Digital Image Processing EECS 740

Assignment 1

Lohith Nanuvala

2790468

Abstract:

This assignment is meant to get familiar with programming for image processing. Fundamentals like image filtering and enhancement algorithms were emphasized. Power law transformation, Histogram equalization, Gaussian noise and its removal, Laplacian enhancement technique are some of the concepts demonstrated. I have considered two images, one with underexposure and other with overexpose. The transformations discussed above have been applied on these images and results analysed.

**Power law transformation:**

* Power law transformation is one of the intensity transformation techniques.
* The transformation is given by the equation s=crᵞ where r is the intensity of the input image, ᵞ is the power of the transform, c is a constant, s is the intensity of the output image.
* When ᵞ value is larger, the lower intensities are stretched and vice versa.
* Power law transform is mainly used for contrast enhancement.

**Histogram:**

* Histogram of an image gives the distribution of # of times an intensity occurs.
* Histogram can be useful to understand the over and under exposure in the image, the brightness, contrast and dynamic range.
* Histogram only provides statistical information, it does not give any information about the spatial distribution of intensities, and thus we cannot use histogram to reconstruct an image.
* Matlab syntax: imhist(I), I is the input image.

**Histogram Equalization:**

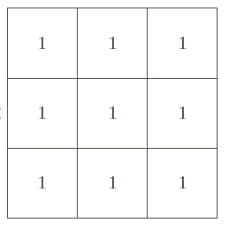
* Histogram equalization is the method of contrast adjustment using the image's histogram.
* We use a monotonic function to transform the input intensities into the output intensities which will be evenly distributed.
* Matlab syntax: histeq(I, N), Where I is the input image, N is the maximum intensity value.

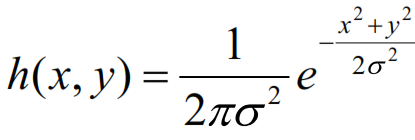
**Gaussian Noise:**

* Gaussian noise isstatistical noise having a probability density function (PDF) equal to that of the normal distribution.
* Principal sources of Gaussian noise in digital images arise during acquisition e.g. sensor noise caused by poor illumination and/or high temperature, and/or transmission e.g. electronic circuit noise.
* Gaussian noise can be reduced using a spatial filter, though when smoothing an image, an undesirable outcome may result in the blurring of fine-scaled image edges.

Filtering using average filter is obtained by convolving filter with the image.

Ex:

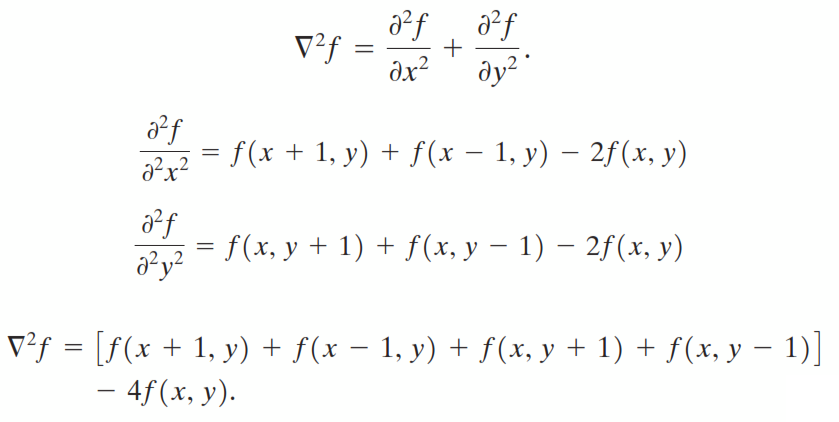
Is a 3x3 averaging filter.

A Gaussian filter is obtained by 

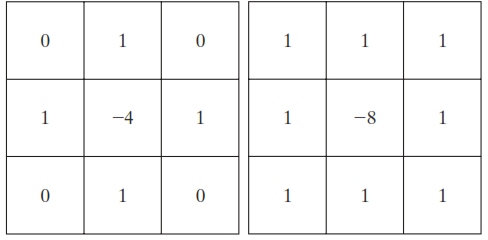
Laplacian Enhancement:

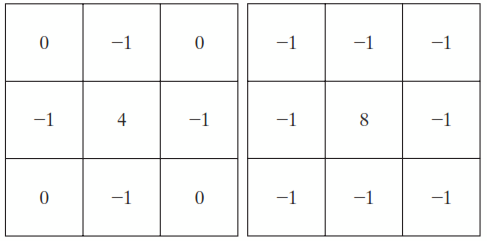
* Laplacian is a derivative operator, which highlights grey-level discontinuities in an image and deemphasizes regions with slowly varying grey levels.
* It is used for image sharpening.

