Name: Brihat Ratna Bajracharya

Roll No.: 19/075

Solution to Exercise 4 (Image Processing and Pattern Recognition)

Thresholding

Question 1. Consider the following 8 by 8 image:

3	148	117	148	145	178	132	174
2	176	174	110	185	155	118	165
0	100	124	113	193	136	146	108
0	155	170	106	158	130	178	170
9	196	138	113	108	127	144	139
6	188	143	183	137	162	105	169
9	122	156	119	188	179	100	151
8	176	137	114	135	123	134	183

Threshold it at

(a) level 100

(b) level 150

What happens to the results of thresholding as the threshold level is increased?

Solution

```
clearall;

mat_8_8 = [3 148 117 148 145 178 132 174;
2 176 174 110 185 155 118 165;
0 100 124 113 193 136 146 108;
0 155 170 106 158 130 178 170;
9 196 138 113 108 127 144 139;
6 188 143 183 137 162 105 169;
9 122 156 119 188 179 100 151;
8 176 137 114 135 123 134 183;];

figure
subplot(2,2,[1 2]);
imshow(mat2gray(mat_8_8, [min(mat_8_8(:))
max(mat_8_8(:))]),'InitialMagnification','fit');
title('Original 8 * 8 Matrix');
```

```
threshold = 100;
new_mat_8_8 = mat_8_8 > threshold;
subplot(2,2,3);
imshow(new_mat_8_8,'InitialMagnification','fit');
title(['(a) Threshold = ' num2str(threshold)]);

threshold = 150;
new_mat_8_8 = mat_8_8 > threshold;
subplot(2,2,4);
imshow(new_mat_8_8,'InitialMagnification','fit');
title(['(b) Threshold = ' num2str(threshold)]);
```

Output

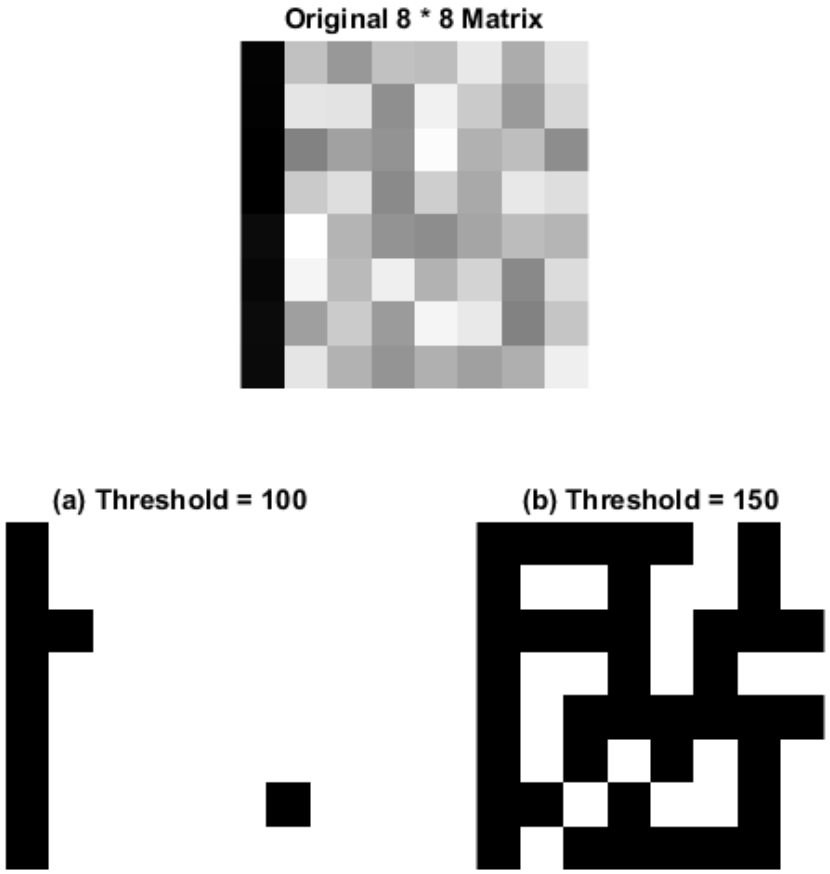

Figure 1: Solution to Question 1

Discussion

The more the threshold value is incresed, more finer gray shade is considered as black pixel.

Question 2. Superimpose the image text.tif onto the image cameraman.tif. You can do this with:

```
>> t = imread('text.tif');
>> c = imread('cameraman.tif');
>> m = uint8(double(c) + 255 * double(t));
```

Can you threshold this new image m to isolate the text?

Solution

```
clearall;

t = imread('text.tif');
c = imread('cameraman.tif');

figure
subplot(2,2,1);
imshow(t,'InitialMagnification','fit');
title('(1) text.tif');

subplot(2,2,2);
imshow(c,'InitialMagnification','fit');
title('(2) cameraman.tif');

m = uint8(double(c) + 255 * double(t));
subplot(2,2,3);
imshow(m,'InitialMagnification','fit');
title('(3) Superimposed result');

m_text_isolate = m >= 255;
subplot(2,2,4);
imshow(m_text_isolate,'InitialMagnification','fit');
title('(4) Text isloated from (3)');
```

Description

Yes, we can isolate the text only from the superimposed images by setting the threshold to 255 (only completely white pixels in superimposed image). The result in shown in Figure 2 (4).

