KAUNAS UNIVERSITY OF TECHNOLOGY

Electrical and Electronics faculty
Electronics engineering dep.

Biomedical Image Processing and Analysis

Laboratory work report

Lab 1

Alexandros Veremis, (Erasmus 2020)
Submission date: 2020-03-31

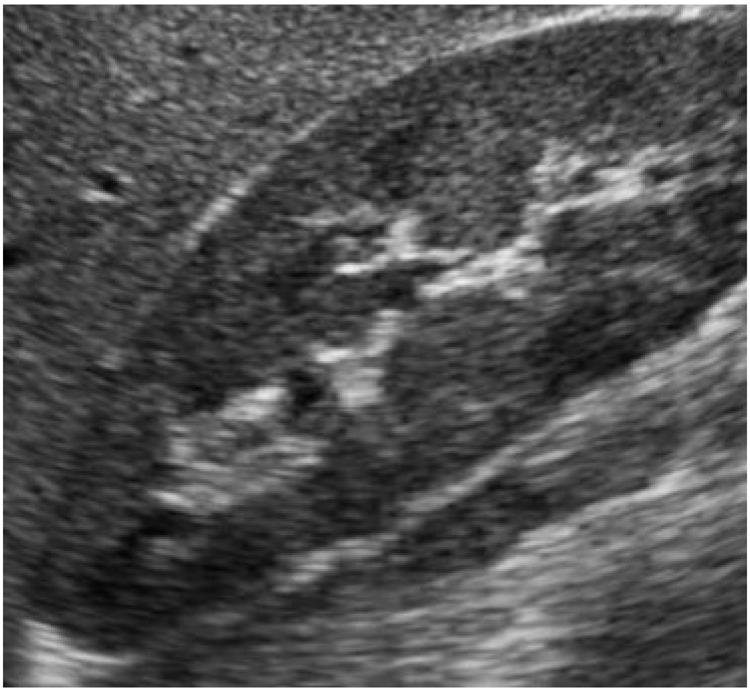
KAUNAS

REPORT FOR LABORATORY 1:

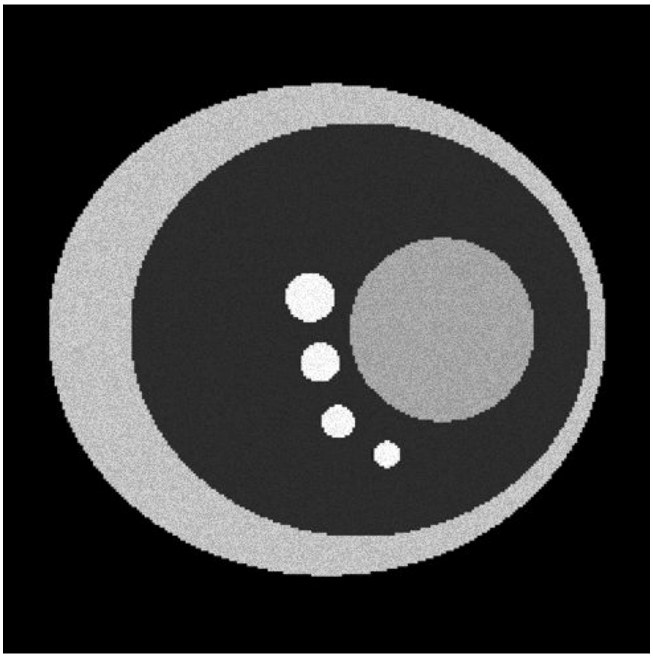
A) Introduction and Problem Statement

I am using Matlab and I will solve the below problems:
1. Mean filter. Implement smoothing local mean filter.
2. Application. Try to filter given images ‘synthetic_image.png’ and ‘kidney.png’. Observe the effects of filtering by applying: (i) 3 × 3 window, (ii) 5 × 5 window; (iii) 9 × 9 window; (iiii) 13 × 13 window etc.
3. Lee filter. Implement noise-adaptive Lee filter. What are the main difference between mean (eq. 1) and Lee filters (eq. 2)?
4. Application. Apply Lee filter on given images ‘synthetic_image.png’ and ‘kidney.png’. Observe the effects of filtering by applying: (i) 3 × 3 window, (ii) 5 × 5 window; (iii) 9 × 9 window; (iiii) 13 × 13 window etc.
5. Comparison. Compare the results of filtering in both cases (use zoomed images and image line profiles). Does the filtering reduces intensity variation in the image? How the filtering affects edges of the image? How the ideal boundary between different regions could be defined? How the size k is related to smoothing effect? Estimate contrast to noise ratio CNR of: (i) original image, filtered by (ii) local mean filter, filtered by (iii) adaptive Lee filter.

