Lab Report 1

Image Processing Operations

Digital Image Processing

CSE438

Section: 03

Semester: Spring 2025

Submitted To:

Md. Asif Khan Rifat

Lecturer

Department of Computer Science

and Engineering

Submitted By:

Suddip Paul Arnab

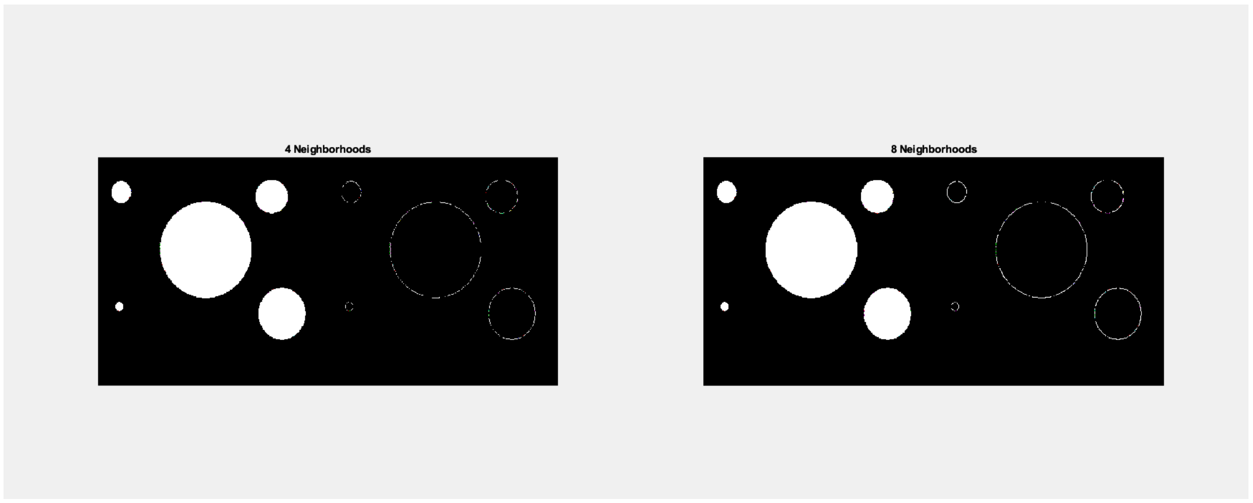
2022-1-60-356

Date of submission: 3 March 2025

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

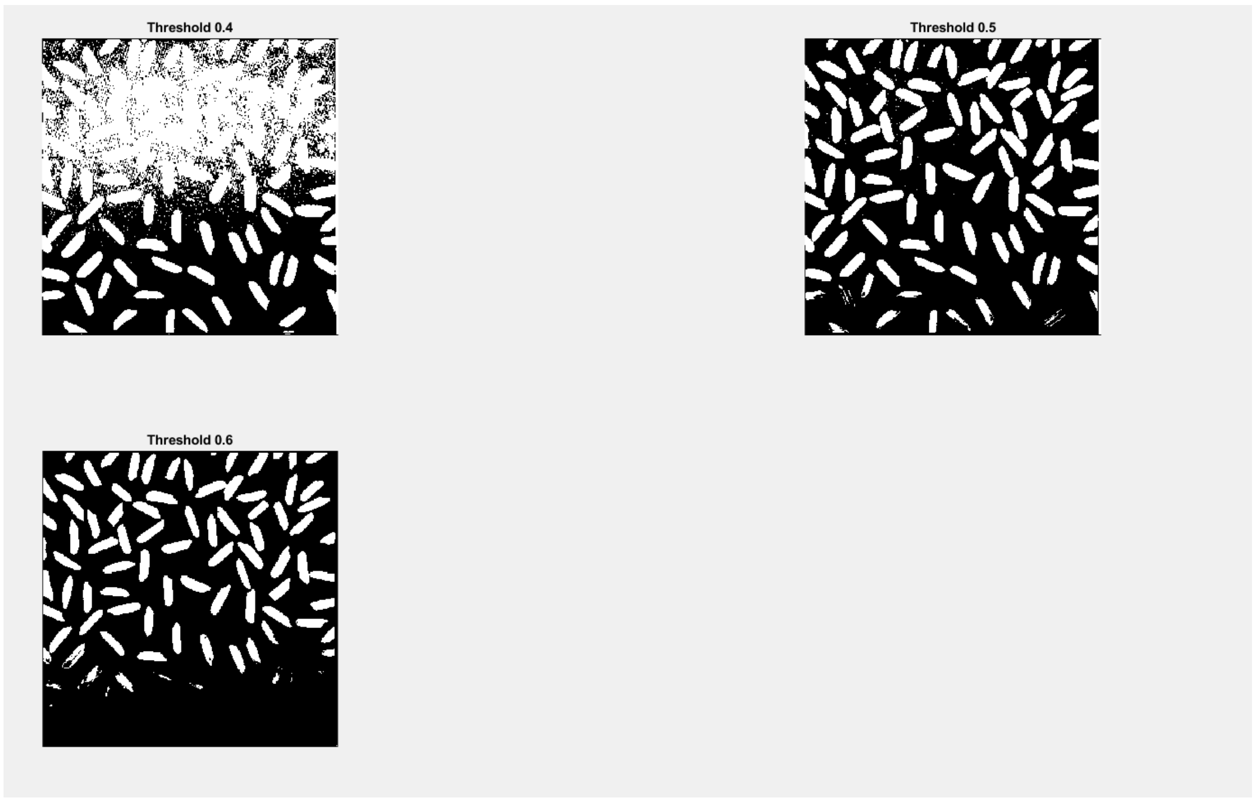
   Code:

```
img = imread("Picture1.png");
imshow(img);
bin_img = imbinarize(img);
n4 = bwperim(bin_img, 4);
n8 = bwperim(bin_img, 8);
subplot(1,2,1),imshowpair(bin_img,n4,'montage');
title("4 Neighborhoods");
subplot(1,2,2),imshowpair(bin_img,n8,'montage');
title("8 Neighborhoods");
```