Output

Figure 2: Solution to Question 2

Question 3. Create a function that does a contrast stretching transformation for different values of m and E. Apply it to spectrum.tif. Also try a logarithmic transformation.

Hint:

```
g = c * log(1 + double(f));
g = 1 ./ (1 + (m ./ (double(f) + eps)) .^ E);
```

Solution

```
function e4q3
    function output = contrast_stretch(f, m, E)
        output = 1 ./ (1 + (m ./ (double(f) + eps)) .^ E);
```

```
end

function output = log_transform(c, f)
    output = c * log(1 + double(f));
end

close all;
f = imread('spectrum.tif');
m = mean2(f);
figure
subplot(3,5,1);
imshow(f);
title('spectrum.tif');

cf = contrast_stretch(f, m, 0.25);
subplot(3,5,2);
imshow(cf);
title('CS (m=2.08, E=0.25)');

cf = contrast_stretch(f, m, 0.5);
subplot(3,5,3);
imshow(cf);
title('CS (m=2.08, E=0.5)');

cf = contrast_stretch(f, m, 1);
subplot(3,5,4);
imshow(cf);
title('CS (m=2.08, E=1)');

cf = contrast_stretch(f, m, 2);
subplot(3,5,5);
imshow(cf);
title('CS (m=2.08, E=2)');

cf = contrast_stretch(f, 16, 0.5);
subplot(3,5,6);
imshow(cf);
title('CS (m=2.08, E=16)');

cf = contrast_stretch(f, 0.5, 0.5);
subplot(3,5,7);
```

```
imshow(cf);
title('CS (m=0.5, E=0.5)');

cf = contrast_stretch(f, 1, 0.5);
subplot(3,5,8);
imshow(cf);
title('CS (m=1, E=0.5)');

cf = contrast_stretch(f, 2, 0.5);
subplot(3,5,9);
imshow(cf);
title('CS (m=2, E=0.5)');

cf = contrast_stretch(f, 4, 0.5);
subplot(3,5,10);
imshow(cf);
title('CS (m=4, E=0.5)');

cf = contrast_stretch(f, 8, 0.5);
subplot(3,5,11);
imshow(cf);
title('CS (m=8, E=0.5)');

c = 0.25;
g = im2uint8(mat2gray(log_transform(c, f)));
subplot(3,5,12);
imshow(g);
title(['log transform (c=' num2str(c) ')']);

c = 0.5;
g = im2uint8(mat2gray(log_transform(c, f)));
subplot(3,5,13);
imshow(g);
title(['log transform (c=' num2str(c) ')']);

c = 1;
g = im2uint8(mat2gray(log_transform(c, f)));
subplot(3,5,14);
imshow(g);
title(['log transform (c=' num2str(c) ')']);
```

```
    c = 2;
    g = im2uint8(mat2gray(log_transform(c, f)));
    subplot(3,5,15);
    imshow(g);
    title(['log transform (c=' num2str(c) ')']);
end
```

Output

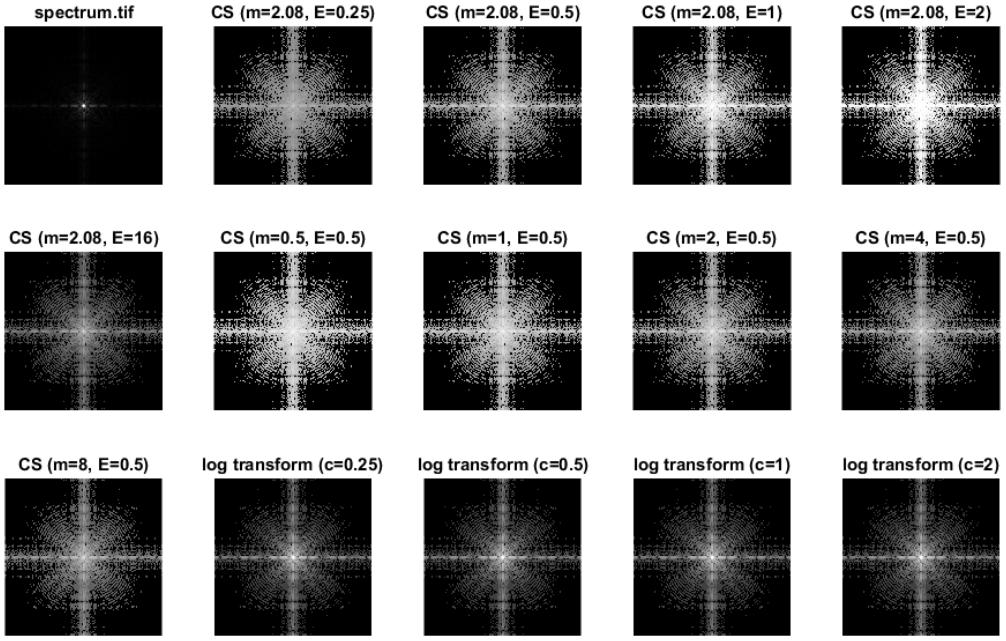

Figure 3: Solution to Question 3 (CS= Contrast Stretch),
CS is done for m=0.5, 1, 2, 4, 8 with E=0.5, and E=0.25, 0.5, 1, 2 for m=2.08
and log transform for c=0.25, 0.5, 1, 2