B)Figures and Results - C) Comments And Conclusion

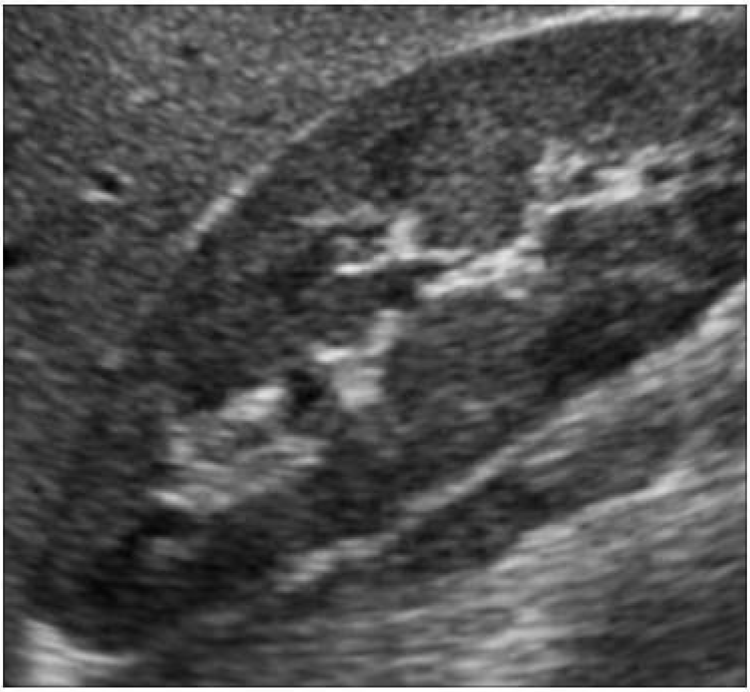


The original Kidney Image

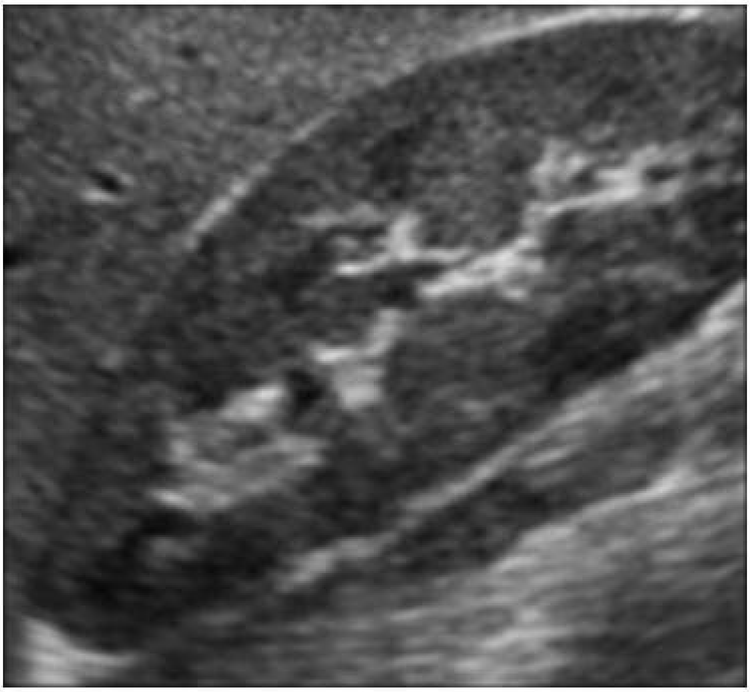


The original Synthetic Image

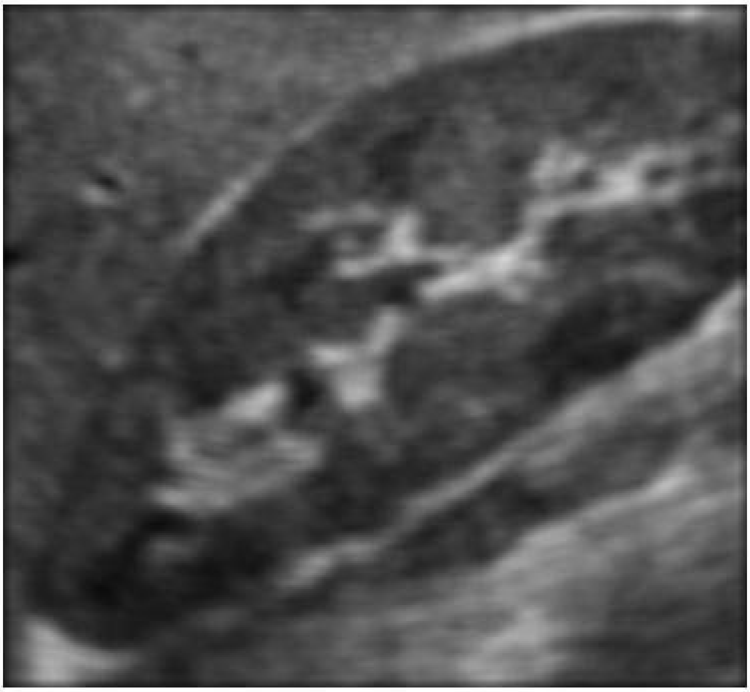
2) We observe that : The bigger the window, the blurrer the image. The pixel values are changed and so the different regions in the image are more difficult now to be distinguished. The reason behind it is that the sharp features are removed.