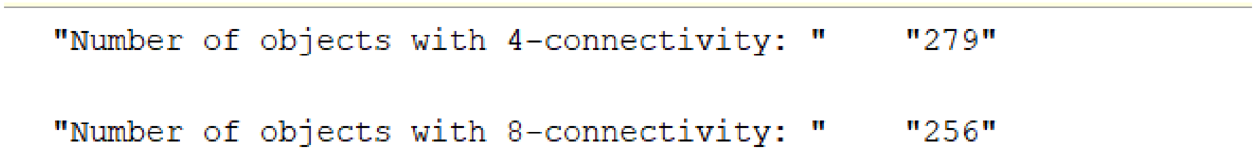
2. Create a binary image using a threshold.
   Code:

```
img = imread("Picture2.png");
gray_img = rgb2gray(img);
imshow(gray_img);
binary_img1 = imbinarize(gray_img, 0.4);
binary_img2 = imbinarize(gray_img, 0.5);
binary_img3 = imbinarize(gray_img, 0.6);
subplot(2,2,1),imshow(binary_img1);
title("Threshold 0.4");
subplot(2,2,2),imshow(binary_img2);
title("Threshold 0.5");
subplot(2,2,3),imshow(binary_img3);
title("Threshold 0.6");
```



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

```
img = imread("Picture2.png");
gray_img = rgb2gray(img);
binary_img2 = imbinarize(gray_img, 0.5);
n4 = bwconncomp(binary_img2, 4);
n8 = bwconncomp(binary_img2, 8);
num_objects_4 = n4.NumObjects;
num_objects_8 = n8.NumObjects;
disp(["Number of objects with 4-connectivity: ", num2str(num_objects_4)]);
disp(["Number of objects with 8-connectivity: ", num2str(num_objects_8)]);
```

