Image Processing | CS 452000

許木羽 / 111000177

Proj05-01

1. **Gaussian Noise**

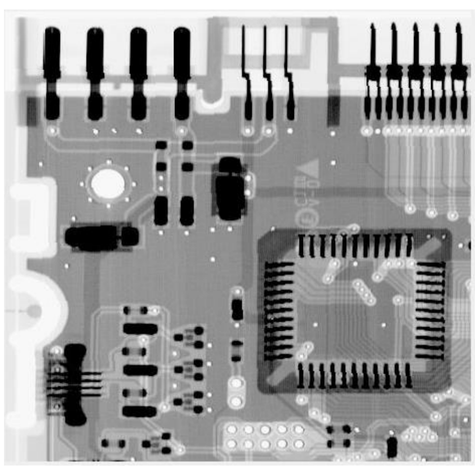
   We can get gaussian noise by formula as : $g(x) = f(x) + noise$

   The noise formula for gaussian noise as : $noise = \sigma * rand(0,1) + \mu$

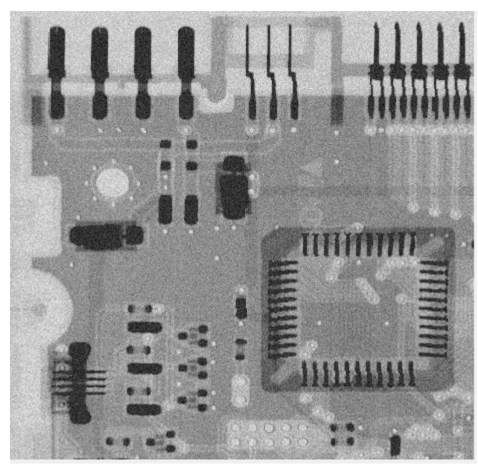
   Where the σ and μ are provided from the input.

2. **Salt and Pepper**

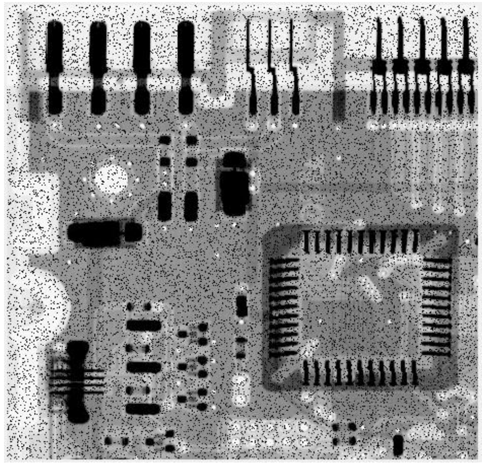
   We generate Random probability (p) for each element in array. We have probability Ps for salt, and Pp for pepper. To distribute the probability equally, for p < Ps become salt, and for 1-p < Pp becomes Pepper.



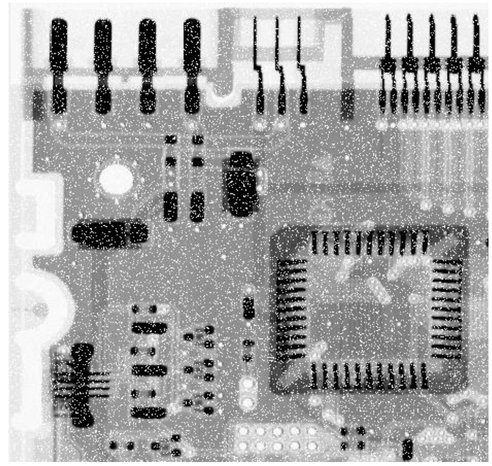
Original Image



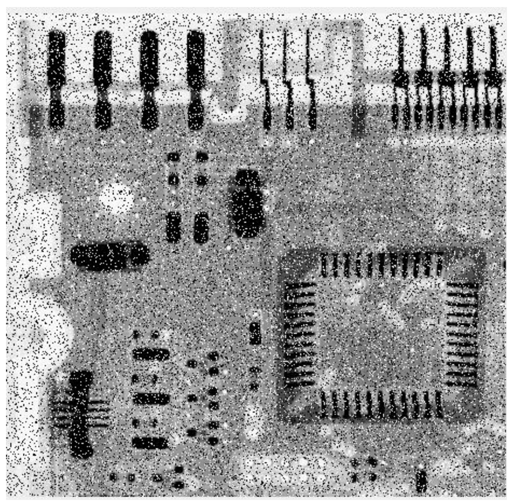
Gaussian Noise input $\sigma = 0$ and $\mu = 20$



