Lecture 6

Fourth stage

Medical Physical Department

Medical Imaging Processing

Image Noise Removal

By

Asst. Prof. Dr. Mehdi Ebady Manaa

Noise

We may define noise to be any degradation in the image signal, caused by external disturbance. The errors will appear on the image output in different ways depending on the type of disturbance in the signal. Cleaning an image corrupted by noise is thus an important area of image restoration.

Types of noise:

There are four different noise types:

1-Salt and pepper noise:

Also called impulse noise, shot noise, or binary noise. This degradation can be caused by sharp, sudden disturbances in the image signal; its appearance is randomly scattered white or black (or both) pixels over the image. As shown in figure (1)

(a) Original image (b) With added salt & pepper noise

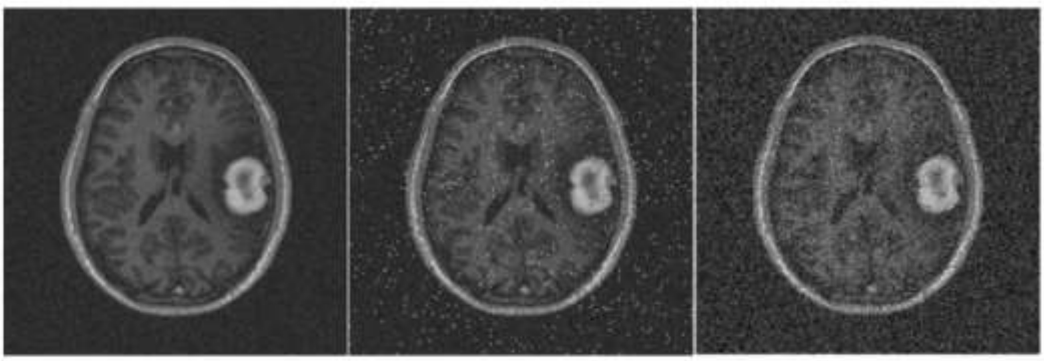


Figure (1) Noise on an image

2 -Gaussian noise:

Gaussian noise is an idealized form of white noise, which is caused by random fluctuations in the signal (white noise by watching a television) is normally distributed. we can model a noisy image by simply adding the image I and noise N: I+N. Assume that I is a matrix whose elements are the pixel values of our image, and N is a matrix whose elements are normally distributed.

3- Speckle Noise:

Whereas Gaussian noise can be modelled by random values added to an image; speckle noise (or more simply just speckle) can be modelled by random values multiplied by pixel values, hence it is also called multiplicative noise. Speckle noise is a major problem in some radar applications. As shown in figure (2)

(a) Gaussian noise (b) Speckle noise

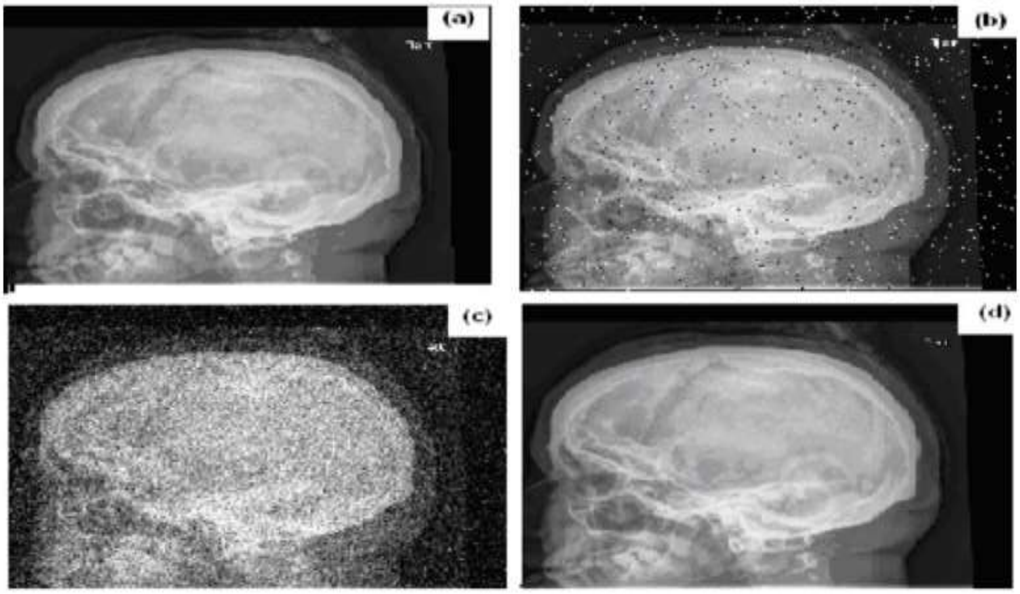


Figure (2): The twins image corrupted by Gaussian and speckle noise.

Although Gaussian noise and speckle noise appear superficially similar, they are formed by two totally different methods.

4- Periodic noise:

If the image signal is subject to a periodic, rather than a random disturbance, we might obtain an image corrupted by periodic noise. The effect is of bars over the image. As shown in figure (3) Salt and pepper noise, Gaussian noise and speckle noise can all be cleaned by using spatial filtering techniques. Periodic noise, however, requires image transforms for best results.

