**Image Processing EEE412**

**Lab 2：lmage enhancement**

XUN XU 1926930

2019/10/24

Task1

I

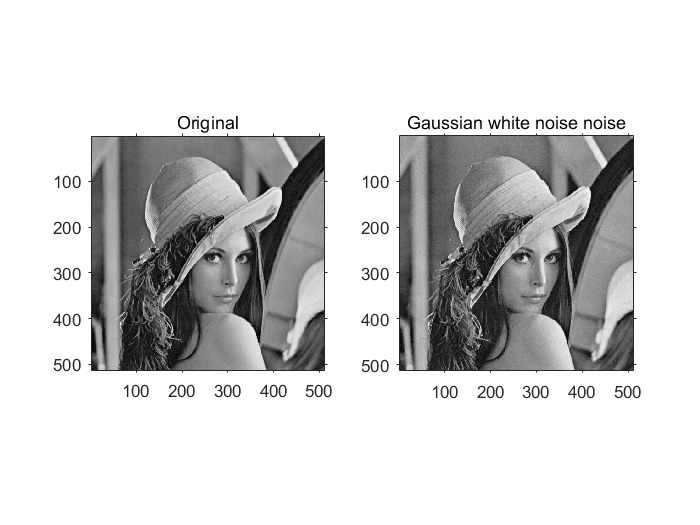
Using imread function to load into the image

Code:

im= imread('lenna512.bmp');

ii

result:

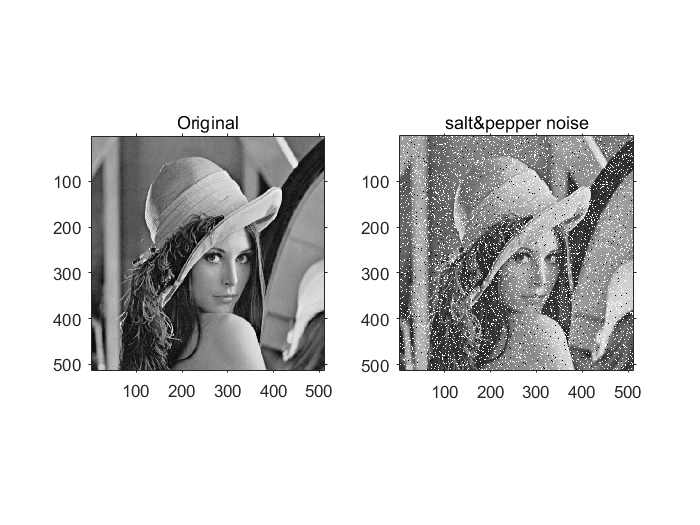


The calculated noise variance and mean are: 9.9762 and -0.0035

Code:

function [img\_noise] =im\_wn(img)  
  
img\_noise=double(img)+sqrt(10)\*randn(size(img));%using randn function to produce random number  
  
end

iii



code:

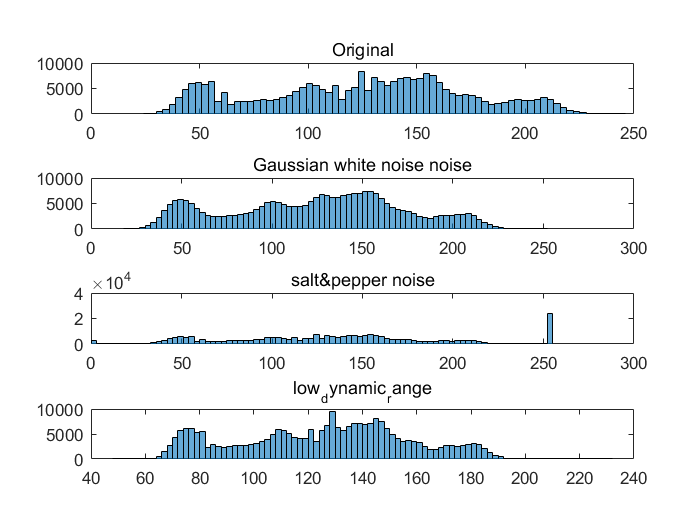
function [ image ] = im\_SP( image )  
  
[width,height,z]=size(image);  
if(z>1)  
 image=rgb2gray(image);%if it is not a grey image,then transform it  
end  
k1=0.1;%Set the ratio of the random number  
k2=0.1;  
a1=rand(height,width)<k1;%Generate a random number for the ordinate  
a2=rand(height,width)<k2;%Generate random numbers for the abscissa  
image(a1&a2)=0;  
image(a1& ~a2)=255;  
subplot(1,2,2);  
end

iv

|  |  |  |  |
| --- | --- | --- | --- |
| Image | im\_wn | im\_SP | im\_low\_dynamic \_range |
| PSNR (dB) | 38.1068 | 15.2281 | 23.9912 |

By comparing the psnr values, we can know that adding Gaussian white noise, low dynamic range, and adding three pictures of salt and pepper noise decreases in comparison with the original picture, which is consistent with our visual observation, in which Gaussian white noise is the most common and basic noise, and the salt and pepper noise is also called impulse noise. The appearance of black and white points has a greater impact on the image.

