclidean distance between two points of the image.

Code:

```
img = imread("img2.png");
imshow (img);


point1 = [2 5];
point2 = [15 17];

distance = sqrt(sum((point1 - point2).^2));

disp(['The Euclidean distance between the two points is: ', num2str(distance)]);
```

Output:



Command Window

New to MATLAB? See resources for Getting Started.

```
>> Lab2
>> Lab3
>> Lab3
Number of objects: 55
>> problem4
The Euclidean distance between the two points is: 17.6918
>>
```

Problem5:

Apply the following operations using Fig.1 and Fig.2:

a. Addition

b. Subtraction

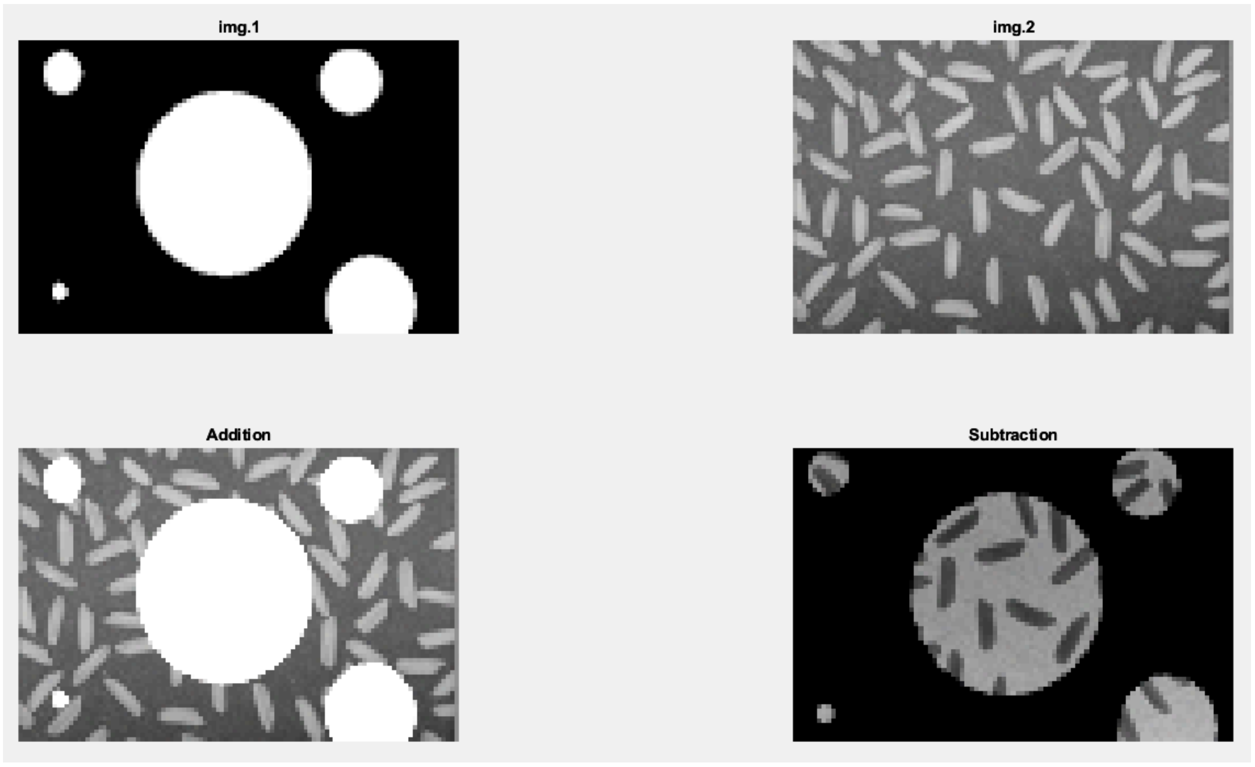
c. Multiplication

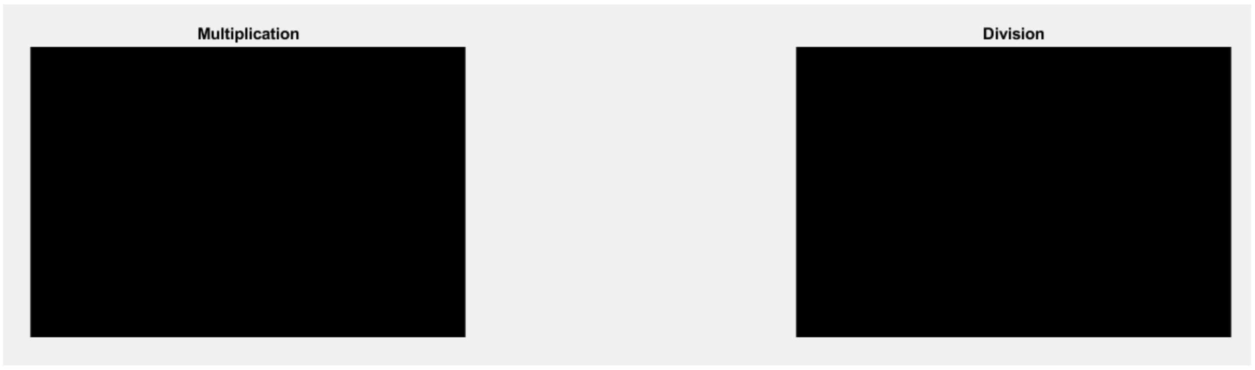
d. Division

Code:

```
img1 = imread("img1.png");
img2 = imread("img2.png");

if ~isequal(size(img1), size(img2))
    error("Images must have the same size for operations.");
end
addition = img1 + img2;

subtraction = img1 - img2;

multiplication = img1 .* img2;
multiplication = uint8(round(multiplication * 255 / max(multiplication(:))));

division = img1 ./ img2;
division(division == Inf) = 128;

figure;

subplot(2,2,1); imshow(img1); title("img.1");
subplot(2,2,2); imshow(img2); title("img.2");

subplot(2,2,3); imshow(addition); title("Addition");
subplot(2,2,4); imshow(subtraction); title("Subtraction");

figure;

subplot(2,2,1); imshow(multiplication); title("Multiplication");
subplot(2,2,2); imshow(division); title("Division");
```

Output:





Problem6:

Apply the following operations using Fig.1 and Fig.2:

