**Digital Assignment 2**

**Name :** Adarsh Dubey

**Reg. No.:** 16BCE1336

CSE4019 - Image Processing

Write Matlab code to perform the following:-

1. Add different types of noises in an original image and restore the images. Try for all six noise PDFs discussed in the class.
2. Obtain a noisy image and try to find a suitable filter to restore it. Use SNR value to compare your restored images.

*Document should be very clear. Clear explanations should be given for the results.*

Solution:

In the following instances, we use cameraman.tif image available in matlab. The six types of probability distribution for the noise are applied to the cameraman.tif and some filters are applied on the noised image to check the performance of those filters on various types of the noises. The SNR value is calculated for the best result and displayed. The SNR for Impulse noise is found by inverse of NSR.

Matlab Implementation:

1. Gaussian Noise

img=imread('cameraman.tif');

% gaussian noise

1. sig=sqrt(255);

m=6;

gausNoise=(sig\*randn(size(img)))+m;

noiseImage=double(img) + gausNoise;

figure,subplot(1,2,1),imshow(img);subplot(1,2,2),imshow(noiseImage/256);

%creating gaussian filter

sigma=3;

gFilt=fspecial('gaussian',[10 10],sigma);

%creating mean filter

aFilt=fspecial('average',[3 3]);

%cleaning median filter

mCleanImg=medfilt2(noiseImage);

figure,imshow(mCleanImg/256);

1. %cleaning image

cleanImg=conv2(noiseImage,gFilt,'same');

figure,imshow(cleanImg/256);

aCleanImg=conv2(noiseImage,aFilt,"same");

figure,imshow(aCleanImg/256);

%snr calculation

snr=mean(mean(aCleanImg./gausNoise));

disp(snr)



Fig. original image and image with Gaussian noise



Fig. cleaning attempt using median filter



Fig. cleaning using a Gaussian filter



Fig. cleaning using mean filter



Fig. calculated snr value for mean filter cleaned image

b. Rayleigh Noise

img=imread('cameraman.tif');

%rayleigh noise

noise=raylrnd(double(img));

noiseImage=double(img)+noise;

figure,

subplot(1,2,1),imshow(img);

subplot(1,2,2),imshow(noiseImage/256);

%filters to clean noisy image

%arith mean filter

aImg=imfilter(noiseImage,fspecial('average',[3 3]));

%harmonic mean filter

hImg=3\*3./imfilter(1./(noiseImage+eps),ones(3,3),'replicate');

%median filter

mImg=medfilt2(noiseImage);

%midpoint filter

mpImg=0.5\* (ordfilt2(noiseImage, 3\*3, ones(3, 3)) + ordfilt2(noiseImage, 1, ones(3, 3)));

figure,imshow(aImg/256);

figure,imshow(hImg/256);

figure,imshow(mImg/256);

figure,imshow(mpImg/256);

%snr for mImg and noise

snr=mean(mean(mpImg./noise));

disp(snr);



Fig. original and noise added image



Fig. result of arithmetic mean filter



Fig. result of harmonic mean filter



Fig. result of a median filter



Fig. result of a mid point filter



SNR value for the result of image obtained by using median filter

1. Gamma(Erlang) Noise

img=imread('cameraman.tif');

%erlang(gamma) noise

noiseImage=randg(double(img));

figure,

subplot(1,2,1),imshow(img);

subplot(1,2,2),imshow(noiseImage/256);

%filters to clean noisy image

%arith mean filter

aImg=imfilter(noiseImage,fspecial('average',[3 3]));

%harmonic mean filter

hImg=3\*3./imfilter(1./(noiseImage+eps),ones(3,3),'replicate');

%median filter

mImg=medfilt2(noiseImage);

%midpoint filter

mpImg=0.5\* (ordfilt2(noiseImage, 3\*3, ones(3, 3)) + ordfilt2(noiseImage, 1, ones(3, 3)));

figure,imshow(aImg/256);

figure,imshow(hImg/256);

figure,imshow(mImg/256);

figure,imshow(mpImg/256);

%snr for mImg and noise

snr=mean(mean(mImg./(noiseImage-double(img))));

disp(snr);



Fig. original and noisy image



Fig. result of using a mean filter



Fig. result of using a harmonic filter



Fig. result of using a median filter



Fig. result of using a mid-point filter



SNR value for result obtained using median filter

1. Exponential Noise

img=imread('cameraman.tif');

%exponential noise

noise=randraw('exp',1,size(img));

noiseImage=double(img)+noise;

figure,

subplot(1,3,1),imshow(img);

subplot(1,3,2),imshow(noise);

subplot(1,3,3),imshow(uint8(noiseImage));

%filters to clean noisy image

%arith mean filter

aImg=imfilter(noiseImage,fspecial('average',[3 3]));

