**Assignment NO #04**

****

Spring 2023

CSE-423 Digital Image Processing

Submitted by: **Ashfaq Ahmad**

Registration No: **19PWCSE1795**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Doctor. Abeer Irfan**

**June** 14, 2023

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**Task01:**

Take the image of your choice and take out hidden objects in the background using the techniques

discussed above.

**Hint for Activity No.1**

Apply a 3 x 3 moving average filter on the input image and detect areas of higher contrast like edges

and then use the co-efficient in g(x,y) to produce the output image.

**MATLAB Code:**

% Read the input image

inputImage = imread('house.png');

% Convert the image to grayscale if necessary

if size(inputImage, 3) > 1

grayImage = rgb2gray(inputImage);

else

grayImage = inputImage;

end

% Apply a 3x3 moving average filter

smoothed = conv2(double(grayImage), ones(3)/9, 'same');

% Compute the gradient magnitude using the Sobel operator

gradMag = sqrt(imfilter(smoothed, fspecial('sobel').').^2 + imfilter(smoothed, fspecial('sobel')).^2);

% Threshold the gradient magnitude

threshold = 0.2 \* max(gradMag(:)); % Adjust the threshold as needed

edgeMap = gradMag > threshold;

% Remove hidden objects using the edge map

outputImage = inputImage;

outputImage(repmat(edgeMap, [1, 1, size(inputImage, 3)])) = 0;

% Display the output image

figure;

subplot(1, 2, 1);

imshow(inputImage);

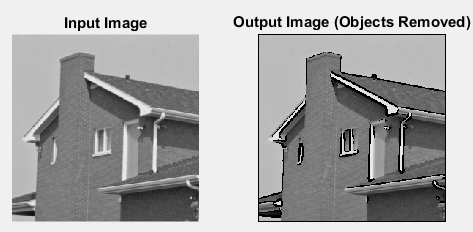
title('Input Image');

subplot(1, 2, 2);

imshow(outputImage);

title('Output Image (Objects Removed)');

**Result:**



**Task02:**

Take the input image and apply the smoothing filter mask of order: i− 3 x 3,

ii− 5 x 5, iii− 9 x 9, and iv− 15 x 15. Sketch the output image and comment on your results.

**MATLAB Code:**

% Load the input image

inputImage = imread('house.png');

% Convert the image to grayscale if necessary

if size(inputImage, 3) > 1

inputImage = rgb2gray(inputImage);

end

% Apply the smoothing filters

order1Image = imfilter(inputImage, ones(3) / 9);

order2Image = imfilter(inputImage, ones(5) / 25);

order3Image = imfilter(inputImage, ones(9) / 81);

order4Image = imfilter(inputImage, ones(15) / 225);

% Display the input and output images

figure;

subplot(2, 3, 1);

imshow(inputImage);

title('Input Image');

subplot(2, 3, 2);

imshow(order1Image);

title('Order 3x3');

subplot(2, 3, 3);

imshow(order2Image);

title('Order 5x5');

subplot(2, 3, 4);

imshow(order3Image);

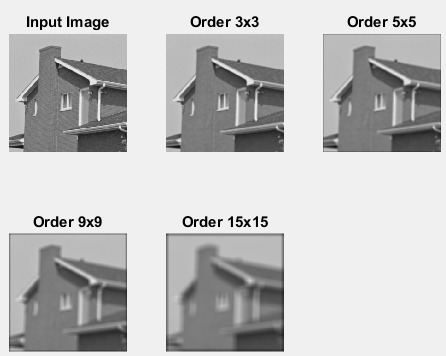
title('Order 9x9');

subplot(2, 3, 5);

imshow(order4Image);

title('Order 15x15');

**Result:**



**Task03:**

Take a noised image and prove which order statistic filter is best.

**Hint for Activity No. 3**

Read an image and add salt and pepper noise with it using imnoise() function. And then write your

code to implement Eq. 4,5, and 6

**MATLAB Code:**

clc

clear all

close all;

% Read the original image

originalImageGray = imread('house.png');

% Add salt and pepper noise to the image

noisyImage = imnoise(originalImageGray, 'salt & pepper', 0.2);

% Display the original and noisy images

figure;

subplot(1, 2, 1);

imshow(originalImageGray);

title('Original Image');

subplot(1, 2, 2);

imshow(noisyImage);

title('Noisy Image');

% Apply Median filter

medianFiltered = medfilt2(noisyImage);

% Apply Min filter

minFiltered = ordfilt2(noisyImage, 1, true(3));

% Apply Max filter

maxFiltered = ordfilt2(noisyImage, 9, true(3));

% Display the denoised images

figure;

subplot(2, 2, 1);

imshow(noisyImage);

title('Noisy Image');

subplot(2, 2, 2);

imshow(medianFiltered);

title('Median Filtered');

subplot(2, 2, 3);

imshow(minFiltered);

title('Min Filtered');

subplot(2, 2, 4);

imshow(maxFiltered);

title('Max Filtered');

**Result:**