Pepper = 0.1



Salt = 0.1



Pepper & Salt = 0.1

Proj05-02

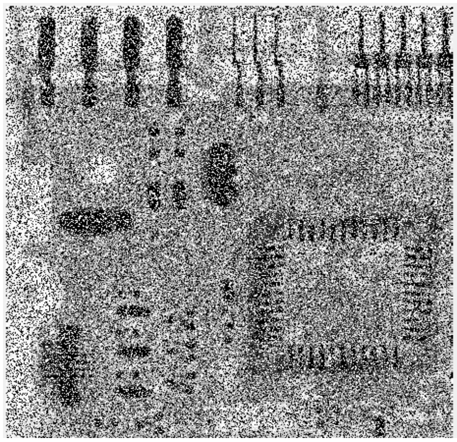
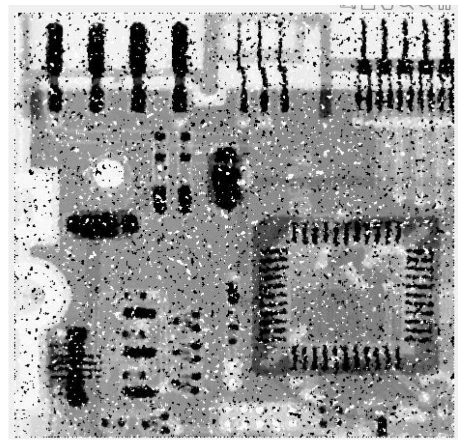
We can remove the noise by convolution of the median, as:

$$output(i,j) = median(input(i-1:i+1, j-1:j+1), ''all'')$$

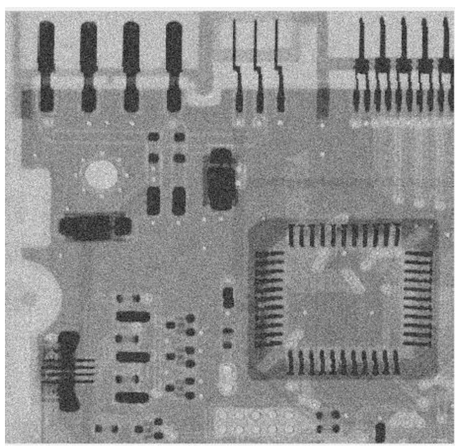
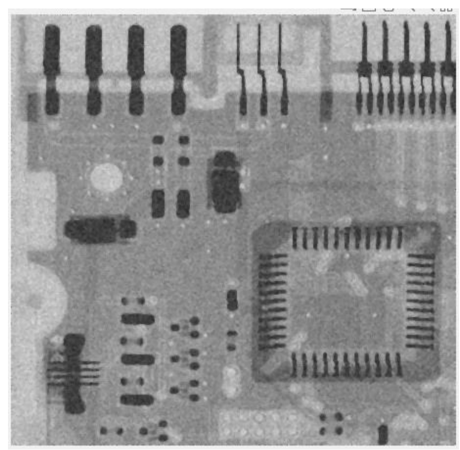
Next page is a representation of image that has noise (left), then filtered by median filter to restore (right)

Before (left) vs After (right) comparison:

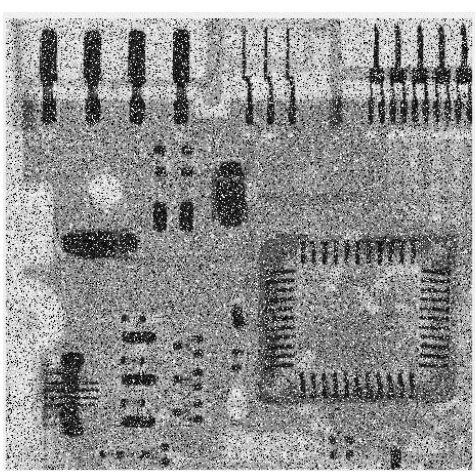
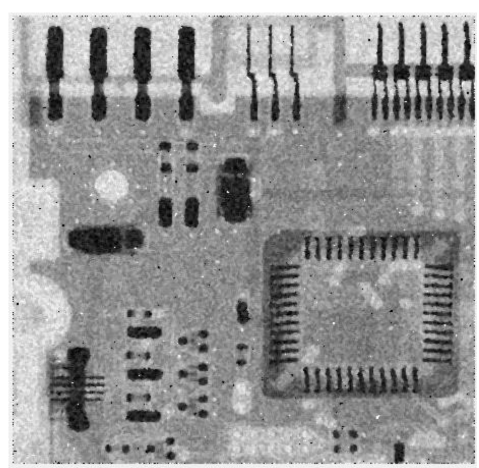
Scenario #1