V



It can be seen that the histogram of the image with higher contrast is more extended.

vi

|  |  |  |  |
| --- | --- | --- | --- |
| Image | im\_wn10 | im\_wn100 | im\_wn1000 |
| PSNR (dB) | 47.7645 | 57.6143 | Inf |

Because the noise of the pictures we add is subject to the normal distribution of ~(0,), the noise of the image processed by this method will have a mean of 0, and as k increases , it will tend to 0. That is to say, the variance of the image noise tends to zero, and the image is closer and closer to the original image. Therefore, as k increases, the value of psnr becomes larger and larger, and the image is getting closer to the original image.

im= imread('lenna512.bmp');  
  
im\_sum=im\_wn(im);  
k=10;  
for i=2:k  
 im\_k=im\_wn(im);  
 im\_sum=double(im\_sum)+im\_k;  
end  
im\_ave=im\_sum/k;  
im\_wn10=im\_ave;  
  
im\_sum=im\_wn(im);  
k=100;  
for i=2:k  
 im\_k=im\_wn(im);  
 im\_sum=double(im\_sum)+im\_k;  
end  
im\_ave=im\_sum/k;  
im\_wn100=im\_ave;  
  
  
im\_sum=im\_wn(im);  
k=1000;  
for i=2:k  
 im\_k=im\_wn(im);  
 im\_sum=double(im\_sum)+im\_k;  
end  
im\_ave=im\_sum/k;  
im\_wn1000=im\_ave;  
  
  
disp(psnr(uint8(im\_wn10),im));  
disp(psnr(uint8(im\_wn100),im));  
disp(psnr(uint8(im\_wn1000),im));  
  
  
figure;  
subplot(141);  
imshow(im,[]);  
title('Original');  
axis on;  
subplot(142);  
imshow(im\_wn10,[]);  
title('imwn10');  
axis on;  
subplot(143);  
imshow(im\_wn100,[]);  
title('imwn100');  
axis on;  
subplot(144);  
imshow(im\_wn1000,[]);  
title('imwn1000');  
axis on;

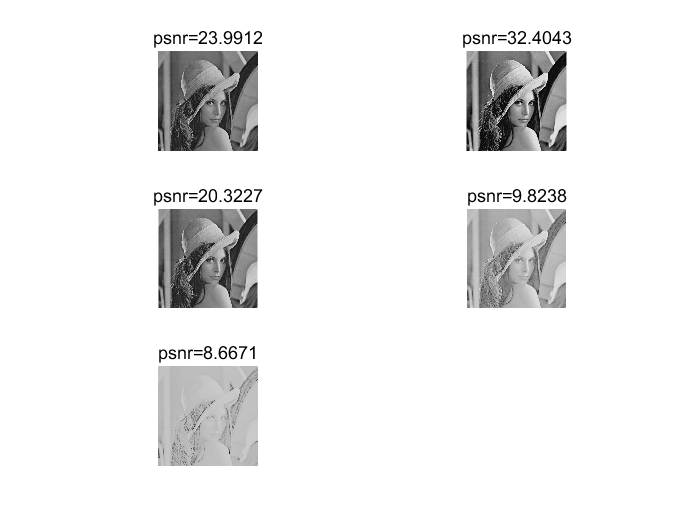
Task2

(1)

(2)

im=imread('lenna512\_low\_dynamic\_range.bmp');  
im1=imread('lenna512.bmp');  
  
figure;  
out1=linear\_mapping\_transform(im,0.05,0.05,0.95,0.95);  
subplot(3,2,1);  
imshow(out1,[]);  
disp(psnr(out1,im2double(im1)));  
title(['psnr=',num2str(psnr(out1,im2double(im1)))]);  
  
out2=linear\_mapping\_transform(im,0.3,0.15,0.7,0.83);  
subplot(3,2,2);  
imshow(out2,[]);  
disp(psnr(out2,im2double(im1)));  
title(['psnr=',num2str(psnr(out2,im2double(im1)))]);  
  
out3=linear\_mapping\_transform(im,0.1,0.15,0.85,0.95);  
subplot(3,2,3);  
imshow(out3,[]);  
disp(psnr(out3,im2double(im1)));  
title(['psnr=',num2str(psnr(out3,im2double(im1)))]);  
  
