Image Enhancement

- Image Enhancement - Spatial Domain
  - Smoothing filter
  - Median filter
- Convolution
  - 1D Discrete
  - 1D Continuous
  - 2D Discrete
  - 2D Continuous
- Sharpening filter

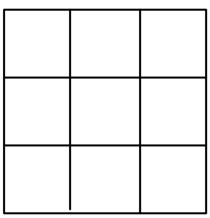
Salt & Pepper Noise

Neighborhood Averaging

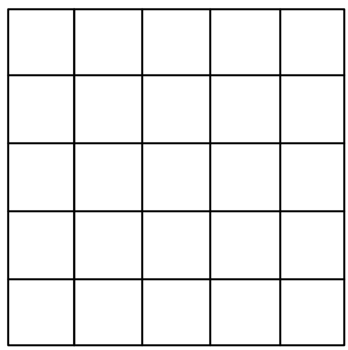
S = neighborhood of pixel (x,y)
M = number of pixels in neighborhood S

$$g(x,y) = (1/M) \sum_{(n,m) \in S} f(n,m)$$

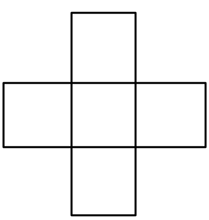
Neighborhoods:

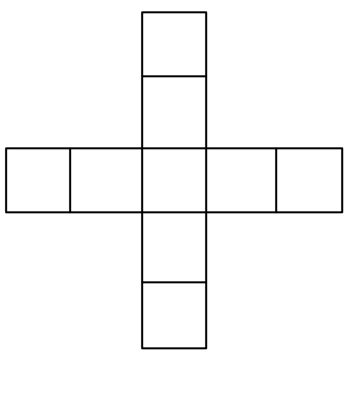


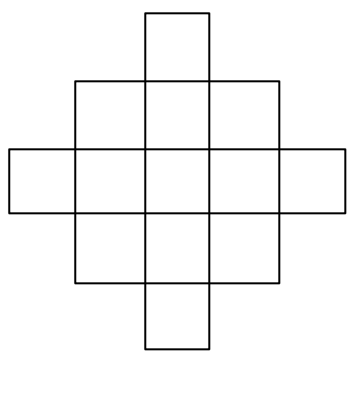
3 x 3



5 x 5

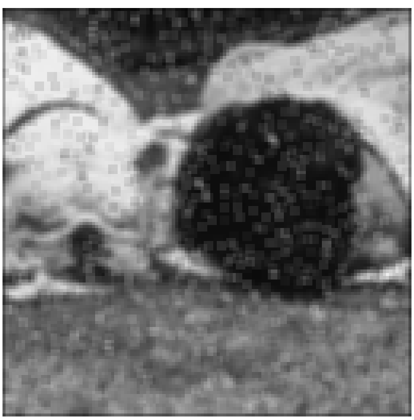




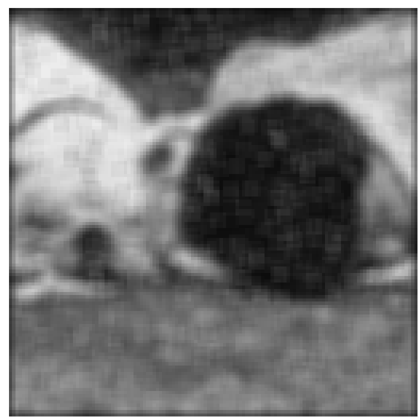


Neighborhood Averaging - Example

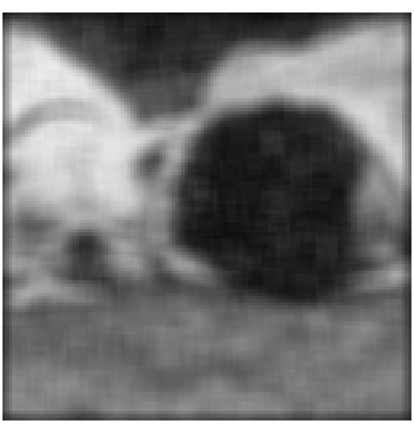
Salt & Pepper Noise



3 X 3 Average



5 X 5 Average



7 X 7 Average

Median

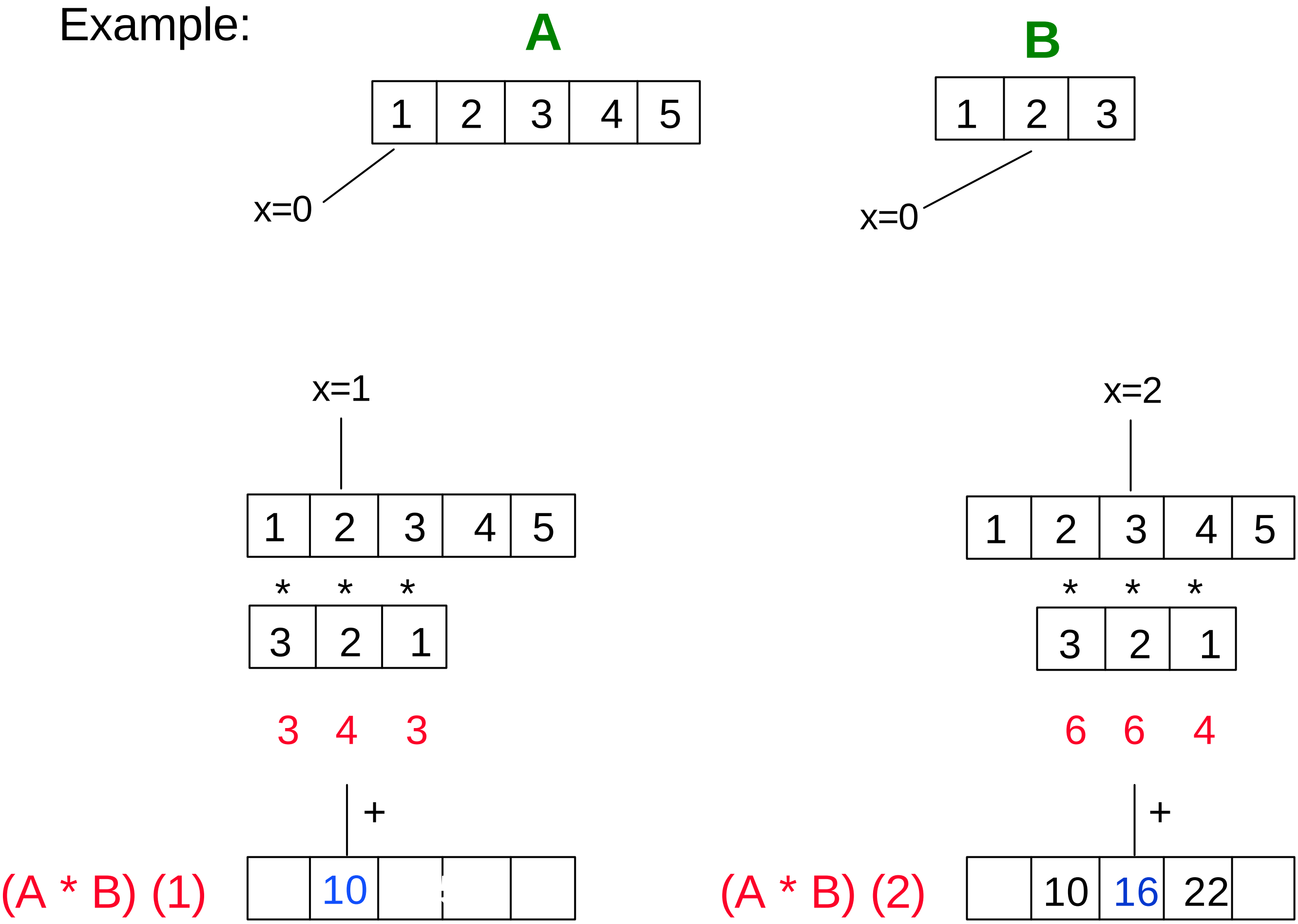
Convolution

A, B = images
B is typically smaller than A and is called the **mask**.