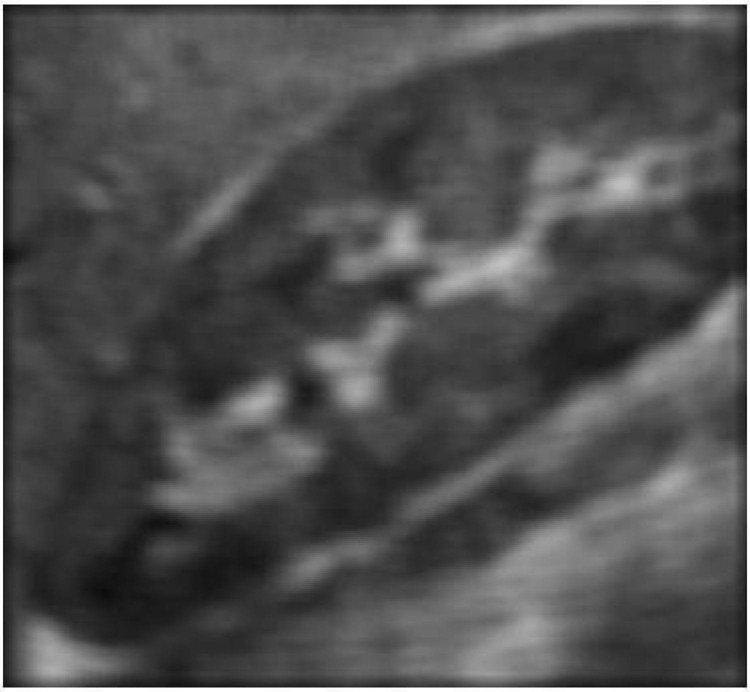
Kidney image after applying smoothing filter with 3x3 window



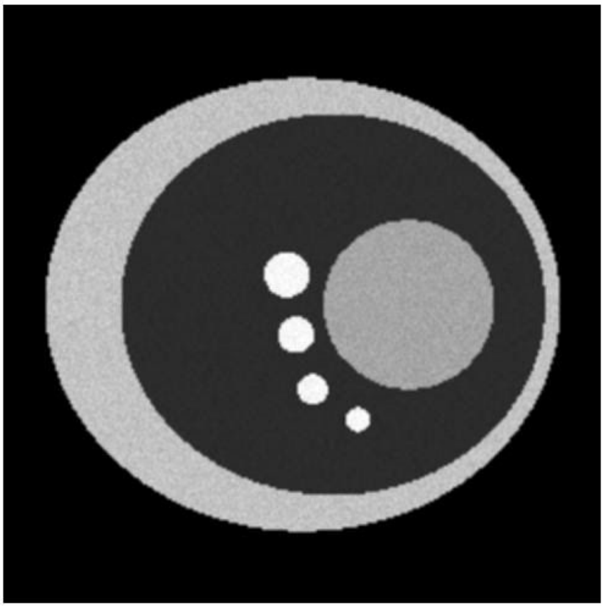
Kidney image after applying smoothing filter with 5x5 window



Kidney image after applying smoothing filter with 9x9 window



Kidney image after applying smoothing filter with 13x13 window



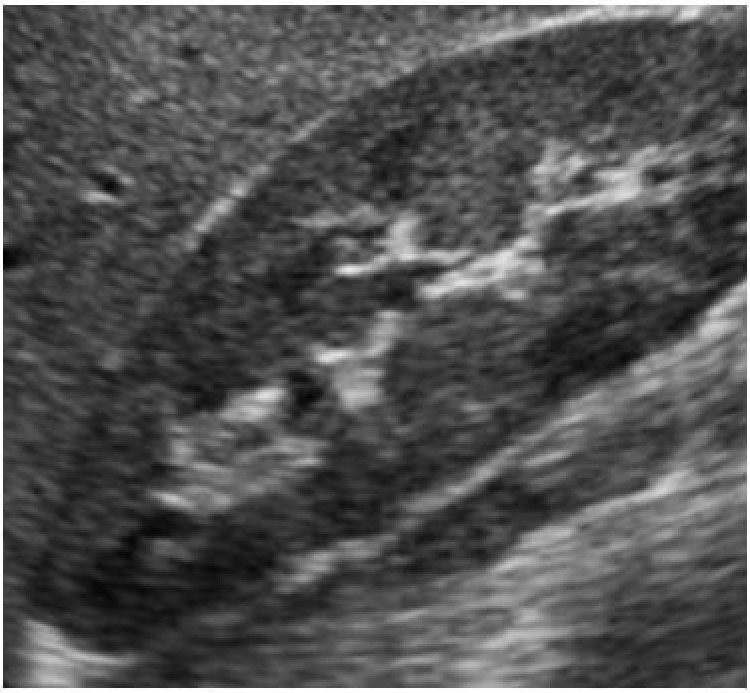
Synthetic image after applying smoothing filter with 3x3 window

