# EE 440 Autumn 2018
# Homework 4 Report

Professor: Ming-Ting Sun

October 21st, 2018

He Feng 1427841

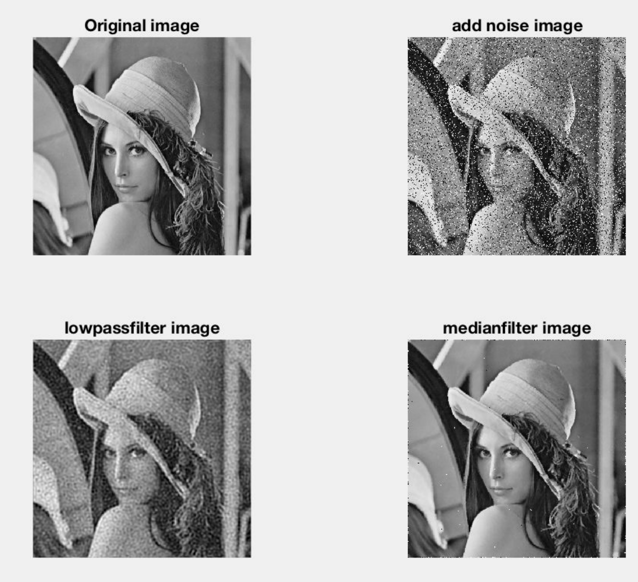
## Problem 1

In this problem we are going to add the salt and pepper noise which will cause 15% pixels being altered. Then we are going to use the low pass filter and median filter to generate the altered image to reduce the noise. Here is the original image of the problem:

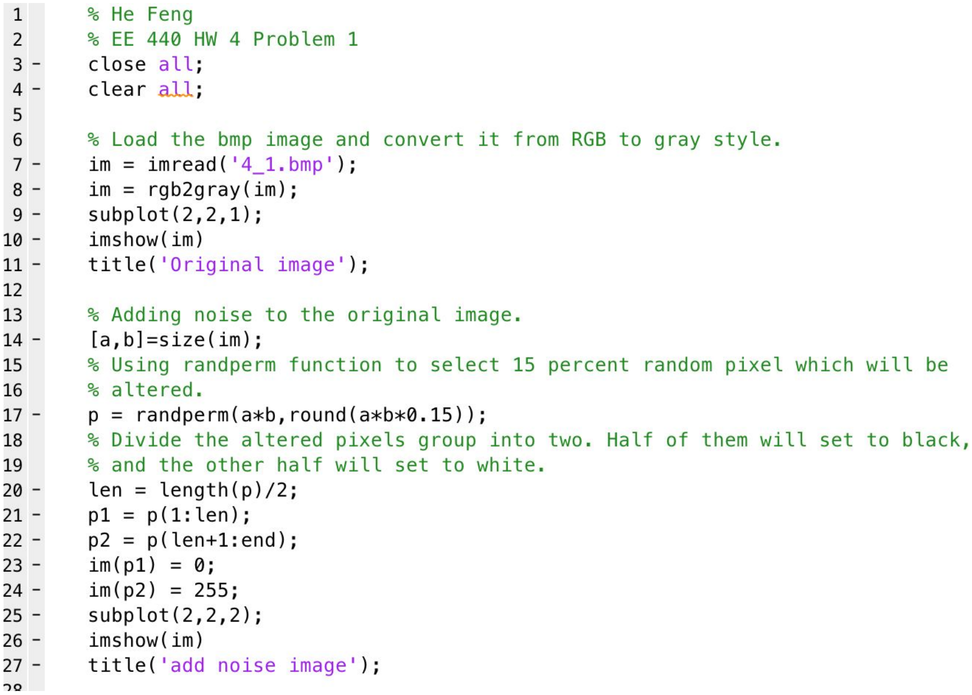
For the Matlab code, I did not write any functions to support this problem. Initially, I load the image by using imread function, then I display the original image into the subplot. Then I add the salt and pepper noise to it. I used the randperm function to select fifteen percent of pixels which will be altered, and I divided them into two groups evenly. The first group will be set to white and the second group will be set to black, and I display the altered image to the second subplot in the same figure. After the steps above, I converted the image to double type such that I can process the image and reduce the noise later.

To reduce the noise by using low pass filter, I created two vectors whose size are the same as the original image's row and column size. Then I used meshgrid function two create two matrices which will be used later to select pixel. Then I use square function to compute distance from point (u,v) to center of frequency rectangle. For this problem, I select D0 as 70 for this low pass filter, and I build a matrix which only contains the pixels that are smaller than 70 in this case. I transfer the double typed image into fourier transform , and multiply it with the matrix which has the pixels under 70. For this last matrix I developed, I passed it to the reverse fourier transform and display the output result to the third subplot. The output result is the reduced noise image by using the low pass filter.