Question 4. Try the following commands (negative image)

```
>> f = imread('chest-xray.tif');
>> imshow(f)
>> g1 = imadjust(f);
>> imshow(g1)
>> g2 = imadjust(f, [0 1], [1 0]);
>> figure,imshow(g2)
```

How does changing the value of gamma affect the image?

What types of images are different values of gamma good for?

Solution

```
clearall;

f = imread('chest-xray.tif');

g1 = imadjust(f);
g2 = imadjust(f, [0 1], [1 0]);
g3 = imadjust(f, [0 1], [1 0], 0.5);
g4 = imadjust(f, [0 1], [1 0], 1);
g5 = imadjust(f, [0 1], [1 0], 2);
g6 = imadjust(f, [0 1], [1 0], 4);
g7 = imadjust(f, [0 1], [1 0], 8);

figure
subplot(3,3,1);
imshow(f);
title('chest-xray.tif');

subplot(3,3,2);
imshow(g1);
title('imadjust result');

subplot(3,3,3);
imshow(g2);
title('imadjust inverted result');

subplot(3,3,4);
imshow(g3);
title('\gamma = 0.5');

subplot(3,3,5);
imshow(g4);
title('\gamma = 1');

subplot(3,3,6);
imshow(g5);
title('\gamma = 2');

subplot(3,3,7);
imshow(g6);
```

```
title('\gamma = 4');

subplot(3,3,8);
imshow(g7);
title('\gamma = 8');
```

Description

Gamma specifies the shape of the curve that maps the intensity values of f to create g. If gamma is less than 1, the mapping is weighted toward higher (brighter) output values. If gamma is greater than 1, the mapping is weighted toward lower (darker) output values. The default value for gamma is 1 (linear mapping). Appropriate gamma values for the images depends upon the type of display it is displayed on.

Output

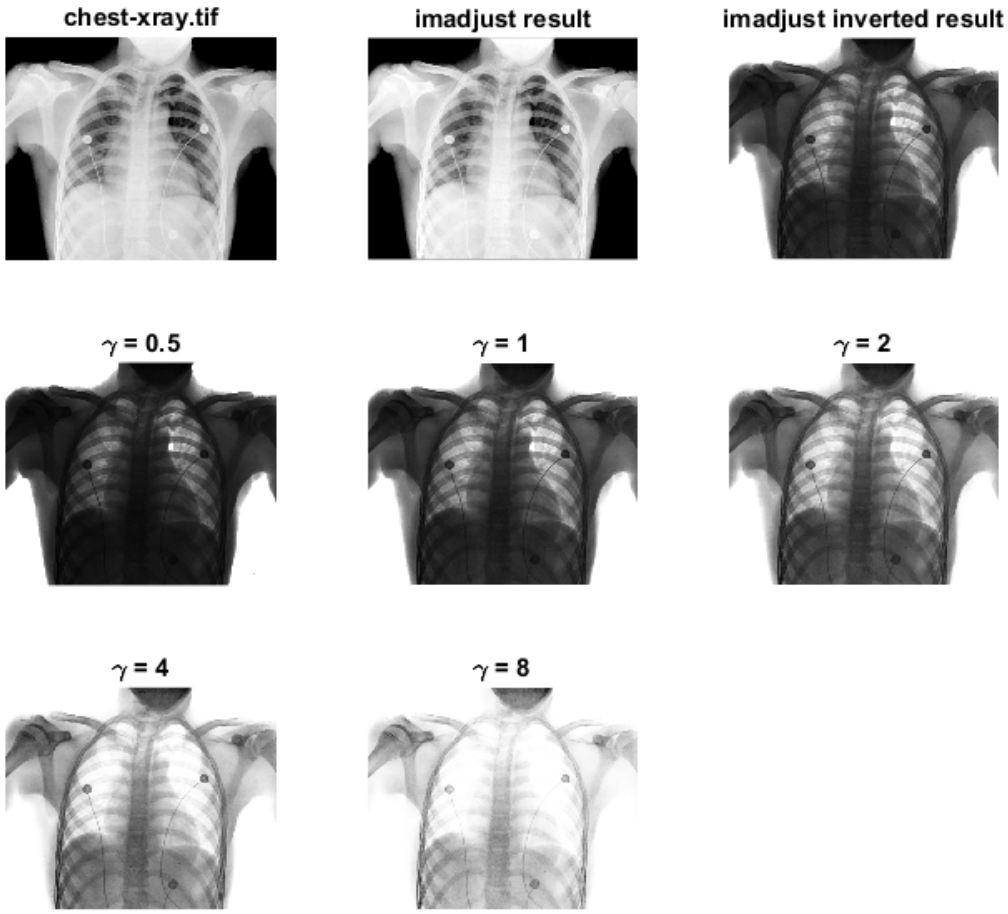

Figure 4: Solution to Question 4

Question 5. A large variety of image processing tasks can be accomplished by a technique called spatial filtering. Spatial filtering involves a mask, consists of an array of values (a-i) and has a center (gray). and translated across all possible pixel positions on the image. A new (filtered) image is produced by replacing the intensity value at the center by a linear combination of the intensity values of the center pixel and all neighboring pixels covered by the mask.