Synthetic image after applying smoothing filter with 5x5 window

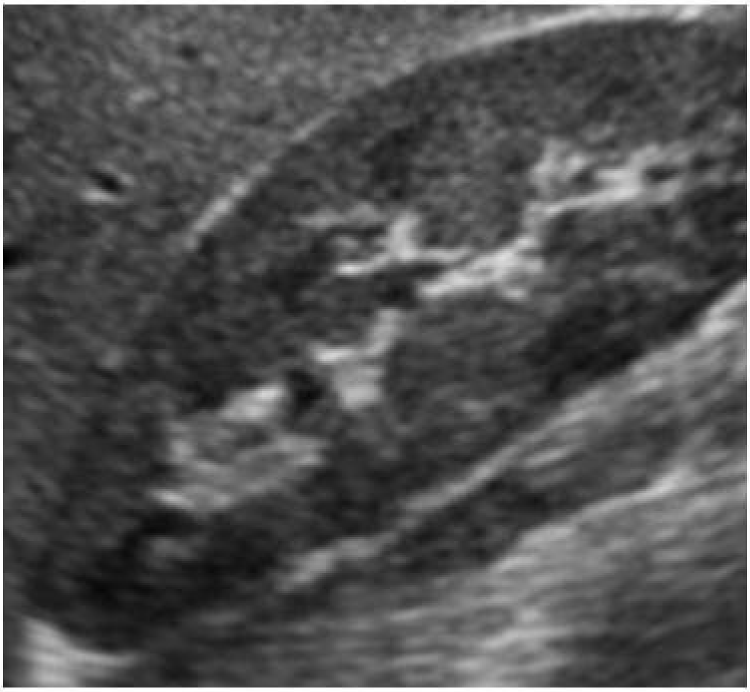
Synthetic image after applying smoothing filter with 9x9 window

Synthetic image after applying smoothing filter with 13x13 window

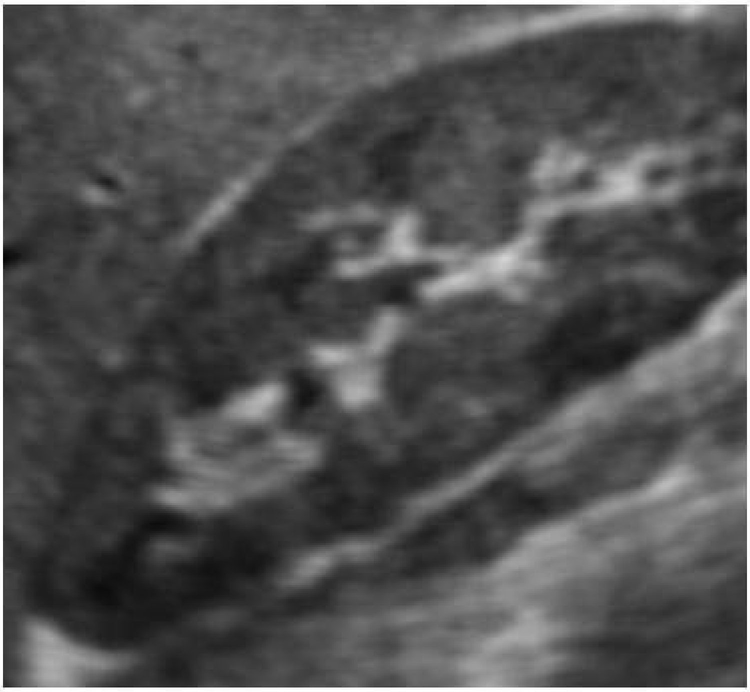
3) The main difference between these two filters is that : while mean filter only uses the local mean of the image in the window applied; Lee filter makes also use of the local variance obtained in the window applied and of the background noise variance in the image.Taken this into account we can say that Lee filter is more reliable in Image Filtering, since it takes more measurements into consideration. We could add that Lee filters are the upgrade of the local mean filters.
4) We can observe the same as in Task 2. The bigger the window, the blurrer the image.
The pixel values are changed and so the different regions in the image are more difficult now to be distinguished. The reason behind it is that the sharp features are removed.