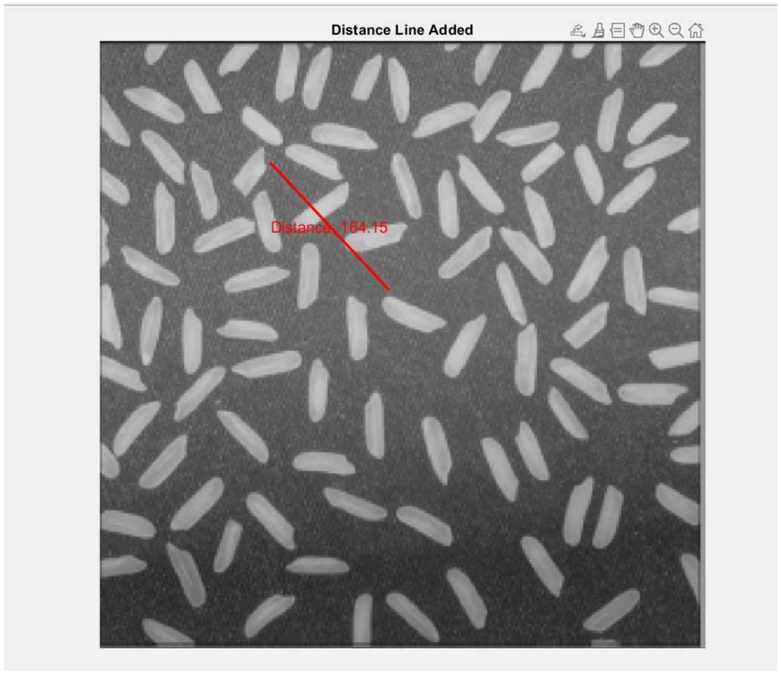


```
"Number of objects with 4-connectivity: "    "279"

"Number of objects with 8-connectivity: "    "256"
```

4. Find the Euclidean distance between two points of the image.

Code:

```
i3 = imread("Picture2.png");

imshow(i3);
title('Select two points on the image');

[x, y] = ginput(2);

distance = norm([x(2) - x(1), y(2) - y(1)]);

fprintf('The Euclidean distance between the points is: %.2f\n', distance);

hold on;
plot(x, y, 'r-', 'LineWidth', 2);
text(mean(x), mean(y), sprintf('Distance: %.2f', distance), 'Color', 'red',
'FontSize', 12, 'HorizontalAlignment', 'center');
hold off;

title('Distance Line Added');
```



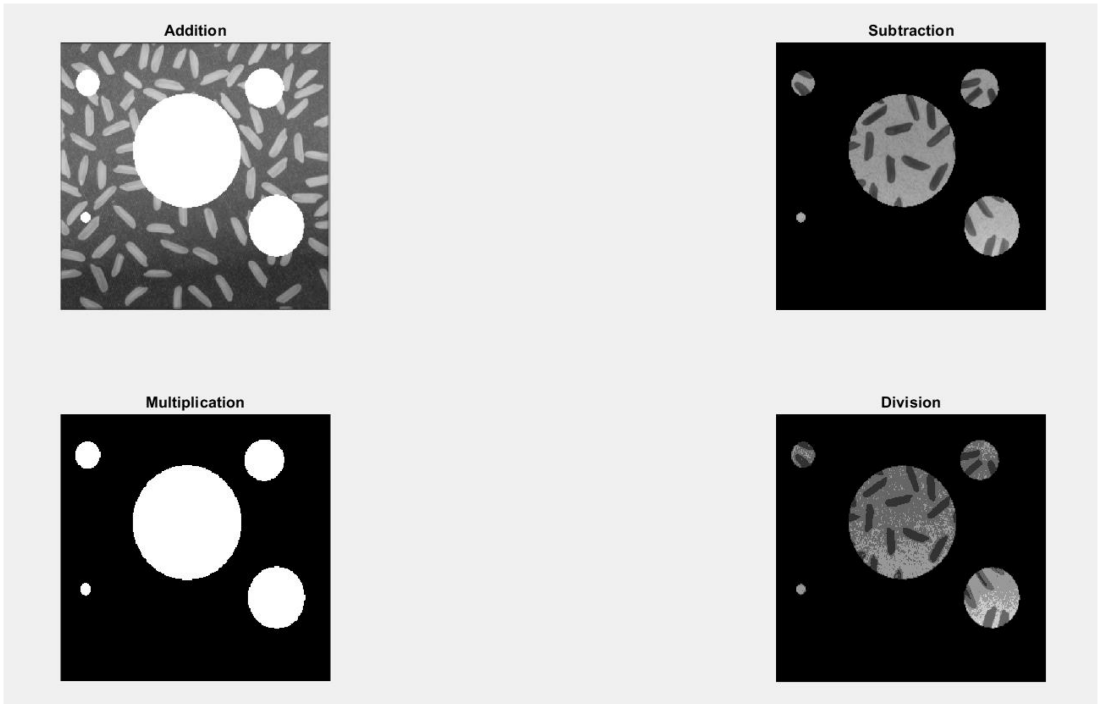
```
The Euclidean distance between the points is: 164.15
>>
```