1. Try adding different types of noise to ckt-board-orig.tif. Look in the help for the different options in imnoise. Then choose the appropriate filter to remove the noise.
2. Repeat for pollen.tif.
3. Capture your own image and repeat the task as in 1.

Solution

```
clearall;

% FOR E4Q5 1.
% f = imread('ckt-board-orig.tif');
% FOR E4Q5 2.
% f = imread('pollen.tif');
% FOR E4Q5 3.
f = rgb2gray(imread('brihat.png'));

figure
subplot(2,3,1);
imshow(f)
title('original image');

fn1 = imnoise(f, 'gaussian', 0.5, 0.02);
subplot(2,3,2);
imshow(fn1)
title('gaussian (\mu = 0.5, \sigma^2 = 0.02)');

fn2 = imnoise(f, 'localvar', 0.04 * rand(size(f)));
subplot(2,3,3);
imshow(fn2)
title('localvar');

fn3 = imnoise(f, 'poisson');
subplot(2,3,4);
imshow(fn3)
```

```
title('poisson');

fn4 = imnoise(f, 'salt & pepper', 0.2);
subplot(2,3,5);
imshow(fn4)
title('salt & pepper (density = 0.2)');

fn5 = imnoise(f, 'speckle', 0.05);
subplot(2,3,6);
imshow(fn5)
title('speckle (\sigma^2 = 0.05)');

% For E4Q5 1.
noisy_image = fn1;

figure
subplot(3,3,1);
imshow(noisy_image);
title('noisy image');

fil1 = fspecial('average', 6);
fil2 = fspecial('disk', 5);
fil3 = fspecial('gaussian', 15, 2);
fil4 = fspecial('laplacian', 0);
fil5 = fspecial('log', 3, 0.6);
fil6 = fspecial('prewitt');
fil7 = fspecial('sobel');

fil_img1 = imfilter(noisy_image, fil1);
subplot(3,3,2);
imshow(fil_img1);
title('average (size = 6)');

fil_img2 = imfilter(noisy_image, fil2);
subplot(3,3,3);
imshow(fil_img2);
title('disk (radius = 5)');

fil_img3 = imfilter(noisy_image, fil3);
subplot(3,3,4);
imshow(fil_img3);
```

```
title('gaussian (size = 15, \sigma = 2)');

fil_img4 = imfilter(noisy_image, fil4, 'circular');
subplot(3,3,5);
imshow(fil_img4);
title('laplacian (\alpha = 0)');

fil_img5 = imfilter(noisy_image, fil5);
subplot(3,3,6);
imshow(fil_img5);
title('log (size = 3, \sigma = 0.6)');

fil_img6 = imfilter(noisy_image, fil6);
subplot(3,3,7);
imshow(fil_img6);
title('prewitt');

fil_img7 = imfilter(noisy_image, fil7);
subplot(3,3,8);
imshow(fil_img7);
title('sobel');

fil_img_sp = medfilt2(noisy_image);
subplot(3,3,9);
imshow(fil_img_sp);
title('median');
% END FOR E4Q5 1.

% FOR E4Q5 2. and 3.
figure
subplot(2,3,1);
imshow(f);
title('original image');

fil3 = fspecial('gaussian', 15, 2);

fil_img_sp = medfilt2(fn1);
subplot(2,3,2);
imshow(fil_img_sp);
title('gaussian - median');
```

```
fil_img = imfilter(fn2, fil3);
subplot(2,3,3);
imshow(fil_img);
title('localvar - gaussian');

fil_img_sp = medfilt2(fn3);
subplot(2,3,4);
imshow(fil_img_sp);
title('poisson - median');

fil_img_sp = medfilt2(fn4);
subplot(2,3,5);
imshow(fil_img_sp);
title('salt & pepper - median');

fil_img_sp = medfilt2(fn5);
subplot(2,3,6);
imshow(fil_img_sp);
title('speckle – median');
% END FOR E4Q5 2. and 3.
```