a. AND

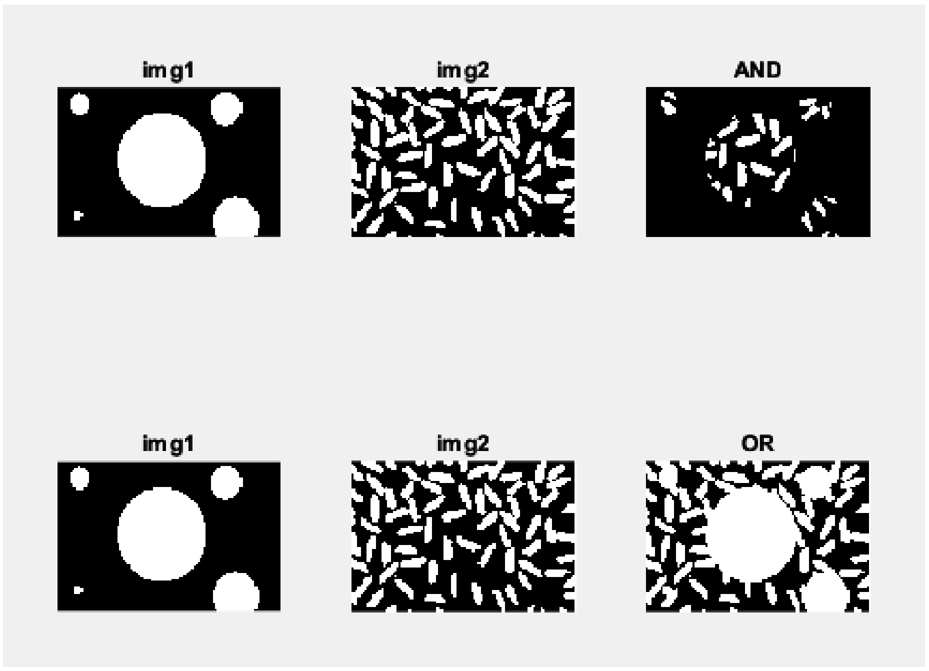
b. OR

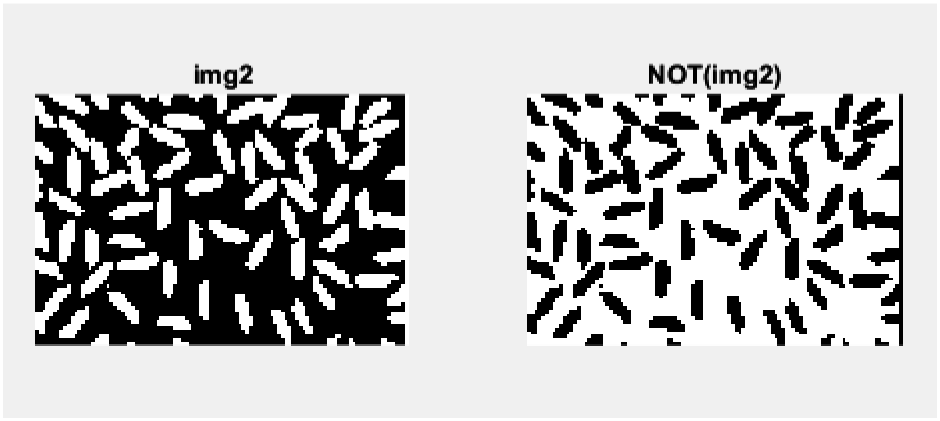
c. NOT

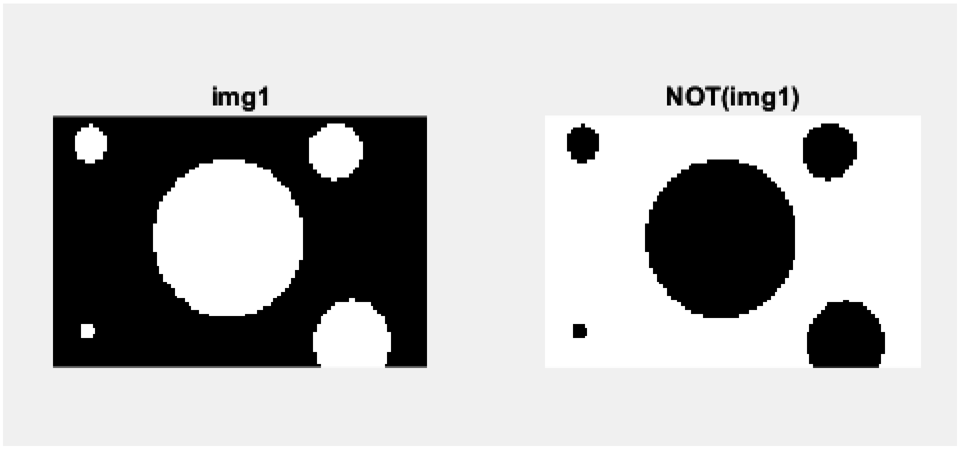
Code:

```
img1 = imread("img1.png");
img2 = imread("img2.png");

if ~isequal(size(img1), size(img2))
    error("Images must have the same size for logical operations.");
end

if ~islogical(img1) && ~islogical(img2)
    img1 = im2bw(img1, 0.5);
    img2 = im2bw(img2, 0.5);
end


and_result = img1 & img2;

or_result = img1 | img2;

not_img1 = ~img1;
not_img2 = ~img2;

figure;

subplot(2,3,1); imshow(img1); title("img1");