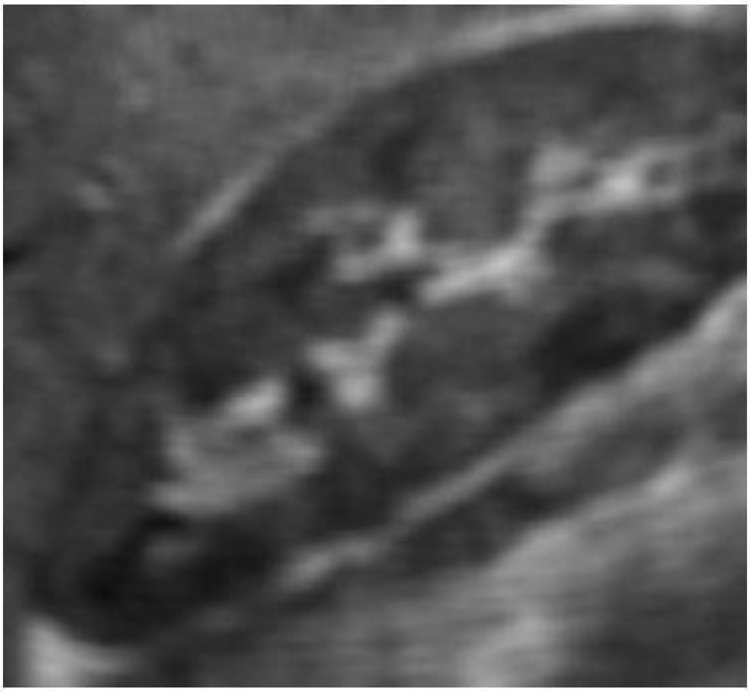
Kidney image after applying Lee filter with 3x3 window



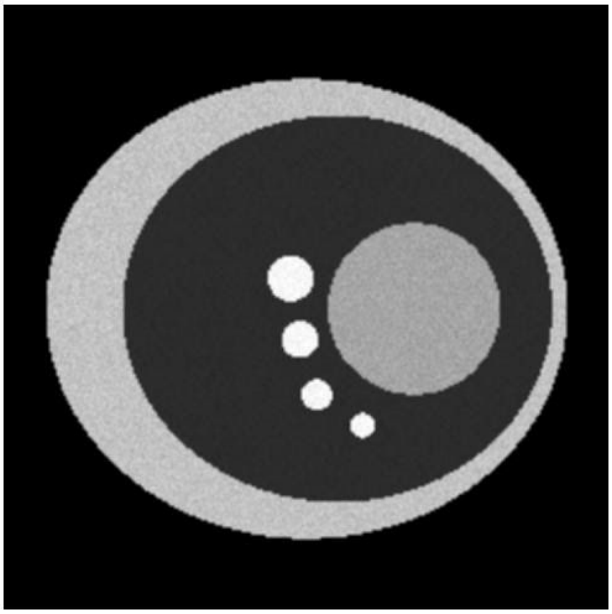
Kidney image after applying Lee filter with 5x5 window



Kidney image after applying Lee filter with 9x9 window



Kidney image after applying Lee filter with 13x13 window



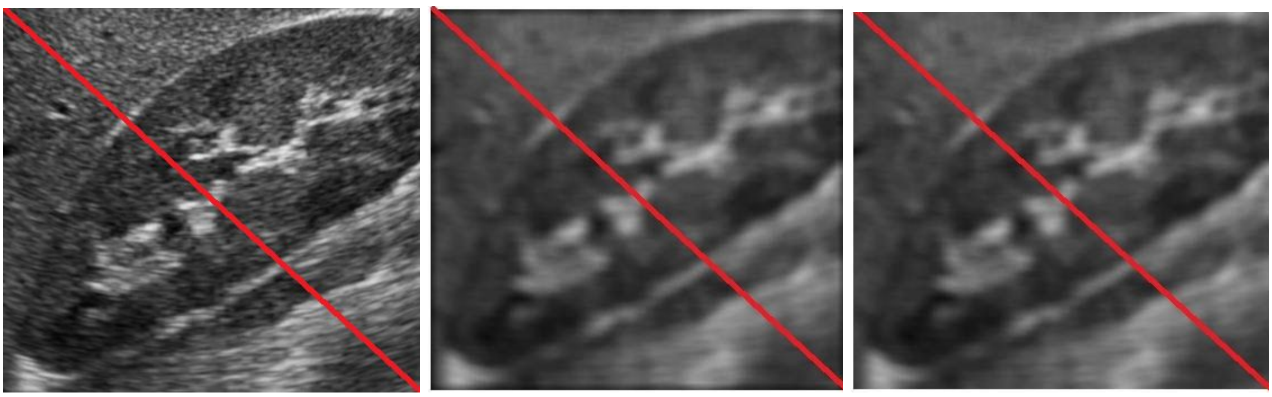
Synthetic image after applying Lee filter with 3x3 window