Outputs

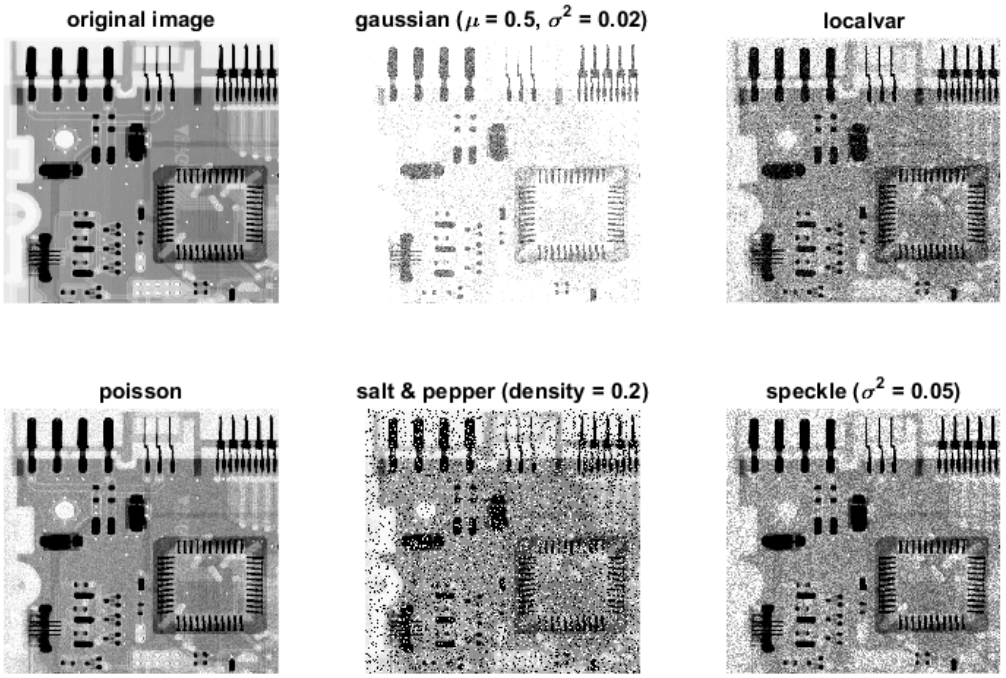

Figure 5: Adding noise to ckt-board-orig.tif

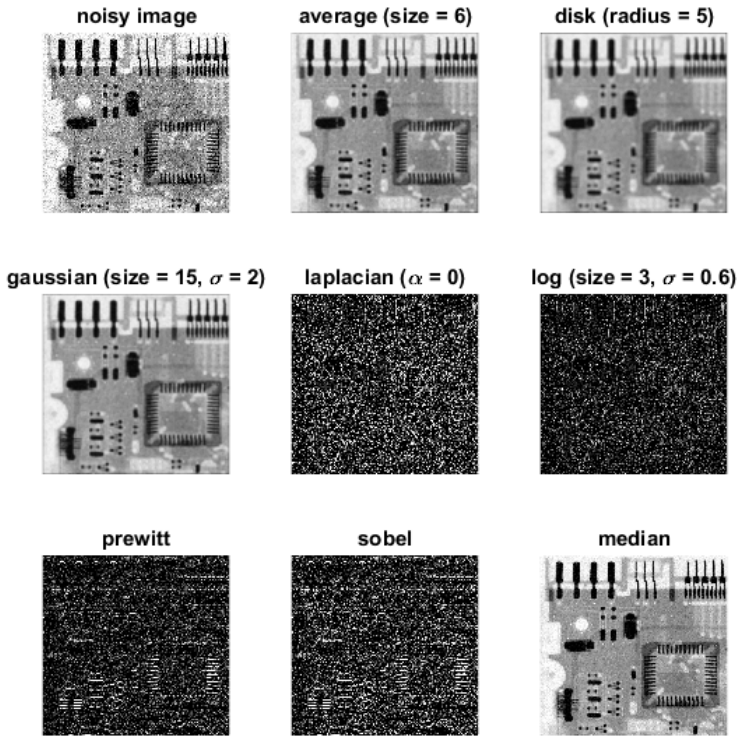

Figure 6: Different filters for localvar noise

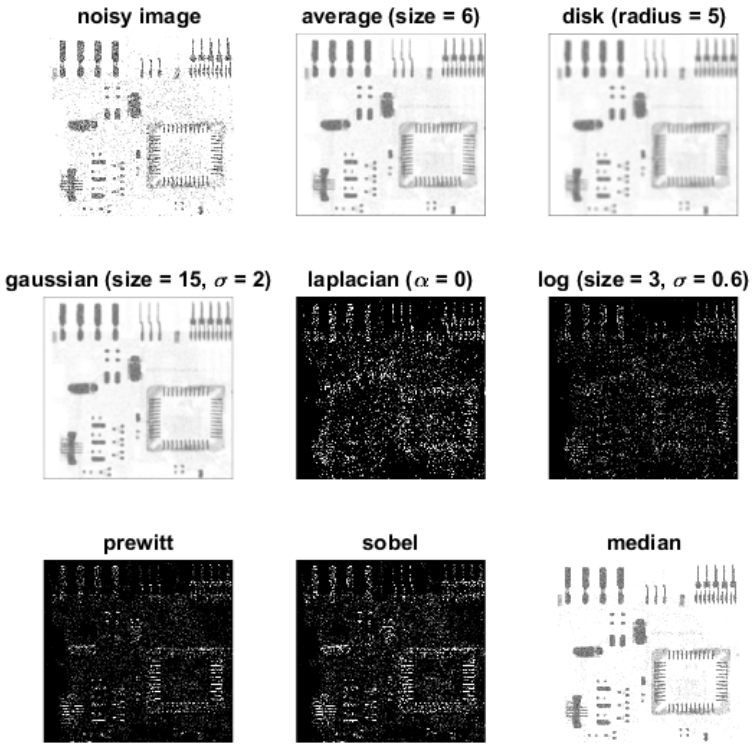