subplot(2,3,2); imshow(img2); title("img2");
subplot(2,3,3); imshow(and_result); title("AND");

subplot(2,3,4); imshow(img1); title("img1");
subplot(2,3,5); imshow(img2); title("img2");
subplot(2,3,6); imshow(or_result); title("OR");

figure;

subplot(1,2,1); imshow(img1); title("img1");
subplot(1,2,2); imshow(not_img1); title("NOT(img1)");

figure;

subplot(1,2,1); imshow(img2); title("img2");
subplot(1,2,2); imshow(not_img2); title("NOT(img2)");
```

Output:





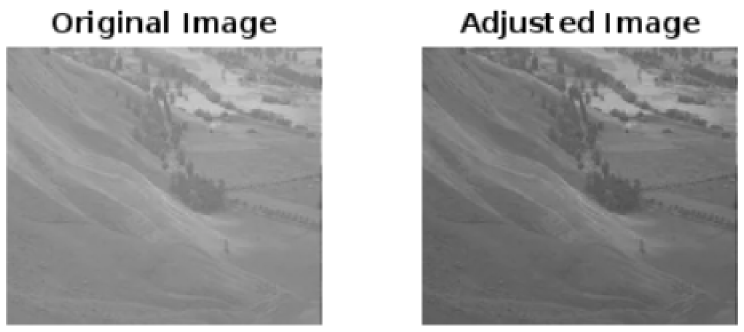


Problem7: Adjust the contrast of the following image.