Laplacian derivate operator is given by,



Some laplacian filter masks are,



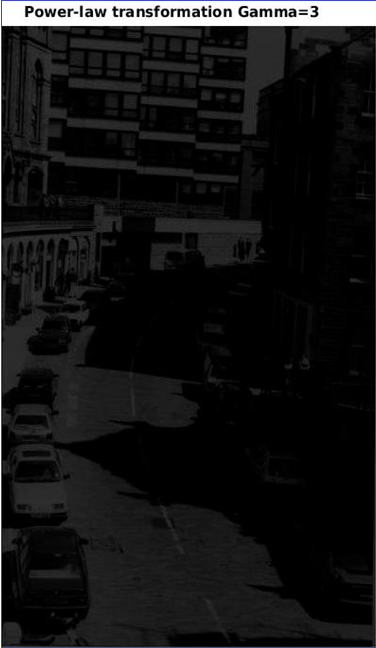


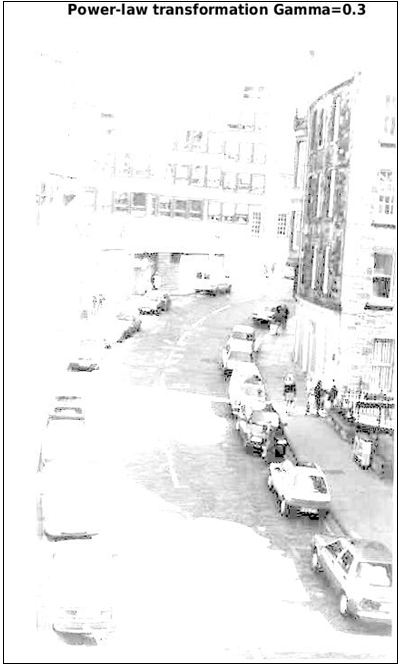
**RESULTS:**

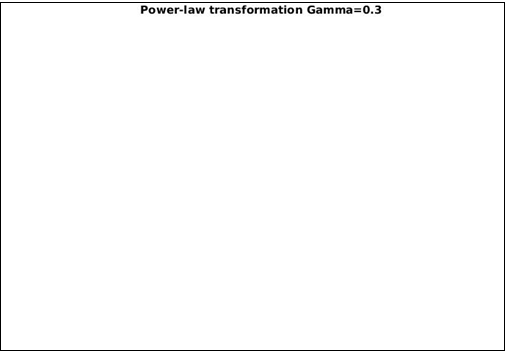
Over exposed image Original:

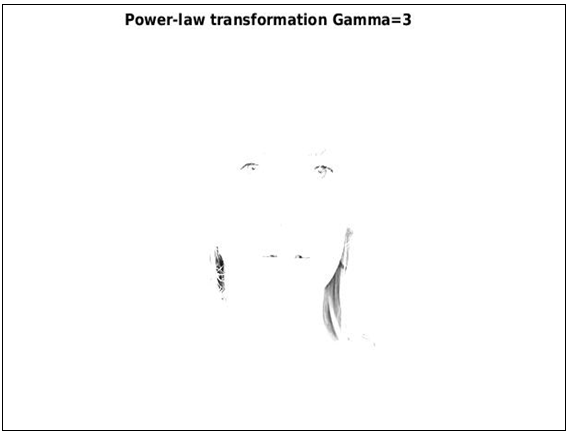
Under exposed image Original:



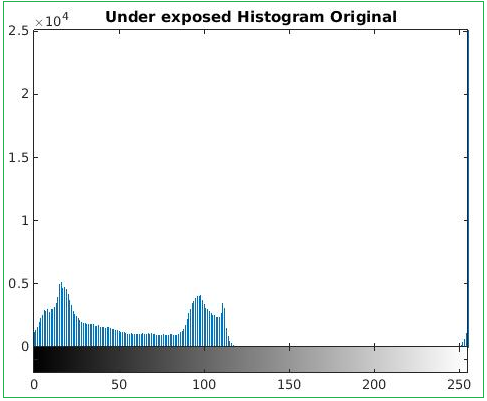




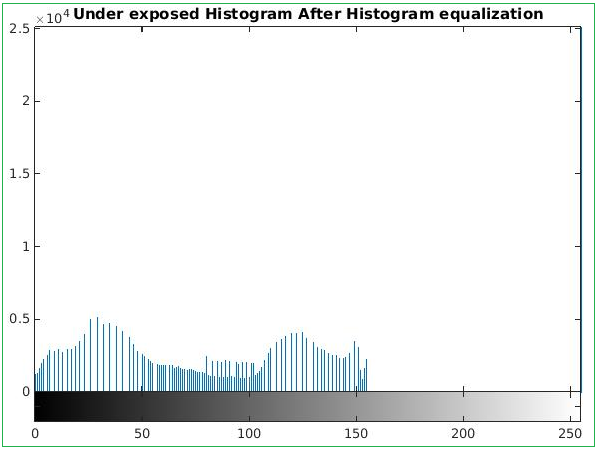




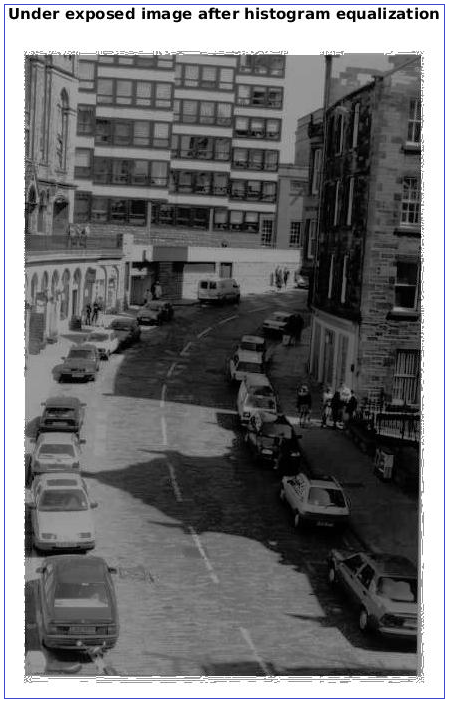
Histogram of under exposed image before HE:



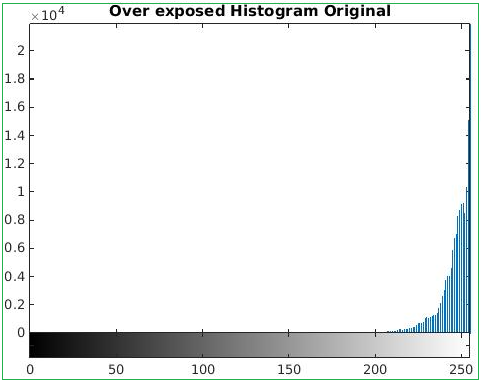
Histogram of under exposed image after HE:



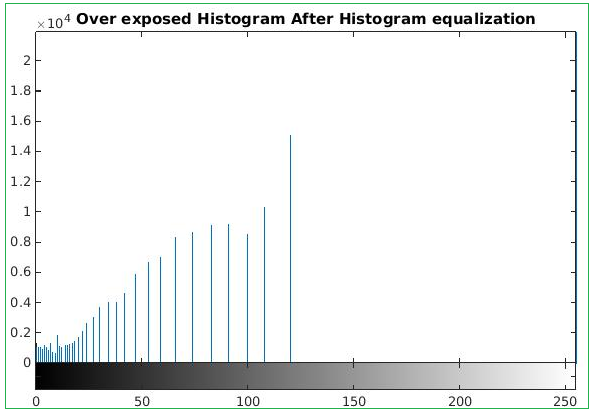
Under exposed image after HE:



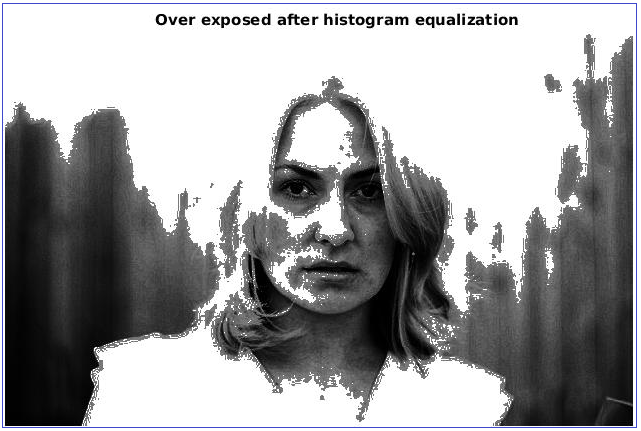
Histogram of over exposed image before HE:



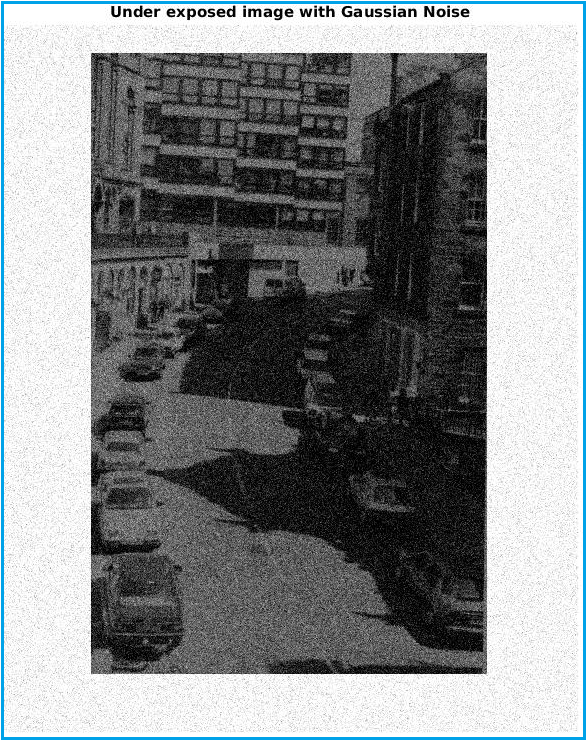
Histogram of over exposed image after HE:



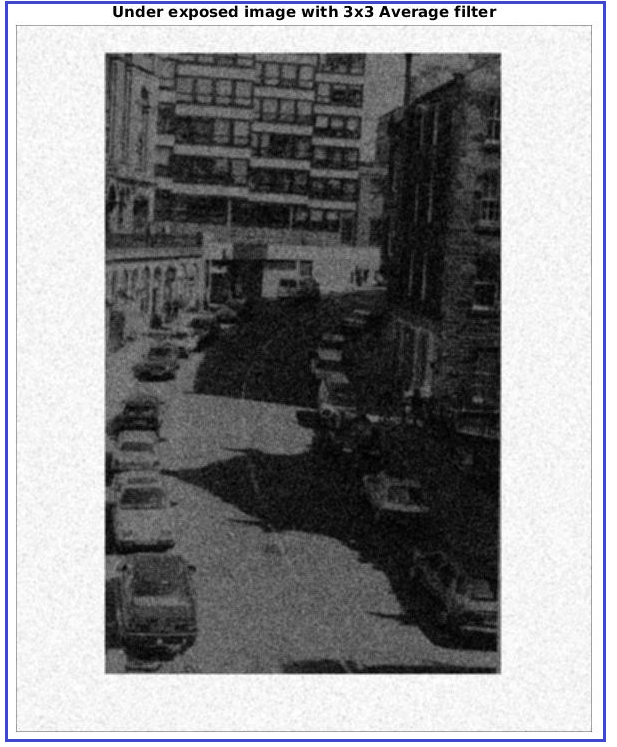
Over exposed image after HE:



Under exposed image with Gaussian noise:



Under exposed image with 3x3 averaging filter:



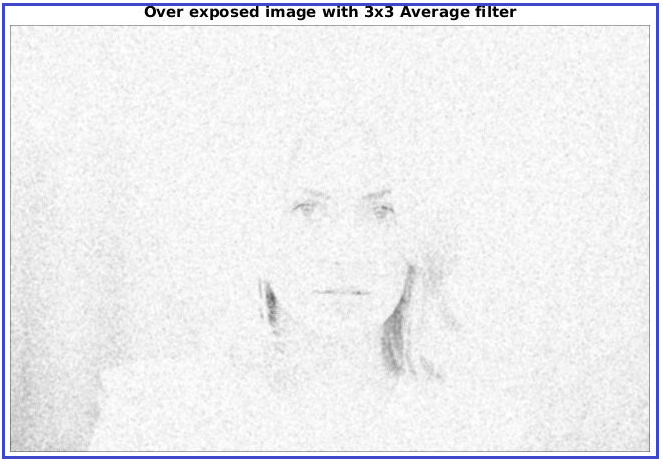
Under exposed image with 5x5 filter:



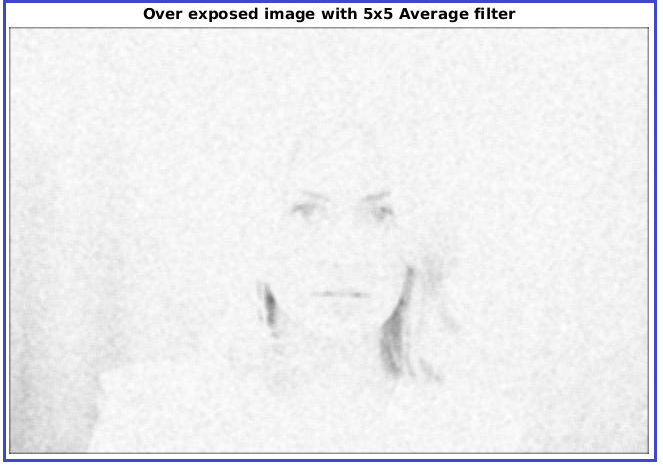
Over exposed image with Gaussian noise:



Over exposed image with 3x3 average filter:



Over exposed image with 5x5 average filter:



**Appendix:**

Program for problem 1:

% Program demonstrates the power law transformation.

Ue = imread('Uexposed.jpg');

Oe = imread('Oexposed.jpg');

% RGB to gray

I = rgb2gray(Ue);

J = rgb2gray(Oe);

% Change to double

I = im2double(I);

J = im2double(J);

% computing size m,n

[m1 n1] = size(I);

[m2 n2] = size(J);

% Computing s = c \* (r ^ gamma) where r and gamma are positive constants

c = 2;

% gamma values

g1=0.3;

g2=3.0;

for p = 1 : m1

for q = 1 : n1

I3(p,q) = c \* I(p,q).^g1;

I4(p,q) = c \* I(p,q).^g2;

end

end

figure, imshow(I3);title('Under exposed,Powerlaw transformation Gamma=0.3');

figure, imshow(I4);title('Under exposed,Power-law transformation Gamma=3');

for p = 1 : m2

for q = 1 : n2

J3(p,q) = c \* J(p,q).^g1;

J4(p,q) = c \* J(p,q).^g2;

end

end

figure, imshow(J3);title('Over exposed,Powerlaw transformation Gamma=0.3');

figure, imshow(J4);title('Over exposed,Power-law transformation Gamma=3');

% end

Program for problem 2:

%Program for problem 2:

%reading the under exposed image

Ue = imread('Uexposed.jpg');

% converting to gray scale image

I = rgb2gray(Ue);

% displaying the histogram of the image

imhist(I);

title('Under exposed Histogram Original');

% histogram equalization

I1 = histeq(I,256);

% histogram of output image

imhist(I1);

title('Under exposed Histogram After Histogram equalization');

figure, imshow(I);

title('Original under exposed image');

figure, imshow(I1);

title('Under exposed image after histogram equalization');

% reading the over exposed image

Oe = imread('Oexposed.jpg');

% converting color image to grey scale

A = rgb2gray(Oe);

% displaying the histogram of the image

imhist(A);

title('Over exposed Histogram Original');

% histogram equalization

A1 = histeq(A,256);

% histogram of output image

imhist(A1);

title('Over exposed Histogram After Histogram equalization');

figure, imshow(A);

title('Original over exposed image');

figure, imshow(A1);

title('Over exposed after histogram equalization');

%end

Program for problem 3:

% Matlab code for problem 3-Noise Addition and removal

% defining 3x3 averaging filter

h = ones(3,3)/9;

% defining 5x5 averaging filter

h5 = ones(5,5)/25;

% Creating the gaussian filter with size = [3 3] and sigma = 1

Gauss3 = fspecial('gaussian',[3 3],1);

% Creating the gaussian filter with size = [5 5] and sigma = 1

Gauss5 = fspecial('gaussian',[5 5],1);

%-------------------------------------------------------------------------%

% reading the input image

Oe = imread('Oexposed.jpg');

% converting to gray image

I = rgb2gray(Oe);

% adding Gaussian noise with default mean values

I1 = imnoise(I,'gaussian');

figure, imshow(I1); title('Oexposed image with Gaussian Noise');

% filtering using 3x3 filter

I2 = imfilter(I1,h);

figure, imshow(I2); title('Oexposed image with 3x3 Average filter');

% filtering using 5x5 filter

I3 = imfilter(I1,h5);

figure, imshow(I3); title('Oexposed image with 5x5 Average filter');

% filtering using 3x3 Gaussian filter

I4 = imfilter(I1,Gauss1,'same');

figure, imshow(I4); title('Oexposed image with 3x3 Gauss filter');