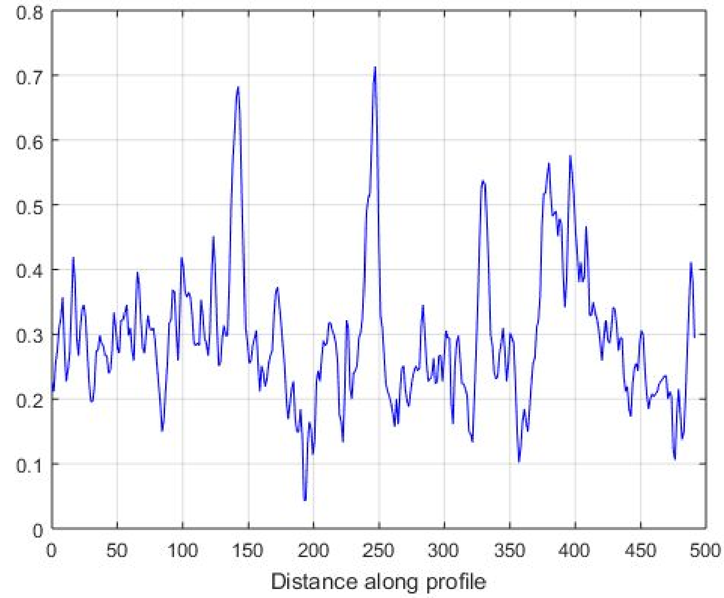
Synthetic image after applying Lee filter with 5x5 window

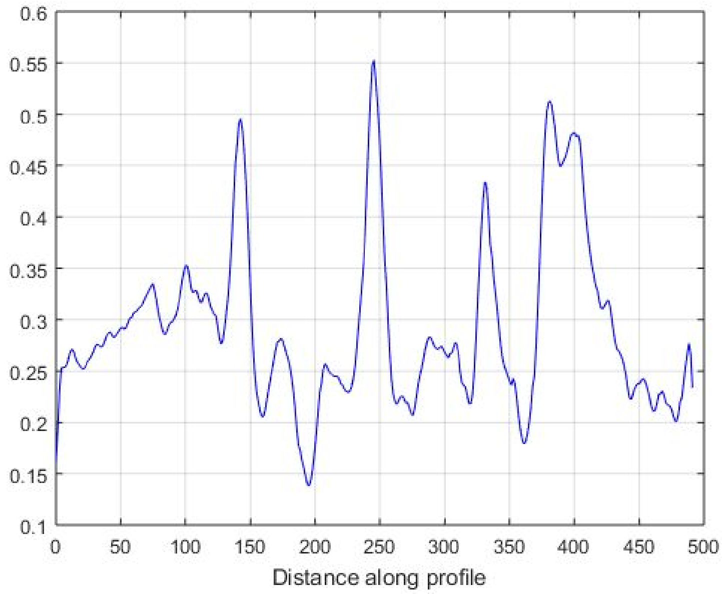
Synthetic image after applying Lee filter with 9x9 window

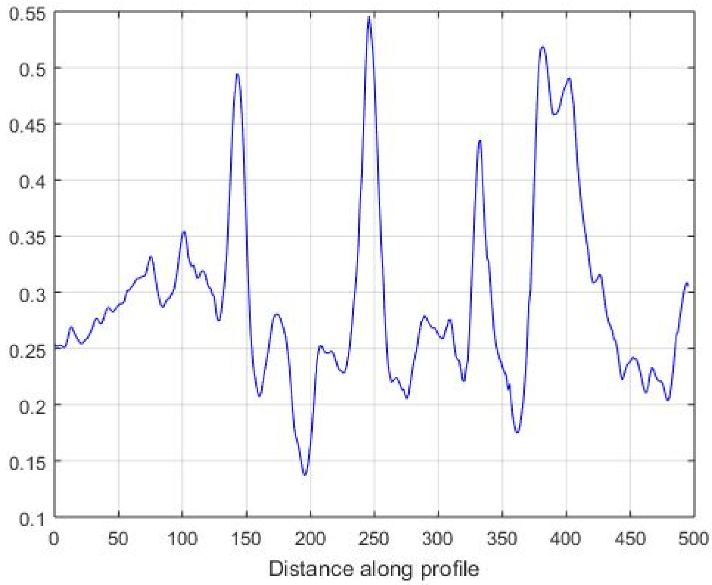
Synthetic image after applying Lee filter with 13x13 window

5) We will compare the results of filtering in the kidney image because it shows greater and more interesting differences since its regions are not clearly distinguishable. The window we will compare is the 13x13 because it allows us to notice the most notable changes of each method we applied. We draw a line diagonally from the top left corner to the bottom right corner and we do that three times in order to plot the image line profiles of the original image and the image filtered by local mean and Lee filter. Below we can see the procedure that we followed and their respective image line profiles. Starting from the original image and finishing with the Lee filtered image.

