A logo for a university

AI-generated content may be incorrect.

**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);

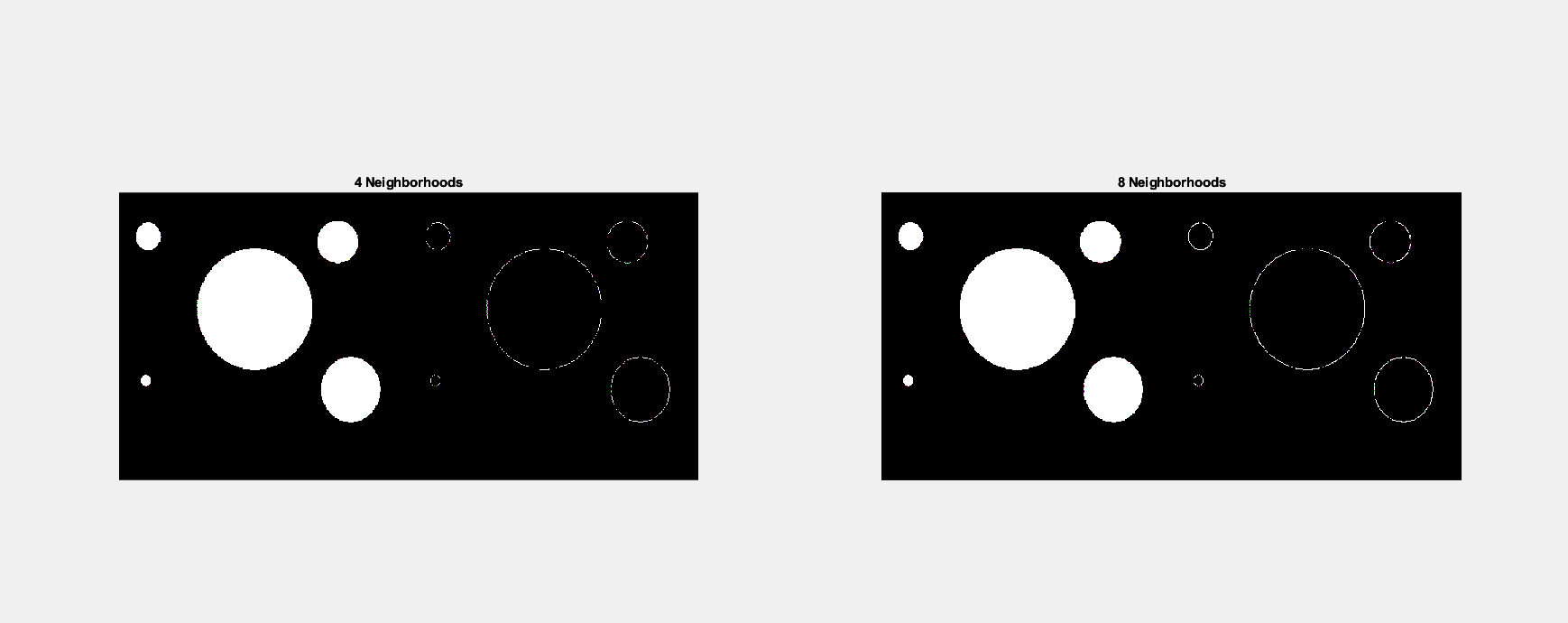
n8 = bwperim(bin\_img, 8);

subplot(1,2,1),imshowpair(bin\_img,n4,'montage');

title("4 Neighborhoods");

subplot(1,2,2),imshowpair(bin\_img,n8,'montage');

title("8 Neighborhoods");



1. 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");

A screenshot of a computer screen

AI-generated content may be incorrect.

1. 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)]);

A close-up of words

AI-generated content may be incorrect.

1. 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');

A screenshot of a computer

AI-generated content may be incorrect.