The last section of this problem is to use median filter to reduce the noise. I still need to convert the image into the double type, which has already be converted before. Then I used two for loops to make nine pixels as a group. I chose the median number among the nine pixels, and I used this as the output number for the center pixel of the group for the new image. After passing through the for loop I have already get the output result for all the pixels, and I just need to display it into the fourth subplot of the figure. The output images are showing below:



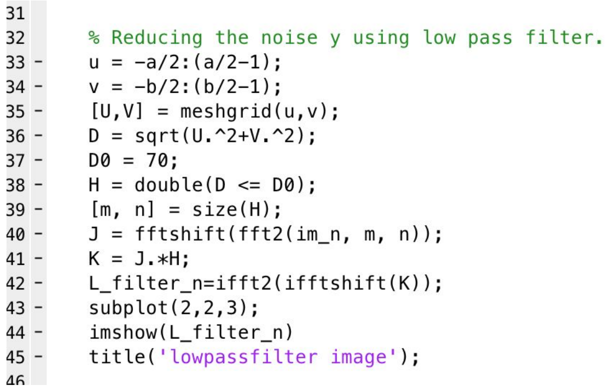
The Matlab code for this problem are showing below:



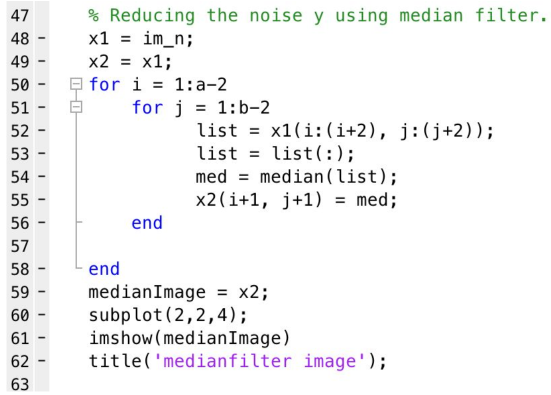
```
% He Feng
% EE 440 HW 4 Problem 1
close all;
clear all;

% Load the bmp image and convert it from RGB to gray style.
im = imread('4_1.bmp');
im = rgb2gray(im);
subplot(2,2,1);
imshow(im)
title('Original image');

% Adding noise to the original image.
[a,b]=size(im);
% Using randperm function to select 15 percent random pixel which will be
% altered.
p = randperm(a*b,round(a*b*0.15));
% Divide the altered pixels group into two. Half of them will set to black,
% and the other half will set to white.
len = length(p)/2;
p1 = p(1:len);
p2 = p(len+1:end);
im(p1) = 0;
im(p2) = 255;
subplot(2,2,2);
imshow(im)
title('add noise image');
```



```
% Reducing the noise y using low pass filter.
u = -a/2:(a/2-1);
v = -b/2:(b/2-1);
[U,V] = meshgrid(u,v);
D = sqrt(U.^2+V.^2);
D0 = 70;
H = double(D <= D0);
[m, n] = size(H);
J = fftshift(fft2(im_n, m, n));
K = J.*H;
L_filter_n=ifft2(ifftshift(K));
subplot(2,2,3);
imshow(L_filter_n)
title('lowpassfilter image');
```



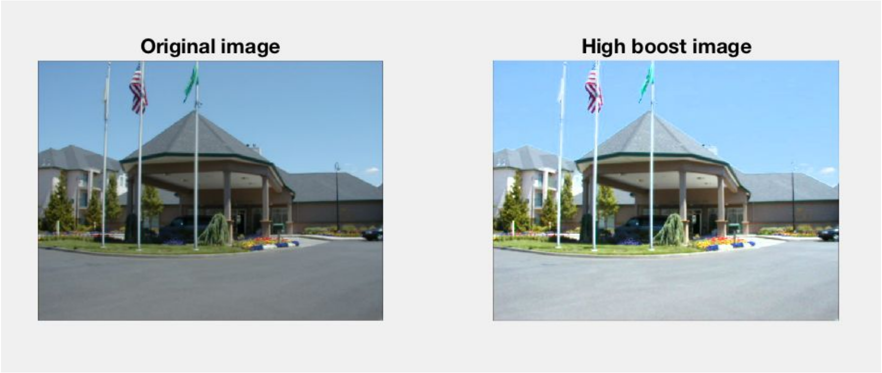
```
% Reducing the noise y using median filter.
x1 = im_n;
x2 = x1;
for i = 1:a-2
    for j = 1:b-2
          list = x1(i:(i+2), j:(j+2));
          list = list(:);
          med = median(list);
          x2(i+1, j+1) = med;
    end

end
medianImage = x2;
subplot(2,2,4);
imshow(medianImage)
title('medianfilter image');
```