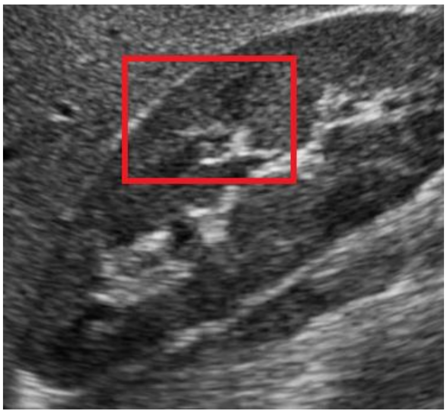



0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 50 100 150 200 250 300 350 400 450 500
Distance along profile

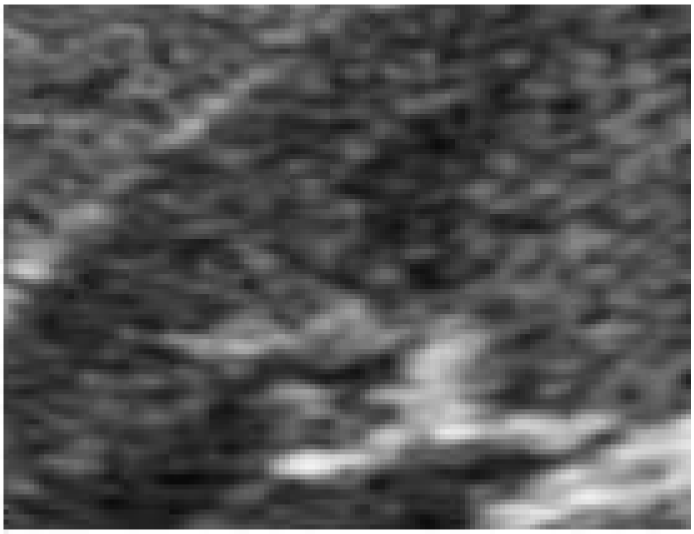

0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0 50 100 150 200 250 300 350 400 450 500
Distance along profile


0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0 50 100 150 200 250 300 350 400 450 500
Distance along profile

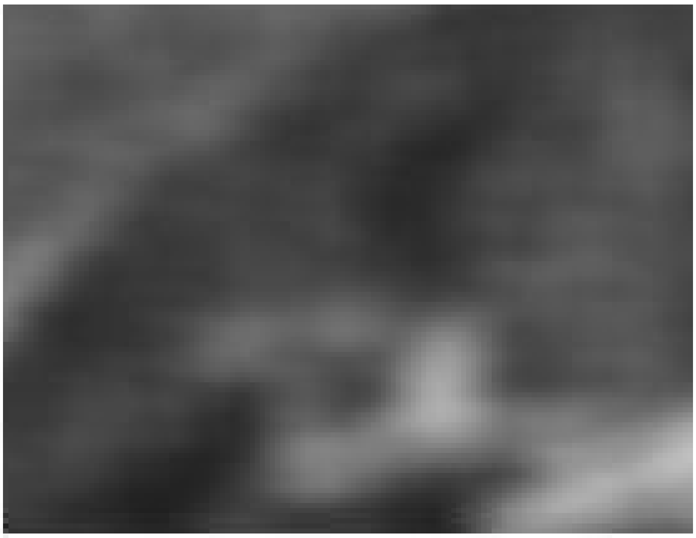
We will zoom in at this part:



So, we will have:



Zoomed-in original image



Zoomed-in mean filtered image

Zoomed-in Lee filtered image.

From the above analysis , we can conclude that the sharp features have been smoothened since there are more curves and less sharp edges at the image profile lines and also the filtered images have been blurred. Moreover, we notice that the filtered image profile lines follow the same form-shape and are almost identical. However, from the zoomed image we can deduce that the mean filtering has created a better result because the image is less blurred and as a result the regions can be defined more easily than in the Lee filtered image. In both images ,because of the 13x13 window ,it is not a simple task to distinguish the regions but three main colors appear that can define them. These are black and white on the foreground and grey as the background. As a last comment we can add that the mean filter has kept the colors of the original image more unaltered, while the Lee filter has added a gray tone to (-has grayed-) the image and that is the cause why it appears more blurred. As we can observe the contrast in our images is enhanced by smoothing the intensity variation. Therefore, the filtering does reduce the intensity variation of the images. It is more easily observed in the kidney image that the edges are enhanced - sharpened (because the synthetic image has a black background). The edges are becoming darker . Therefore, the edges are indeed affected and the image seems to be in a frame. And that becomes more and more obvious as the window gets bigger. The definition of the ideal boundary could be: When neighbouring pixel values differ more than a certain value (depends on each image) , different regions can be introduced. Additionally,

pixels with similar pixel value should be considered to belong to the same region. As it is mentioned above , the bigger the k, the smoother the image parts that had extreme pixel values. Therefore, the smoothing effect becomes more obvious as the k gets bigger , due to the fact that the pixel values that are attenuated are more and more closer to the average. As a conclusion, we should not abuse increasing the k-size of the window because at the end it becomes completely blurred. In the cases we examined , 5x5 window seems to be the optimal choice because it smoothes the image up to a point, without blurring it too much . Now , the CNR calculations of
(i) original images : CNR(kidney) = 3.0270
CNR(synthetic_image)= - 8.1973
(ii)local mean filtered images : CNR(kidney_13x13_mean)=1.5893
CNR(synthetic_image_3x3_mean)=-23.3949
(iii) Lee filtered images: CNR(kidney_13x13_Lee)=2.6166
CNR(synthetic_image_3x3_Lee)=-23.3949