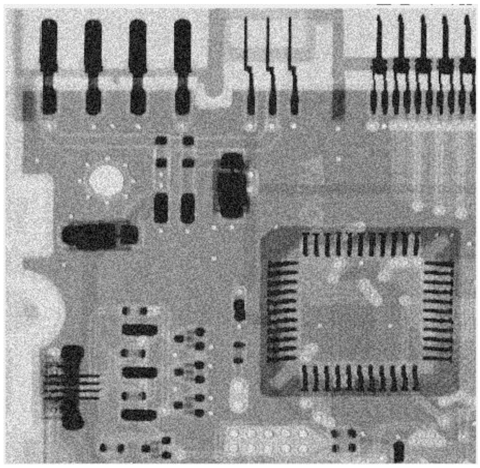
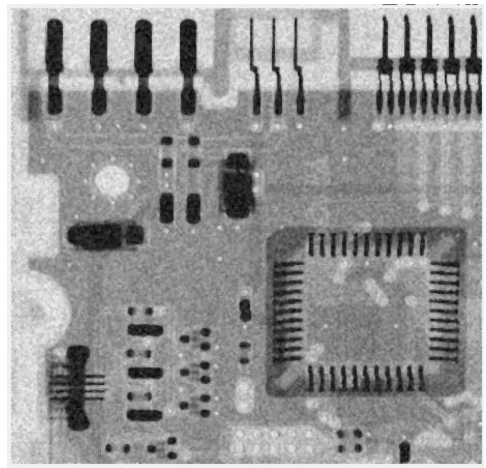
Scenario #2

Scenario #3

Scenario #4

Proj05-03

1. **Sin Noise**

   Add sin noise is using formula below (Example using A = 0.3, $u_o = v_o = 5$)

$$g(x,y) = f(x,y\,) + A * \sin\left(2\pi\left(\frac{u_o x}{M} + \frac{v_o y}{N}\right)\right)$$

$x = row\ index\ starts\ from\ 0$

$y = column\ index\ starts\ from\ 0$

$M = row$

$N = column$

2. **Notch Filtering**

   Computing by the formula given:

$$H(u,v) = \begin{cases} 1 & if \quad D_1(u,v) > D_0\ and\ D_2(u,v) > D_0 \\ 0 & \qquad\qquad otherwise \end{cases}$$

$$D_1(u,v) = \left(\left(u - \frac{M}{2} - u_o\right)^2 + \left(v - \frac{N}{2} - v_o\right)^2\right)^{\frac{1}{2}}$$

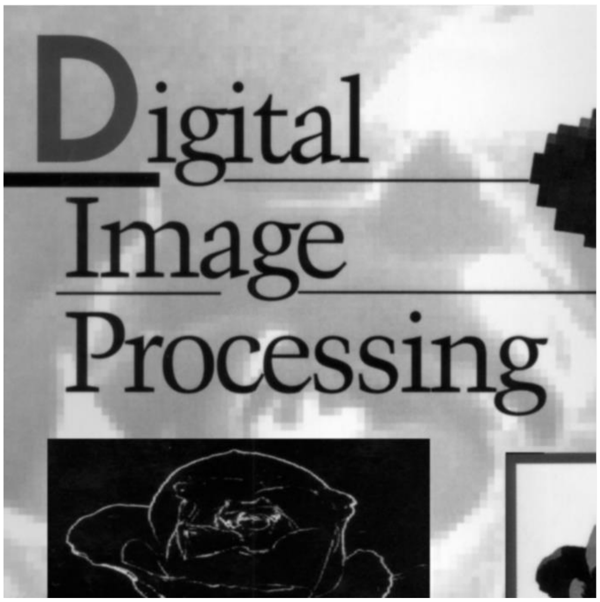
$$D_2(u,v) = \left(\left(u - \frac{M}{2} + u_o\right)^2 + \left(v - \frac{N}{2} + v_o\right)^2\right)^{\frac{1}{2}}$$

3. **PSNR**

Since the maximum is 1 for single type, we can compute the formula below directly:

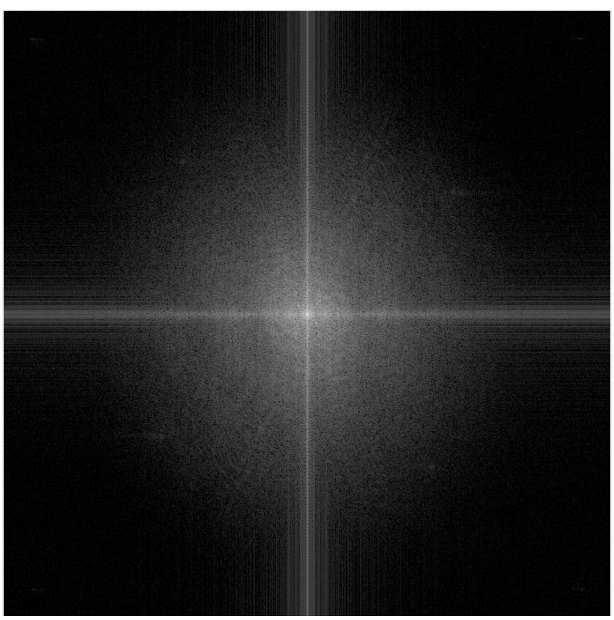
$$\boldsymbol{psnr = 10 * \log_{10}\left(\frac{1}{MSE}\right)}$$