Figure 7: Different filters for gaussian noise

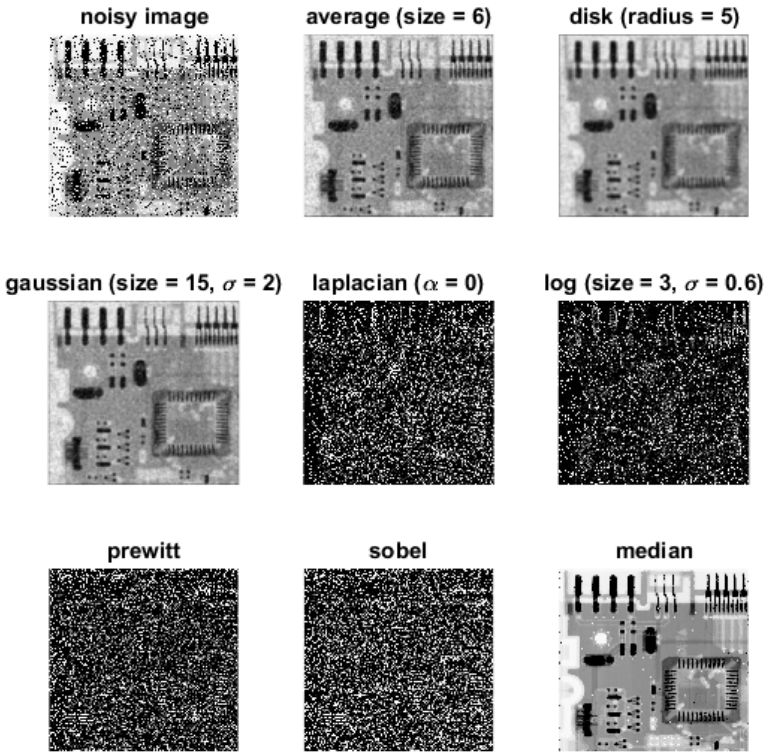

Figure 8: Different filters for salt and pepper noise

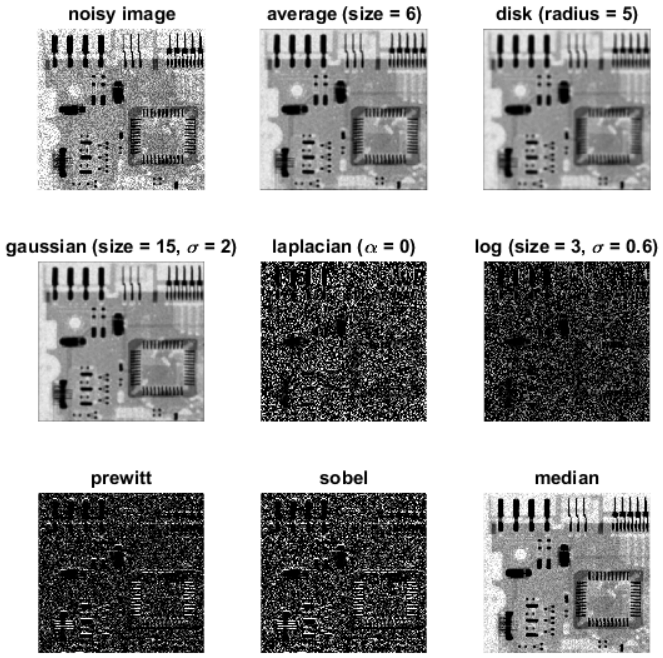

Figure 9: Different filters for speckle noise

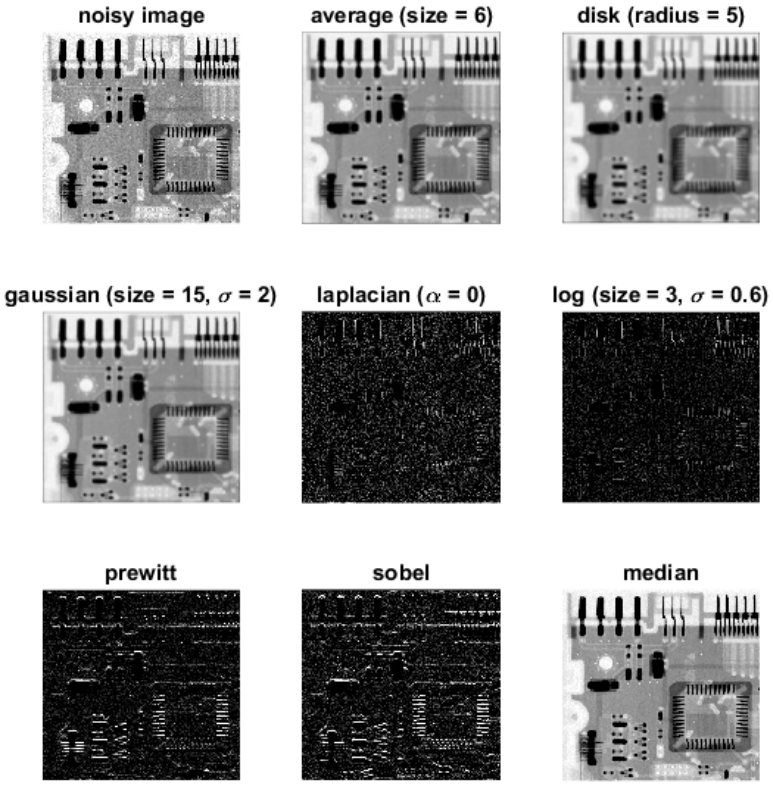