1. Apply the following operations using Fig.1 and Fig.2:
2. Addition
3. Subtraction
4. Multiplication
5. 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);

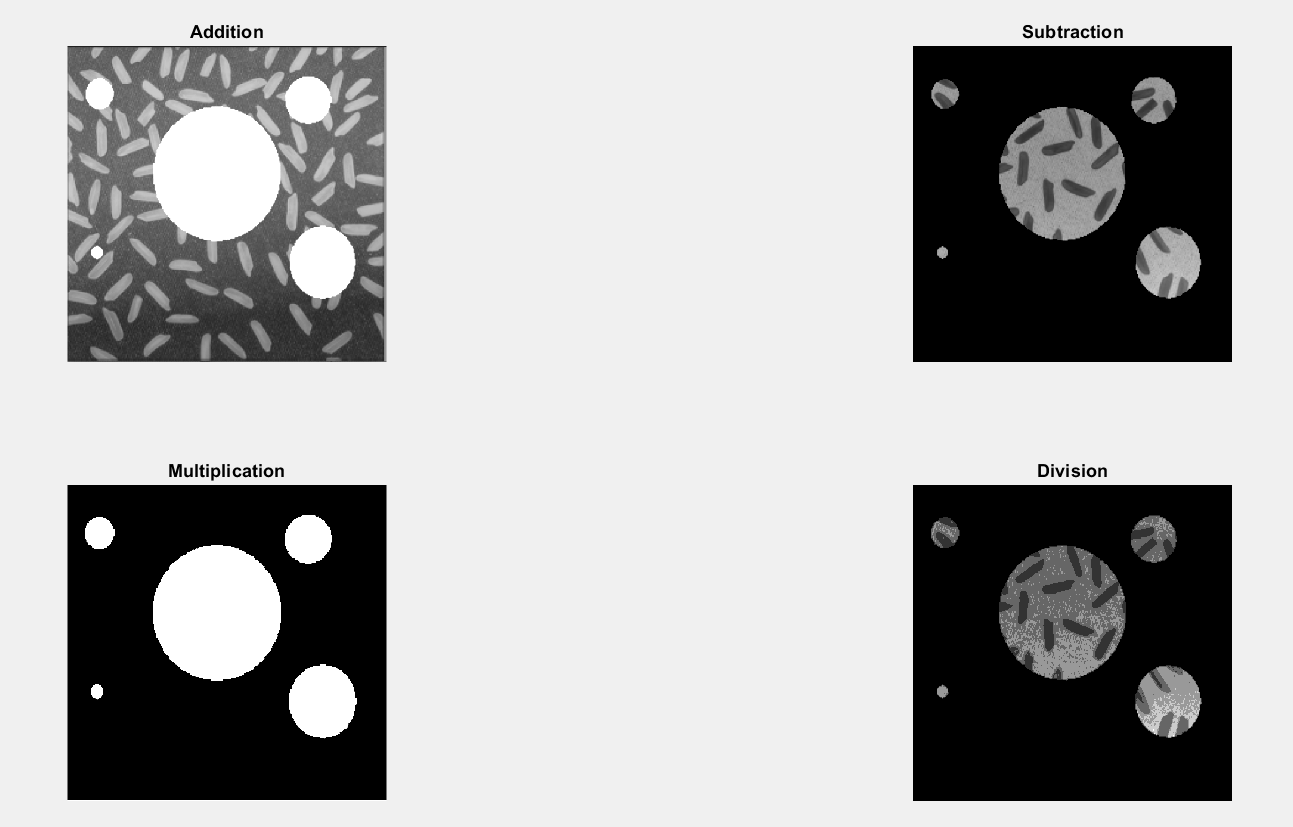
figure;

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');



1. Apply the following operations using Fig.1 and Fig.2:
2. AND
3. OR
4. 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');

A black and white image of different shapes

AI-generated content may be incorrect.

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');

A close-up of a black and white image

AI-generated content may be incorrect.

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');

A close-up of a brain scan

AI-generated content may be incorrect.