1 dimensional:

$$(A * B)(x) = \sum_i A(i)B(x-i)$$

Example:



What happens near the edges?

Convolution with ○

1	2	3

- Option 1: Zero padding

0 0 0

1	2	3	4	5

0 0 0

4	10	16	22	22

- Option 2: Wrap around

3 4 5

1	2	3	4	5

1 2 3 4 5

19	10	16	22	23

- Option 3: Reflection

3 2 1

1	2	3	4	5

5 4 3 2

7	10	16	22	27

Why one image is reflected in the convolution:

With reflection:

···· 1 2 3 ···· * ···· 0 1 0 ···· = ···· 0 3 2 1 0 ····

···· 0 1 0 ···· * ···· 1 2 3 ···· = ···· 0 3 2 1 0 ····

Without reflection:

···· 1 2 3 ···· * ···· 0 1 0 ···· = ···· 0 3 2 1 0 ····

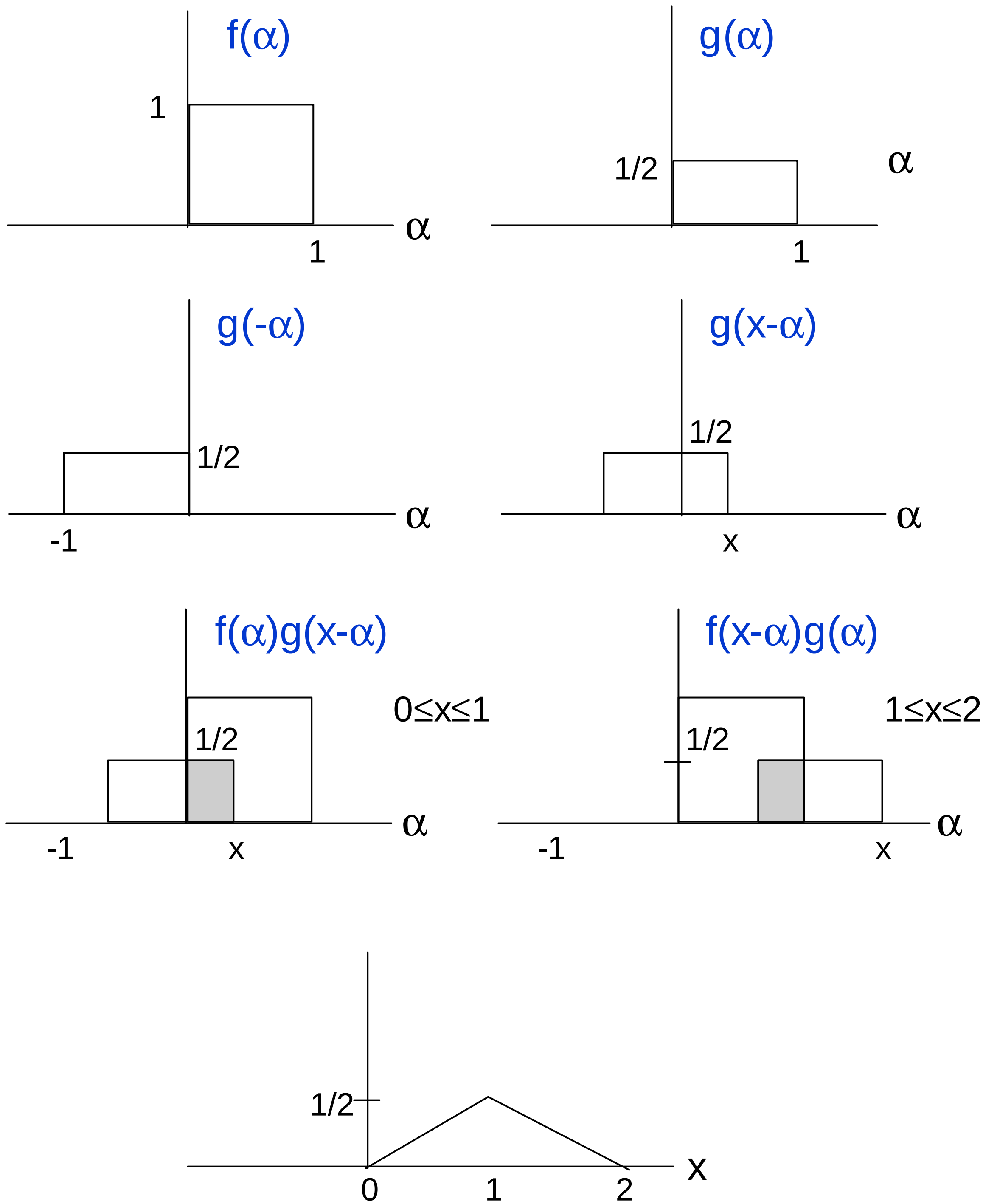
···· 0 1 0 ···· * ···· 1 2 3 ···· = ···· 0 1 2 3 0 ····