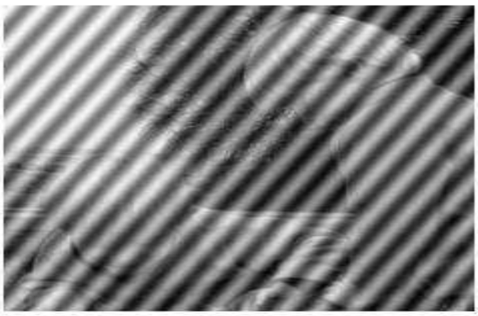


Figure (3): The twins image corrupted by periodic noise.

Cleaning salt and pepper noise:

1- **Low pass filtering:** Given that pixels corrupted by salt and pepper noise are high frequency components of an image, we should expect a low-pass filter should reduce them. As shown in figure (4).

2-Median filtering: Median filtering seems almost tailor-made for removal of salt and pepper noise. A median filter is an example of a non-linear spatial filter. The result is shown in figure (5). The result is a vast improvement on using averaging filters. For example.

50	65	52
63	255	58
61	60	57

$\longrightarrow$ 50 52 57 58 [60] 61 63 65 255 $\longrightarrow$ 60

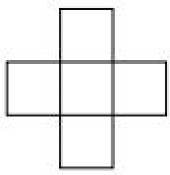
(a) 3 × 3 averaging (b) 7 × 7 averaging

Figure (4): Attempting to clean salt & pepper noise with average filtering.

Figure (5): Cleaning salt and pepper noise with a median filter.

3-Rank-order filter: Median filtering is a special case of a more general process called rank-order filtering. Rather than take the median of a set, we order the set and take the nth value, for some predetermined value of

n. Thus median filtering using a 3x3 mask is equivalent to rank-order filtering with n=5. Similarly, median filtering using a 5x5 mask is equivalent to rank-order filtering with n=13. There is only one reason for using rank-order filtering instead of median filtering, and that is that it allows us to choose the median of non-rectangular masks. For example, if we decided to use as a mask a 3x3 cross shape.



Example01 :

The 5x5 is a standard rank order filter which works in a 5 by 5 environment. When 25 pixel values encountered in the input map are:

2	4	10	3	21
14	1	119	12	9
112	7	5	118	13
8	8	132	15	6
113	2	5	10	12

Then the median filter orders these 25 values as:

1,2,2,3,4,5,5,6,7,8,8,8,9,9,10,10,11,12,12,12,13,13,14,15,21

The value **at rank order 13** is assigned to the pixel in the output map: 9.

Input pixel value 5 is smaller than the value of surrounding pixels in this example, while the output pixel value 9 is the median value of the 25 pixels examined.

Example 02:

The MED3x3 filter is a standard rank order filter which works in a 3 by 3 environment.

When 9 pixel values encountered in the input map are:

9	12	9
11	5	7
8	8	13

Then the median filter orders these 9 values as:

5, 7, 8, 8, 9, 9, 11, 12, 13

The value **at rank order 5** is assigned to the pixel in the output map: 9.

Input pixel value 5 is smaller than the value of surrounding pixels in this example, while the output pixel value 9 is the median value of the 9 pixels examined.

4-An outlier method:

Applying the median filter can in general be a slow operation: each pixel requires the sorting of at least nine values. To overcome this difficulty. We use of cleaning salt and pepper noise by treating noisy pixels as **outliers**. The following approach for noise cleaning:

1. Choose a threshold value D.
2. For a given pixel, compare its value p with the mean m of the values of its eight neighbors.
3. If |p-m|>D then classify the pixel as noisy, otherwise not.
4. If the pixel is noisy, replace its value with m; otherwise leave its value unchanged.

Cleaning Gaussian noise:

Average filtering: If the Gaussian noise has mean 0, then we would expect that an average filter would average the noise to 0. The larger the size of the filter mask, the closer to zero. Unfortunately, averaging tends to blur an image, as we have seen in chapter 3. However, if we are prepared to trade off blurring for noise reduction, then we can reduce noise significantly by this method. Suppose we take the and averaging filters, and apply them to the noisy image t. The results are shown in figure (6) The results are not really particularly pleasing; although there has been some noise reduction.

```
>> a3=fspecial('average');
>> a5=fspecial('average',[5,5]);
>> tg3=filter2(a3,t_ga);
>> tg5=filter2(a5,t_ga);
```

(a) 10 images (b) 100 images

Figure (6): Image averaging to remove Gaussian noise.

Exercises

1. The arrays below represent small greyscale images. Compute the 4 × 4 image that would result in each case if the middle 16 pixels were transformed using a 3 × 3 median filter:

8	17	4	10	15	12
10	12	15	7	3	10
15	10	50	5	3	12
4	8	11	4	1	8
16	7	4	3	0	7
16	24	19	3	20	10

1	1	2	5	3	1
3	20	5	6	4	6
4	6	4	20	2	2
4	3	3	5	1	5
6	5	20	2	20	2
6	3	1	4	1	2

7	8	11	12	13	9
8	14	0	9	7	10
11	23	10	14	1	8
14	7	11	8	9	11
13	13	18	10	7	12
9	11	14	12	13	10

2. Using the same images as in question 1, transform them by using a 3 × 3 averaging filter.