Figure 10: Different filters for poisson noise

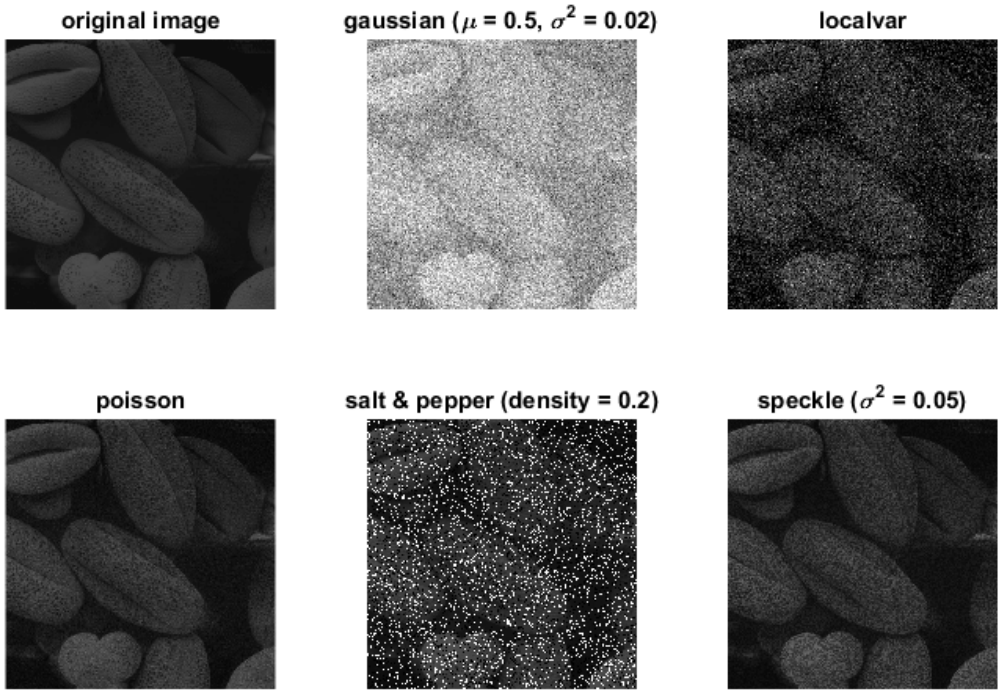

Figure 11: Pollen.tif under different noises

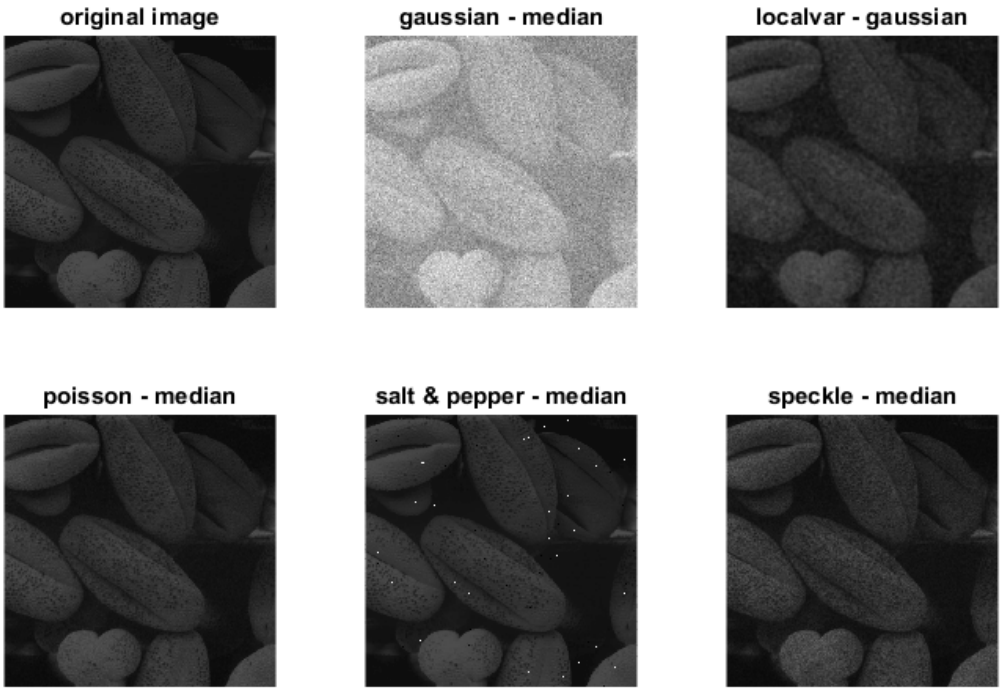

Figure 12: Filtered images for different noises (pollen.tif)