The conclusions from the CNR calculations are:
Firstly, when CNR is positive, it decreases after filtering. On the contrary, when it is negative, its absolute value increases. Secondly, CNR of synthetic image after mean filtering equals the value of CNR after Lee filtering and the reason behind it might be that the regions of this image are clearly distinguishable. Thirdly, the closer to zero the CNR , the closer the averages of foreground and background are. We see that in the 13x13 window these averages are coming closer . But in the synthetic image the average of the background increases , since it gets even more negative. Lastly, we observe that CNR in synthetic image is negative while in kidney is positive. One possible explanation is that the background in synthetic image is larger and clearer than in kidney image.

D) Program Code

main.m

```
close all;
clear all;
clc;
kidney = imread('kidney.png');
kidney=im2double(kidney);
syn_img = imread('synthetic_image.png');
syn_img=im2double(syn_img);

imshow(kidney)
improfile
grid on;
imcrop( kidney , [112.5 39.5 136 105]);
h3 = 1/3*ones(3,1);
h5 = 1/5*ones(5,1);
h9 = 1/9*ones(9,1);
h13= 1/13*ones(13,1);
H = h3*h3';
H5= h5*h5';
H9= h9*h9';
H13=h13*h13';
% kidney be your image
imfilt1 = filter2(H,kidney);
imfilt2 =filter2(H,syn_img);
```

```
imfilt15 = filter2(H5,kidney);
imfilt25 =filter2(H5,syn_img);
imfilt19 = filter2(H9,kidney);
imfilt29 =filter2(H9,syn_img);
%im113= imfilter(kidney,H13);
imfilt113=filter2(H13,kidney);
imfilt213 =filter2(H13,syn_img);
figure ,imshow(imfilt1)
figure ,imshow(imfilt15)
figure ,imshow(imfilt19)
figure ,imshow(imfilt113)
improfile
grid on;
imcrop(imfilt113,[112.5 39.5 136 105]);
figure ,imshow(imfilt2)
figure ,imshow(imfilt25)
figure ,imshow(imfilt29)
figure ,imshow(imfilt213)

CNR1=CNR_fun(imfilt113);
CNR2=CNR_fun(imfilt2);




%lee task
imfilt1 = LeeFilter(kidney,3);
imfilt2 = LeeFilter(syn_img,3);
imfilt15 = LeeFilter(kidney,5);
imfilt25 = LeeFilter(syn_img,5);
imfilt19 = LeeFilter(kidney,9);
imfilt29 = LeeFilter(syn_img,9);
imfilt113 = LeeFilter(kidney,13);
imfilt213 = LeeFilter(syn_img,13);
figure ,imshow(imfilt1)
figure ,imshow(imfilt15)
figure ,imshow(imfilt19)
figure ,imshow(imfilt113)
improfile
grid on;
imcrop(imfilt113,[112.5 39.5 136 105]);
figure ,imshow(imfilt2)
figure ,imshow(imfilt25)
figure ,imshow(imfilt29)
figure ,imshow(imfilt213)

%task 5
CNR3=CNR_fun(kidney);
CNR4=CNR_fun(imfilt113);
CNR5=CNR_fun(syn_img);
CNR6=CNR_fun(imfilt2);
```

LeeFilter.m

```
function f=LeeFilter(I,win_size)
	OIm = double(I);
	window_size = win_size;
	means = imfilter(I, fspecial('average', window_size), 'replicate');
	sigmas = sqrt((I-means).^2/window_size^2);
	sigmas = imfilter(sigmas, fspecial('average', window_size), 'replicate');
	ENLs = (means./sigmas).^2;
	sx2s = ((ENLs.*(sigmas).^2) - means.^2)./(ENLs + 1);
	fbar = means + (sx2s.*(I-means)./(sx2s + (means.^2 ./ENLs)));
	OIm(means~=0) = fbar(means~=0);
	f=OIm;
end
```

CNR_fun.m

```
function f=CNR_fun(I)
	A = I;
	mask = zeros(size(A));
	mask(25:end-25,25:end-25) = 1;
	BW = activecontour(A,mask);
	numberOfPixels = numel(BW); %all pixels
	numberOfTruePixels = sum(BW(:)); %true pixels
	numberOfFalsePixels= numberOfPixels - numberOfTruePixels;
	meanTRUE=mean(BW(:));
	meanFALSE=mean(~BW(:));
	stdTRUE=std(BW(:));
	signal = meanTRUE- meanFALSE;
	noise = stdTRUE;
	f = 20*log10(signal/noise);
end
```