5. Apply the following operations using Fig.1 and Fig.2:
   a. Addition
   b. Subtraction
   c. Multiplication

d. Division

Code:

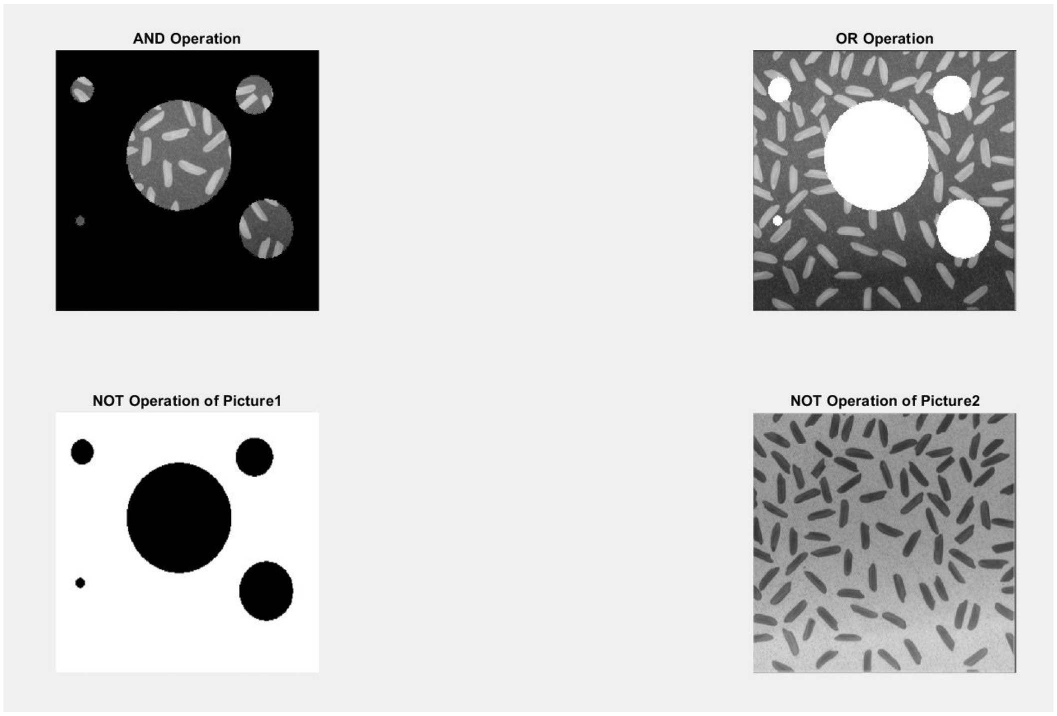
```
img1 = imread("Picture1.png");
img2 = imread("Picture2.png");

img2 = imresize(img2, size(img1(:,:,1)));

gray_img1 = rgb2gray(img1);
gray_img2 = rgb2gray(img2);

img_add = imadd(gray_img1, gray_img2);
img_subtract = imsubtract(gray_img1, gray_img2);
img_multiply = immultiply(gray_img1, gray_img2);
img_divide = imdivide(gray_img1, gray_img2);

figure;
subplot(2,2,1), imshow(img_add), title('Addition');
subplot(2,2,2), imshow(img_subtract), title('Subtraction');
subplot(2,2,3), imshow(img_multiply), title('Multiplication');
subplot(2,2,4), imshow(img_divide, []), title('Division');
```



6. Apply the following operations using Fig.1 and Fig.2:
   a. AND
   b. OR
   c. NOT

Code:

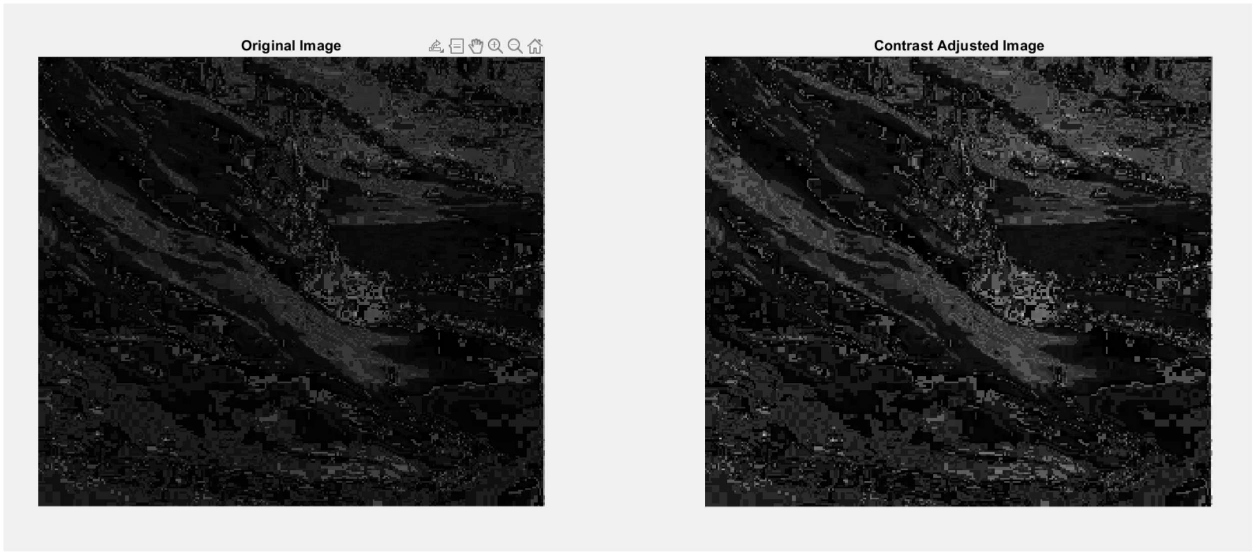
```
img1 = imread("Picture1.png");
img2 = imread("Picture2.png");

img2 = imresize(img2, size(img1(:,:,1)));

gray_img1 = rgb2gray(img1);
gray_img2 = rgb2gray(img2);

img_and = bitand(gray_img1, gray_img2);
img_or = bitor(gray_img1, gray_img2);
img_not1 = bitcmp(gray_img1);
img_not2 = bitcmp(gray_img2);

figure;
subplot(2,2,1), imshow(img_and), title('AND Operation');
subplot(2,2,2), imshow(img_or), title('OR Operation');
subplot(2,2,3), imshow(img_not1), title('NOT Operation of Picture1');
subplot(2,2,4), imshow(img_not2), title('NOT Operation of Picture2');
```



7. Adjust the contrast of the following image.

Code:

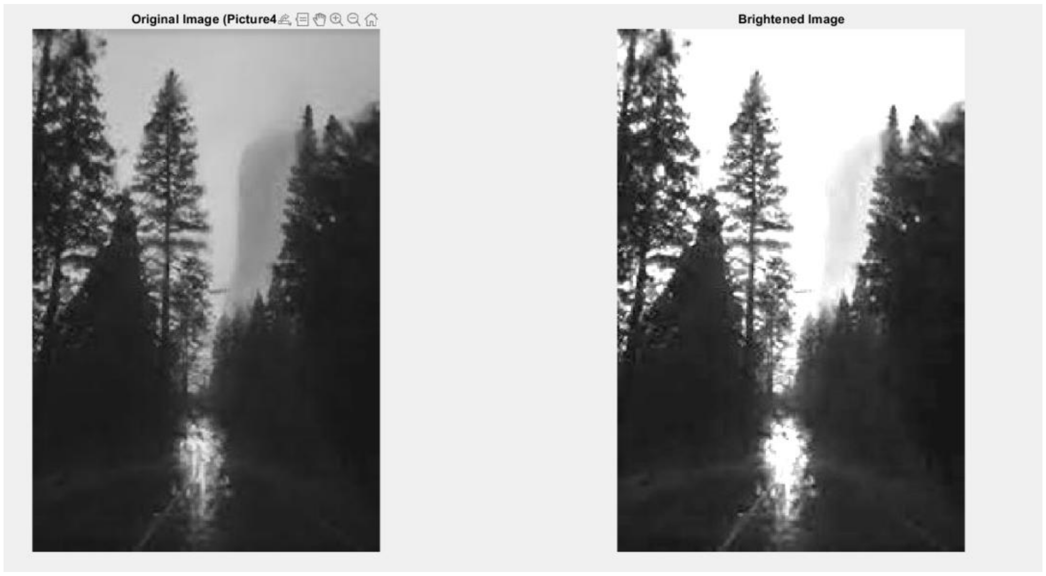
```
img3 = imread("Picture3.png");
gray_img3 = im2gray(img3);

num_of_levels = 128;
contrast_img = floor(gray_img3 / (256 / num_of_levels)) * (256 /
num_of_levels);

figure;
subplot(1,2,1), imshow(gray_img3), title('Original Image');
subplot(1,2,2), imshow(contrast_img, []), title('Contrast Adjusted Image');
```