%harmonic mean filter

hImg=3\*3./imfilter(1./(noiseImage+eps),ones(3,3),'replicate');

%median filter

mImg=medfilt2(noiseImage);

%midpoint filter

mpImg=0.5\* (ordfilt2(noiseImage, 3\*3, ones(3, 3)) + ordfilt2(noiseImage, 1, ones(3, 3)));

figure,imshow(aImg/256);

figure,imshow(hImg/256);

figure,imshow(mImg/256);

figure,imshow(mpImg/256);

%snr for mImg and noise

snr=mean(mean(mImg./noise));

disp(snr);

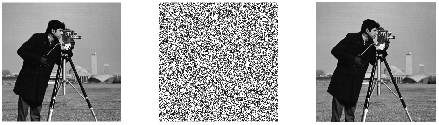


Fig. original image, exponential distribution noise, and noisy image



Fig. result of the mean filter



Fig. result of harmonic mean filter



Fig. result of the median filter



Fig. result of the mid-point filter



SNR value for result of the median filter

1. Uniform Noise

img=imread('cameraman.tif');

%uniform noise

noise=randraw('uniform',[0 150],size(img));

noiseImage=double(img)+noise;

figure,

subplot(1,3,1),imshow(img);

subplot(1,3,2),imshow(noise/256);

subplot(1,3,3),imshow(uint8(noiseImage));

%filters to clean noisy image

%arith mean filter

aImg=imfilter(noiseImage,fspecial('average',[3 3]));

%harmonic mean filter

hImg=3\*3./imfilter(1./(noiseImage+eps),ones(3,3),'replicate');

%median filter

mImg=medfilt2(noiseImage);

%midpoint filter

mpImg=0.5\* (ordfilt2(noiseImage, 3\*3, ones(3, 3)) + ordfilt2(noiseImage, 1, ones(3, 3)));

figure,imshow(aImg/256);

figure,imshow(hImg/256);

figure,imshow(mImg/256);

figure,imshow(mpImg/256);

%snr for mpImg and noise

snr=mean(mean(mpImg./noise));

disp(snr);

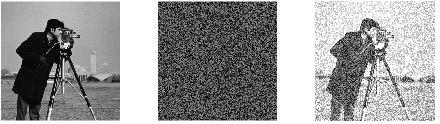


Fig. original image, noise, noise added image



Fig. result obtained using arithmetic mean filter



Fig. result obtained using harmonic mean filter



Fig. result obtained using median filter



Fig. result obtained using mid point filter



SNR value for the result obtained using mid point filter

1. Impulse (salt and pepper) Noise

img=imread('cameraman.tif');

%salt and pepper noise

[m n]=size(img);

noise=randi([0 1],m,n);

noiseImage= img+noise;

figure,

subplot(1,3,1),imshow(img);

subplot(1,3,2),imshow(noise);

subplot(1,3,3),imshow(uint8(noiseImage));

%filters to clean noisy image

%arith mean filter

aImg=imfilter(noiseImage,fspecial('average',[3 3]));

%harmonic mean filter

hImg=3\*3./imfilter(1./(noiseImage+eps),ones(3,3),'replicate');

%median filter

mImg=medfilt2(noiseImage);

1. %midpoint filter

mpImg=0.5\* (ordfilt2(noiseImage, 3\*3, ones(3, 3)) + ordfilt2(noiseImage, 1, ones(3, 3)));

figure,imshow(aImg);

figure,imshow(hImg);

figure,imshow(mImg);

figure,imshow(mpImg);

figure,imshow(m\_nImg);

%snr

%snr=1/rns since if we devide by noise, we get infinity because of 0

%devision

snr=1/(mean(mean(double(noise)./double(mImg))));

disp(snr);

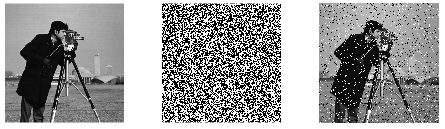


Fig. original image, noise, noisy image



Fig. result of the arithmetic mean filter

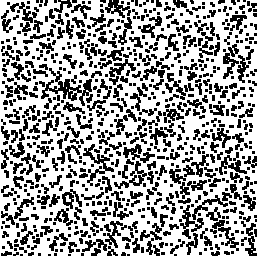


Fig. result of harmonic mean filter



Fig. result of median filter



Fig. result of mid-point filter

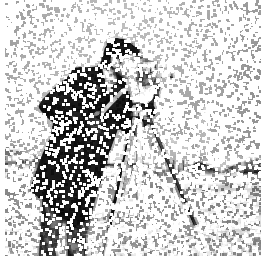


Fig. result of max filter

%min max filters can be implemented using imerode and imdilate functions with

%a structuring element of square of 3x3 size



SNR value for the result of median filter