% filtering using 5x5 Gaussian filter

I5 = imfilter(I1, Gauss2, 'same');

figure, imshow(I5); title('Oexposed image with 5x5 Gauss filter');

%-------------------------------------------------------------------------%

% adding Gaussian noise with mean=100 and variance 0.05

J = imnoise(I,'gaussian',100,0.05);

figure, imshow(J); title('Oexposed image with Gaussian Noise; M=100, sigma=0.05');

% filtering using 3x3 averaging filter

J1 = imfilter(J,h);

figure, imshow(J1); title('Oexposed image(Gauss: M=100, v=0.05) 3x3 Average filter');

% filtering using 5x5 averaging filter

J2 = imfilter(J,h5);

figure, imshow(J2); title('Oexposed image(Gauss: M=100, v=0.05) 5x5 Average filter');

% filtering using 3x3 Gaussian filter

J3 = imfilter(J,Gauss1,'same');

figure, imshow(J4); title('Oexposed image(Gauss: M=100, v=0.05) 3x3 Gauss filter');

% filtering using 5x5 Gaussian filter

I5 = imfilter(I1, Gauss2, 'same');

figure, imshow(I5); title('Oexposed image(Gauss: M=100, v=0.05) 5x5 Gauss filter');

%-------------------------------------------------------------------------%

% reading the image

A = imread('Uexposed.jpg');

A = rgb2gray(A);

% adding Gaussian noise with default mean values

A1 = imnoise(A,'gaussian');

figure, imshow(A1); title('Uexposed image with Gaussian Noise');

% filtering using 3x3 averaging filter

A2=imfilter(A1,h);

figure, imshow(A2); title('Uexposed image with 3x3 Average filter');

% filtering using 5x5 averaging filter

A3=imfilter(A1,h5);

figure, imshow(A3); title('Uexposed image with 5x5 Average filter');

% Filtering using Gauss 3x3

A4 = imfilter(A1,Gauss1,'same');

imshow(A4); title('Uexposed image with 3x3 Gaussian filter');

% Filtering using Gauss 5x5

A5 = imfilter(A1,Gauss2,'same');

imshow(Ig); title('Uexposed image with 5x5 Gaussian filter');

%-------------------------------------------------------------------------%

% adding Gaussian noise with mean=100 and variance 0.05

B = imnoise(A,'gaussian',100,0.05);

figure, imshow(B); title('Uexposed image with Gaussian Noise:M=100, sigma=0.05');

% filtering using 3x3 averaging filter

B1 = imfilter(B,h);

figure, imshow(B1); title('Uexposed image(Gauss: M=100, v=0.05) 3x3 Average filter');

% filtering using 5x5 averaging filter

B2 = imfilter(B,h5);

figure, imshow(B2); title('Uexposed image(Gauss: M=100, v=0.05) 5x5 Average filter');

% filtering using 3x3 Gaussian filter

B3 = imfilter(B,Gauss1,'same');

figure, imshow(B3); title('Uexposed image(Gauss: M=100, v=0.05) 3x3 Gauss filter');

% filtering using 5x5 Gaussian filter

B4 = imfilter(B, Gauss2, 'same');

figure, imshow(B4); title('Uexposed image(Gauss: M=100, v=0.05) 5x5 Gauss filter');

%-------------------------------------------------------------------------%

% end

Program for Problem 4:

% Matlab code for problem 4-Laplacian Enhancement

% defining the simple laplacian filter

lap1 = [0 1 0; 1 -4 1; 0 1 0];

% laplacian filter involving the diagonal pixels as well

lap2 = [1 1 1; 1 -8 1; 1 1 1];

% lap2 with negation

lap3 = [-1 -1 -1; -1 8 -1; -1 -1 -1];

Ue = imread('Uexposed1.jpg');

I = rgb2gray(Ue);

I1 = imfilter(I,lap1);

I2 = imfilter(I,lap2);

%I3 = imfilter(I,lap3);

figure, imshow(I); title('Uexposed Original');

figure, imshow(I1); title('Uexposed filtered using Laplacian');

figure, imshow(I2); title('Uexposed filtered using Laplacian diagonal');

%figure, imshow(I3); title('Uexposed filtered using Laplacian diagonal inverse');

Oe = imread('Oexposed.jpg');

A = rgb2gray(Oe);

A1 = imfilter(A,lap1);

A2 = imfilter(A,lap2);

figure, imshow(A); title('Over exposed Original');

figure, imshow(A1); title('Over exposed using Laplacian filter');

figure, imshow(A2); title('Oexposed using Laplacian filter diagonal');

%end