8. Brighten the following image

Code:

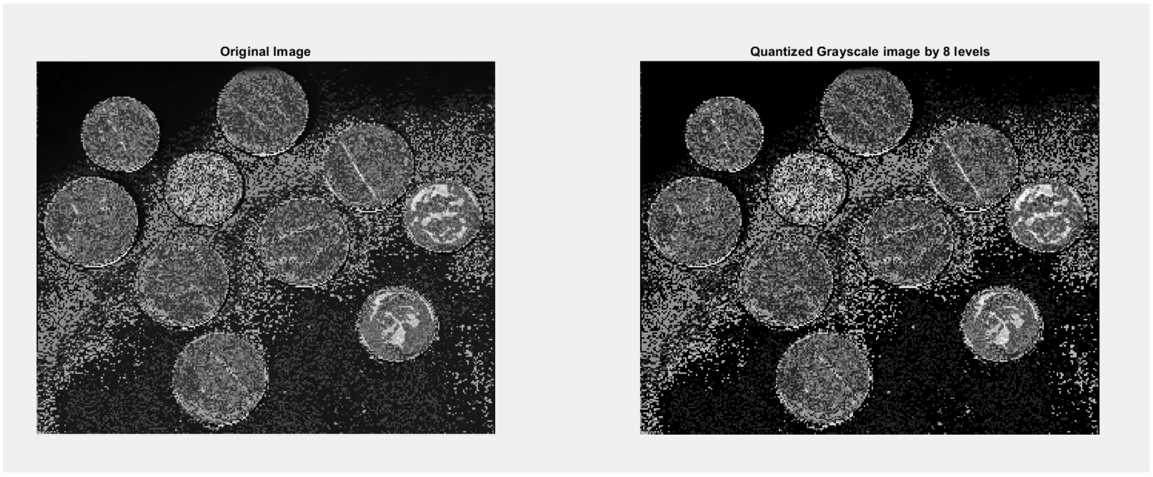
```
img4 = imread("Picture4.jpg");
gray_img4 = im2gray(img4);
brightness_increase = 100;
brightened_img = min(gray_img4 + brightness_increase, 255);
figure;
subplot(1,2,1), imshow(gray_img4), title('Original Image (Picture4)');
subplot(1,2,2), imshow(brightened_img, []), title('Brightened Image');
```



9. Quantize the Grayscale image by 8 levels.

Code:

```
img = imread("Picture5.png");
gray_img = im2gray(img);
quantized_img = gray_img;
[r, c] = size(quantized_img);
for i = 1:r
    for j = 1:c
        f = quantized_img(i, j);
        if f <= 32
            quantized_img(i, j) = 16;
        elseif f > 32 && f <= 64
            quantized_img(i, j) = 48;
        elseif f > 64 && f <= 96
            quantized_img(i, j) = 80;
        elseif f > 96 && f <= 128
            quantized_img(i, j) = 112;
        elseif f > 128 && f <= 160
            quantized_img(i, j) = 144;
        elseif f > 160 && f <= 192
            quantized_img(i, j) = 176;
        elseif f > 192 && f <= 224
            quantized_img(i, j) = 208;
        else
            quantized_img(i, j) = 240;
        end
    end
end
figure;
subplot(1,2,1), imshow(gray_img), title('Original Image');
subplot(1,2,2), imshow(quantized_img, []), title('Quantized Grayscale image by
8 levels');
```



10. Find the digital negative of the image.

Code:

```
img = imread("Picture6.png");
gray_img = im2gray(img);
negative_img = 255 - gray_img;
figure;
subplot(1,2,1), imshow(gray_img), title('Original Grayscale Image');
subplot(1,2,2), imshow(negative_img), title('Digital Negative Image');
```