out4=linear\_mapping\_transform(im,0.3,0.7,0.4,0.75);  
subplot(3,2,4);  
imshow(out4,[]);  
disp(psnr(out4,im2double(im1)));  
title(['psnr=',num2str(psnr(out4,im2double(im1)))]);  
  
out5=linear\_mapping\_transform(im,0.3,0.8,0.8,0.85);  
subplot(3,2,5);  
imshow(out5,[]);  
disp(psnr(out5,im2double(im1)));  
title(['psnr=',num2str(psnr(out5,im2double(im1)))]);  
  
  
figure;  
histeq(im);  
subplot(1,2,1);  
imshow(out2);  
title(['psnr=',num2str(psnr(out2,im2double(im1)))]);  
subplot(1,2,2);  
imshow(b);  
title(['psnr=',num2str(psnr(histeq(im),im1))]);

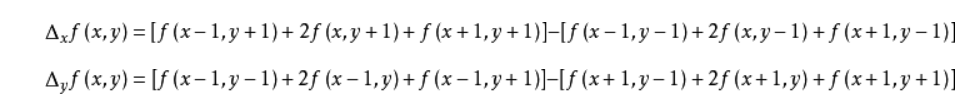
23.9912  
  
 32.4043  
  
 20.3227  
  
 9.8238  
  
 8.6671



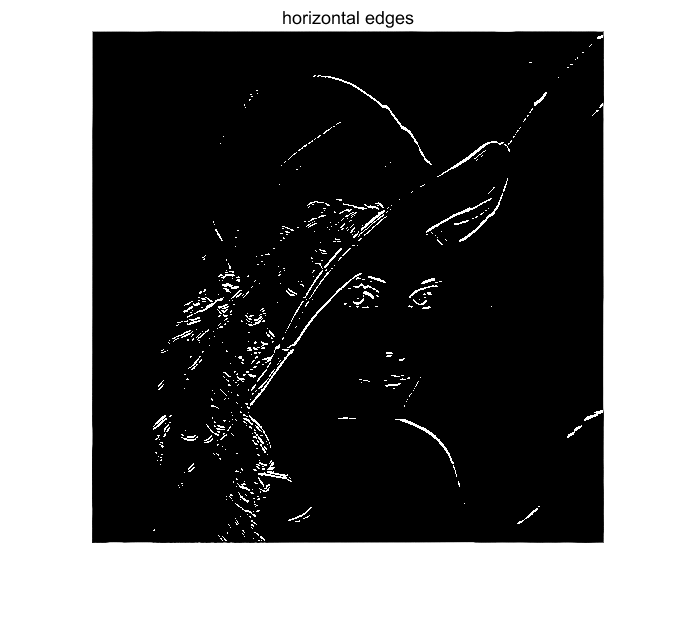


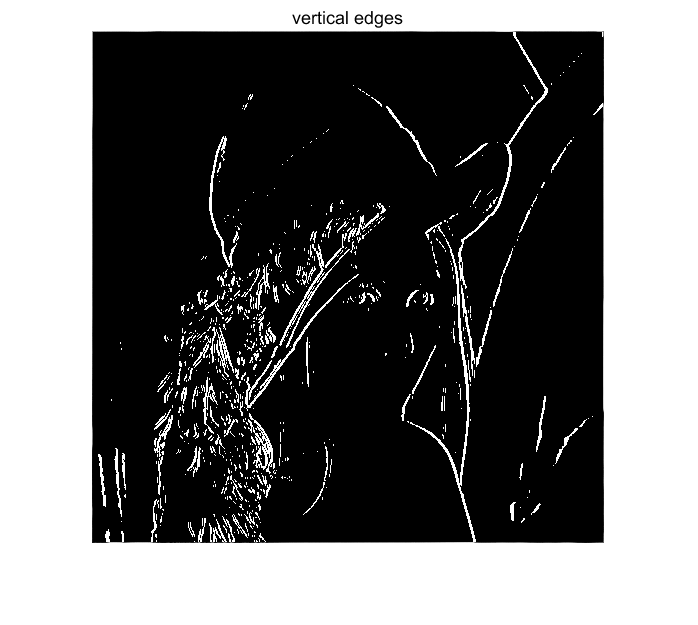
提高了对比度，但是psnr值并没有提升很大，最好线性变换图片虽然更接近原图，但是却没有直方图均衡化后的图片对比度高，