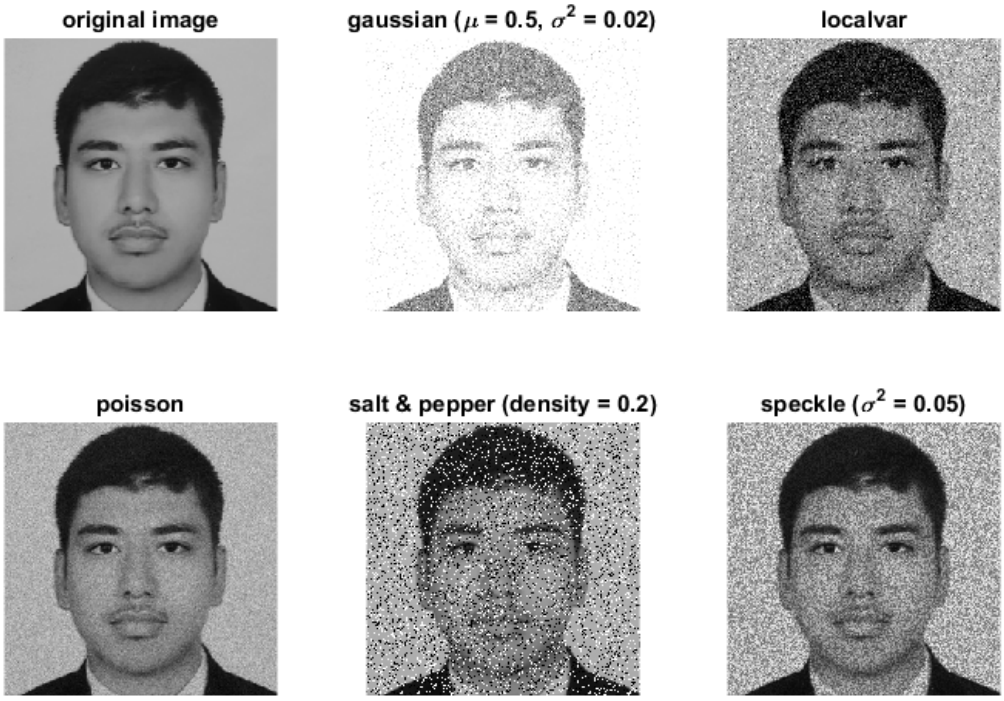

Figure 13: My picture under different noises

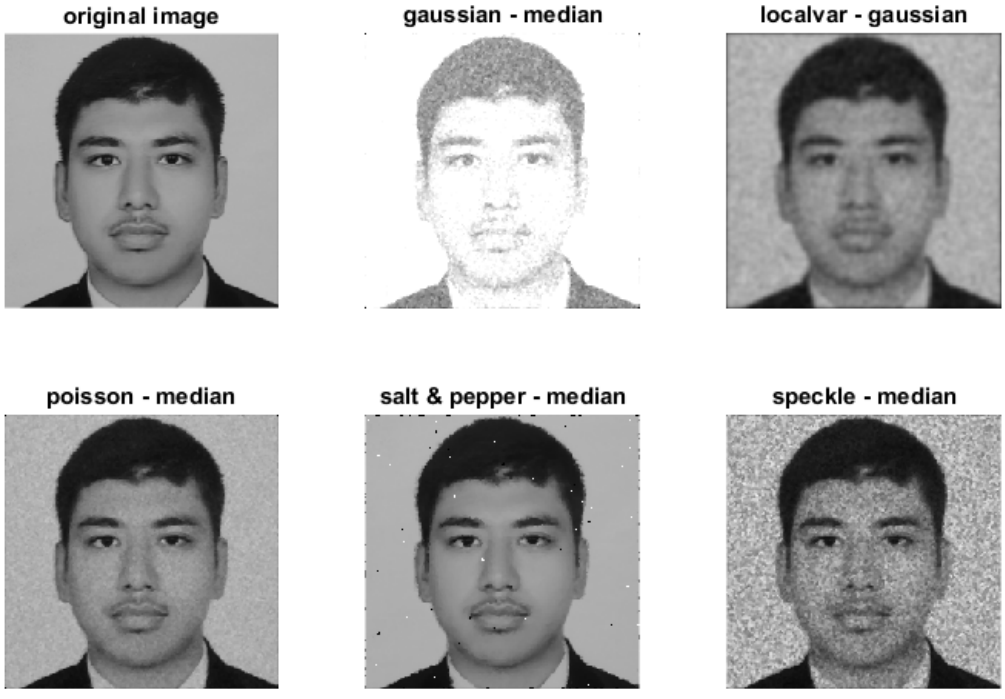

Figure 14: My filtered pictures from different noises