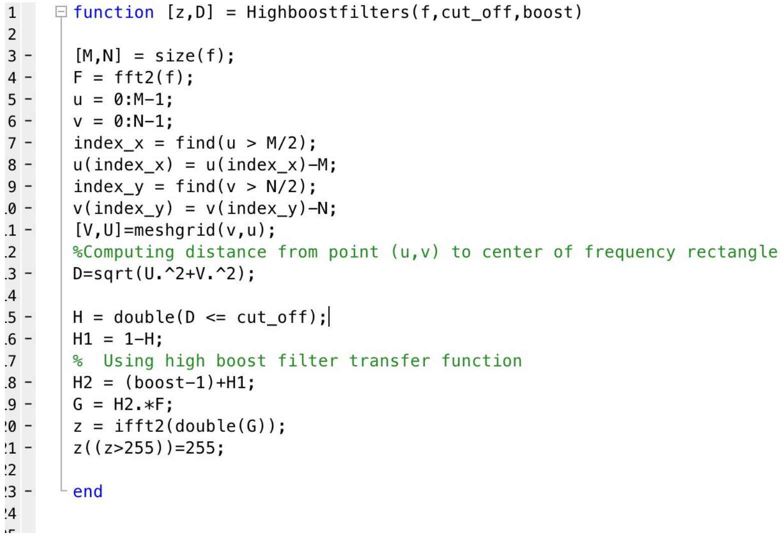
## Problem 2

For this problem we need to sharpen the image by using boost filter. I create a boost filter function for this problem. In the function, I initially get the size of the given image, and then convert it into fourier transform. Similar to the last problem, I created two variables which has the same size of the image's column number and row number. Because this is a boost filter, we need to control the index of the two variables above. Then I used meshgrid to compute two matrices, and I used the square function to compute the distance from (u,v), which are the two variables I defined before, to center of the frequency triangle. Then I used the boost filter transfer function, a new matrix whose pixels value are smaller than the cut-off value, and the boost value to compute a new output. After I convert this output result by using reverse fourier transform, I set the value which exceed 255 to 255.

In the main script of this problem, I initially display the original image in the first subplot. Then I convert the image to HSV style, and I abstract the V image and let it passed through the boost filter. I divided the output to 255 and I convert it back to the RGB image. Finally I display it in the subplot of this figure. Here are the output image:



Here is the Matlab code for the function:



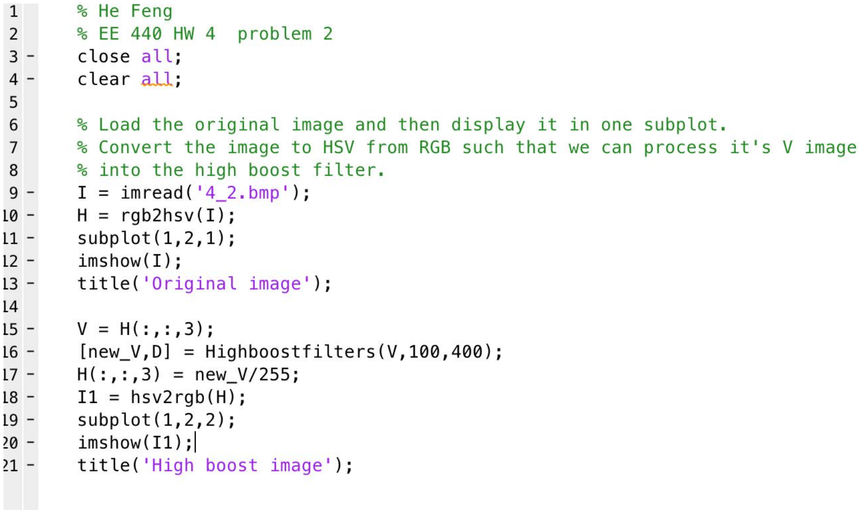
```
function [z,D] = Highboostfilters(f,cut_off,boost)

[M,N] = size(f);
F = fft2(f);
u = 0:M-1;
v = 0:N-1;
index_x = find(u > M/2);
u(index_x) = u(index_x)-M;
index_y = find(v > N/2);
v(index_y) = v(index_y)-N;
[V,U]=meshgrid(v,u);
%Computing distance from point (u,v) to center of frequency rectangle
D=sqrt(U.^2+V.^2);

H = double(D <= cut_off);
H1 = 1-H;
%  Using high boost filter transfer function
H2 = (boost-1)+H1;
G = H2.*F;
z = ifft2(double(G));
z((z>255))=255;

end
```

Here is the Matlab code for the problem2.m:



```
% He Feng
% EE 440 HW 4  problem 2
close all;
clear all;

% Load the original image and then display it in one subplot.
% Convert the image to HSV from RGB such that we can process it's V image
% into the high boost filter.
I = imread('4_2.bmp');
H = rgb2hsv(I);
subplot(1,2,1);
imshow(I);
title('Original image');

V = H(:,:,3);
[new_V,D] = Highboostfilters(V,100,400);
H(:,:,3) = new_V/255;
I1 = hsv2rgb(H);
subplot(1,2,2);
imshow(I1);
title('High boost image');
```