Code:

```
img = imread('img7.png');

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

low_in = min(img(:));
high_in = max(img(:));
gamma = 1.5;
adjusted_img = imadjust(img, [low_in/high_in, high_in/high_in], [0, 1], gamma);

subplot(1, 2, 2);
imshow(adjusted_img);
title('Adjusted Image');
```

Output:

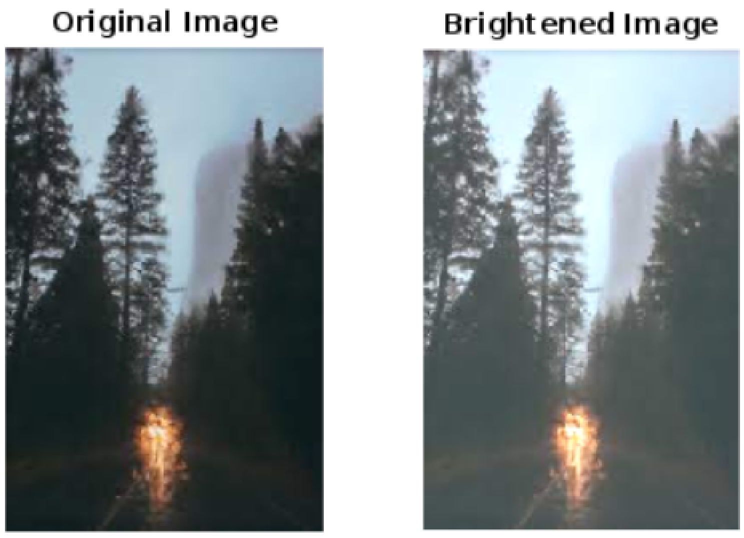


Problem8: Brighten the following image

Code:

```
img = imread('img8.png');

brightness_factor = 50;
brightened_img = img + brightness_factor;

brightened_img = min(brightened_img, 255);

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

subplot(1, 2, 2);
imshow(brightened_img);
title('Brightened Image');
```

Output:

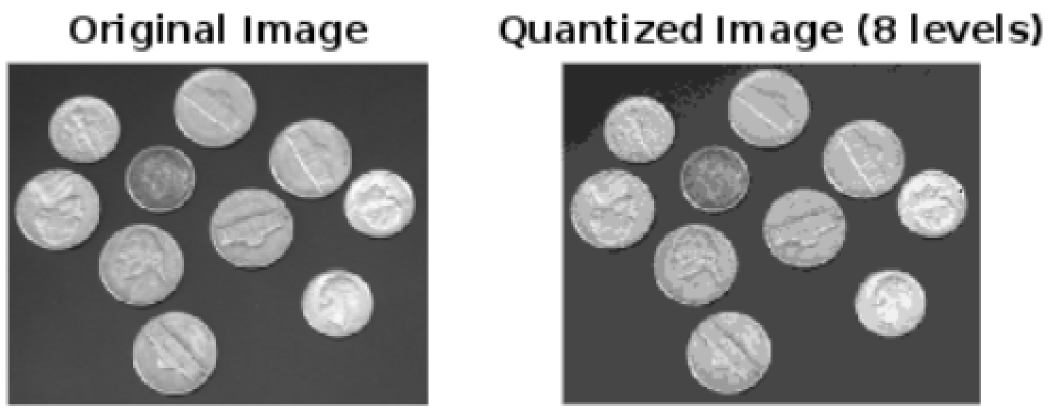


Problem9: Quantize the Grayscale image by 8 levels.

Code:

```
img = imread('img9.png');

min_intensity = double(min(img(:)));
max_intensity = double(max(img(:)));

thresholds = linspace(min_intensity, max_intensity, 8+1);

quantized_img = zeros(size(img));
for i = 1:numel(thresholds)-1
    mask = img >= thresholds(i) & img < thresholds(i+1);
    quantized_img(mask) = round(mean([thresholds(i), thresholds(i+1)]));
end

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

subplot(1, 2, 2);
imshow(uint8(quantized_img));
title('Quantized Image (8 levels)');
```

Output:

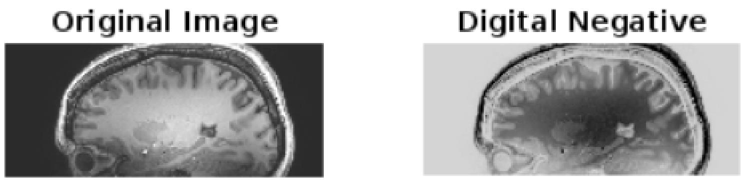


Problem10: Find the digital negative of the image.

Code:

```
img = imread('img10.png');

img_double = double(img);

max_intensity = max(img_double(:));

negative_img = max_intensity - img_double;

negative_img_uint8 = uint8(negative_img);

subplot(1, 2, 1);
imshow(img, []);
title('Original Image');

subplot(1, 2, 2);
imshow(negative_img_uint8, []);
title('Digital Negative');
```

Output:


Original Image
Digital Negative