Task3



clear;  
im=imread('lenna512.bmp');  
grayimage=mat2gray(im);  
[m,n]=size(grayimage);  
  
newgrayimage=grayimage;  
sobelNum=0;  
sobelThreshold=0.7;  
for j=2:m-1  
 for k=2:n-1  
 sobelNum=abs(grayimage(j-1,k-1)+2\*grayimage(j-1,k)+grayimage(j-1,k+1)-grayimage(j+1,k-1)-2\*grayimage(j+1,k)-grayimage(j+1,k+1));  
 if(sobelNum > sobelThreshold)  
 newgrayimage(j,k)=255;  
 else  
 newgrayimage(j,k)=0;  
 end  
 end  
end  
figure;  
imshow(newgrayimage);  
title('horizontal edges');  
  
newgrayimage=grayimage;  
sobelNum=0;  
sobelThreshold=0.7;  
for j=2:m-1  
 for k=2:n-1  
 sobelNum=abs(grayimage(j-1,k+1)+2\*grayimage(j,k+1)+grayimage(j+1,k+1)-grayimage(j-1,k-1)-2\*grayimage(j,k-1)-grayimage(j+1,k-1));  
 if(sobelNum > sobelThreshold)  
 newgrayimage(j,k)=255;  
 else  
 newgrayimage(j,k)=0;  
 end  
 end  
end  
figure;  
imshow(newgrayimage);  
title('vertical edges');  
  
newgrayimage=grayimage;  
sobelNum=0;  
sobelThreshold=0.7;  
for j=2:m-1  
 for k=2:n-1  
 sobelNum=abs(grayimage(j-1,k+1)+2\*grayimage(j,k+1)+grayimage(j+1,k+1)-grayimage(j-1,k-1)-2\*grayimage(j,k-1)-grayimage(j+1,k-1))+abs(grayimage(j-1,k-1)+2\*grayimage(j-1,k)+grayimage(j-1,k+1)-grayimage(j+1,k-1)-2\*grayimage(j+1,k)-grayimage(j+1,k+1));  
 if(sobelNum > sobelThreshold)  
 newgrayimage(j,k)=255;  
 else  
 newgrayimage(j,k)=0;  
 end  
 end  
end  
figure;  
imshow(newgrayimage);  
title('all edges ');







Comparing the three figures, it can be seen that the sobel operator is better for vertical edge recognition.

Task4

comment on how effectively the noise is reduced while sharp edges and features in the image are preserved

3×3

sharp edges and features in the image are preserved well 但是有少许噪点没有消除

5×5 噪点基本消除，但是sharp edges and features in the image are preserved 差

图像下降，对比度下降

*As you experimented with the mean and median algorithms what*

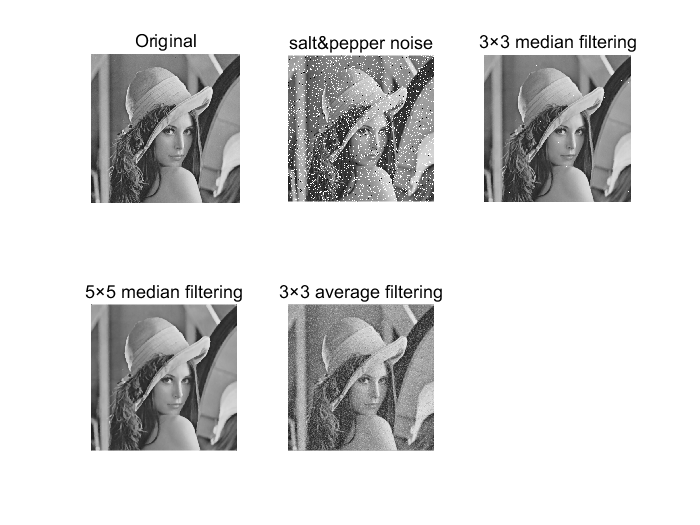
*different “performance” did you notice? Was the average or median filter better, and*

*why?*

中值滤波和均值滤波的区别，谁更好，为什么搜索

im=imread('lenna512.bmp');  
J=im\_SP(im);  
subplot(231),imshow(im);title('Original');  
subplot(232),imshow(J);title('salt&pepper noise')  
k1=medfilt2(J); %3×3 template median filtering  
k2=medfilt2(J,[5 5]); %5×5 template median filtering  
h1=fspecial('average',3);  
k3=imfilter(J,h1);% 3×3 template average filtering  
  
subplot(233),imshow(k1);title('3×3 median filtering')  
subplot(234),imshow(k2);title('5×5 median filtering')  
subplot(235),imshow(k3);title('3×3 average filtering')  
  
a=psnr(k1,im);  
disp(a);  
b=psnr(k2,im);  
disp(b);  
c=psnr(k3,im);  
disp(c);

31.8377  
  
 30.4971  
  
 22.3539



[*Published with MATLAB? R2016a*](http://www.mathworks.com/products/matlab)