Here is the image after following all 3 steps:

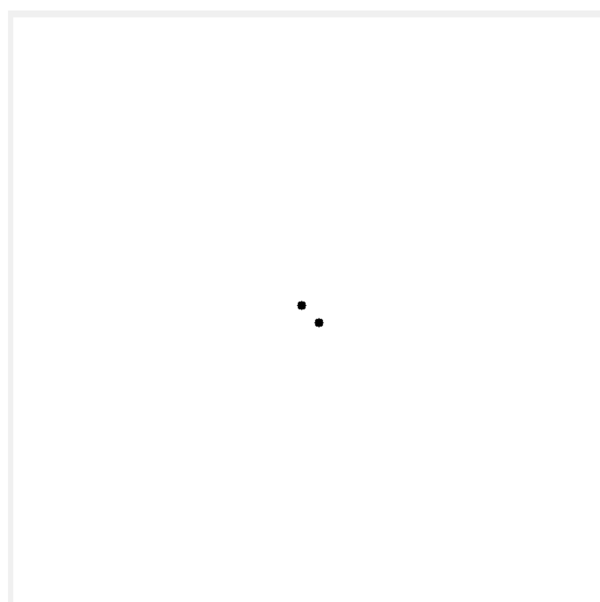


Original Image

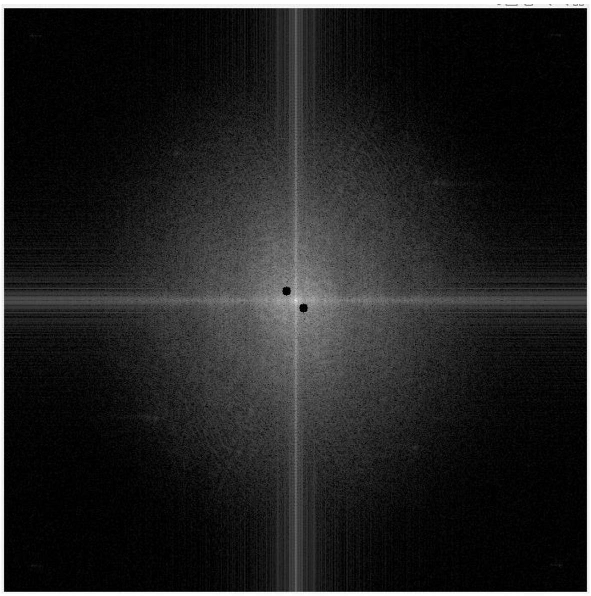
Add Sin Function



Frequency Domain (FFT)



Notch



After notch Frequency

Inverse FFT

PSNR = 23.3

Proj05-04

1. **Motion Blur Formula**

$$H(u,v) = \frac{T}{\pi\,(a*u+b*v)}\sin\big(\pi\,(a*u+b*v)\big)*e^{-j\pi\,(a*u+b*v)}$$

$$output_f(u,v) = input_f(u,v)*H(u,v)$$

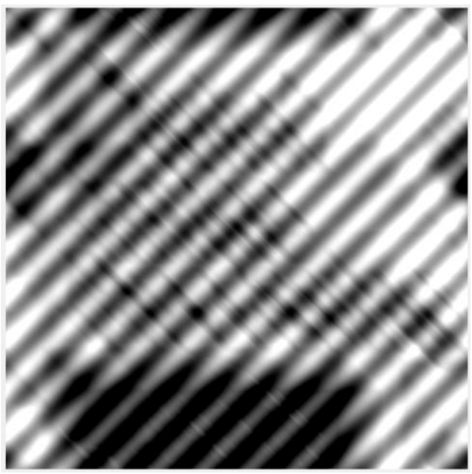
2. **Weiner Filtering**

$$W(u,v) = \frac{1}{H(u,v)}*\frac{abs\big(H(u,v)\big)^2}{abs\big(H(u,v)\big)^2+K}$$

$$output_f(u,v) = input_f(u,v)*W(u,v)$$

Original Image

Motion Blur



Sin Function

Weiner filter K = 0.1, psnr = 41

Weiner filter K = 0.01, psnr = 37.6

Weiner filter K = 0.0001, psnr =32.7