Reflection is needed so that convolution is commutative:

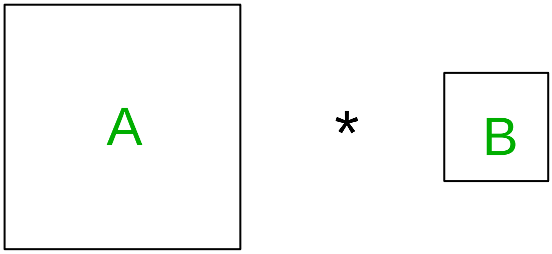
$A * B = B * A$

Convolution - Continuous Case

$$(f * g)(x) = \int_{-\infty}^{\infty} f(a)\, g(x-a)\, da$$



Convolution - 2 Dimensions



$$(A * B)(x,y) = \sum_{i} \sum_{j} A(i,j)\, B(x-i, y-j)$$

10	5	20	20	20
10	5	20	20	20
10	5	20	20	20
10	5	20	20	20
10	5	20	20	20

*

-1	1
0	1

=

-10	5	-15	0	0
-10	15	-10	20	20
-10	15	-10	20	20
-10	15	-10	20	20
-10	15	-10	20	20

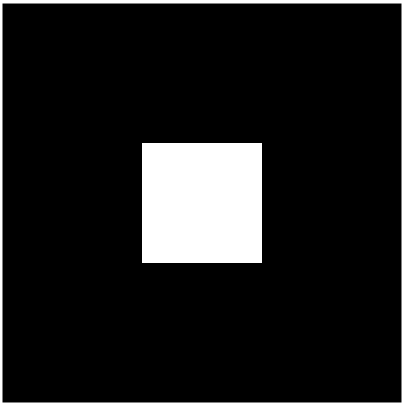
(zero padding)

Convolution - 2D Continuous Case:

$$(f * g)(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, \text{ß})\, g(x-a, y-\text{ß})\, da\, d\text{ß}$$

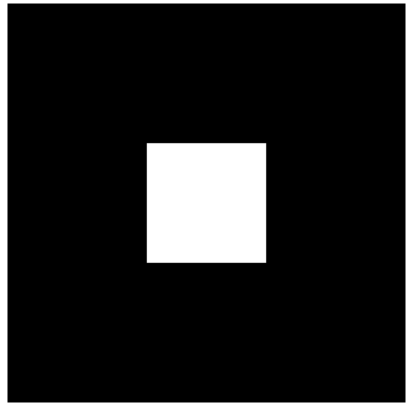
Grayscale Convolution - Example

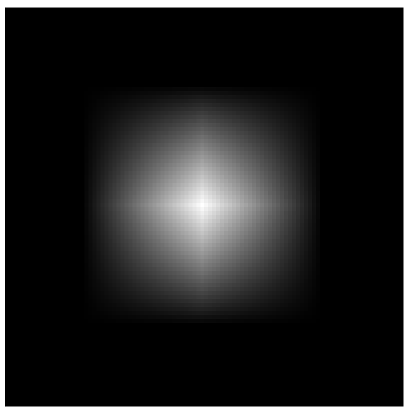
A



*

B





A * B

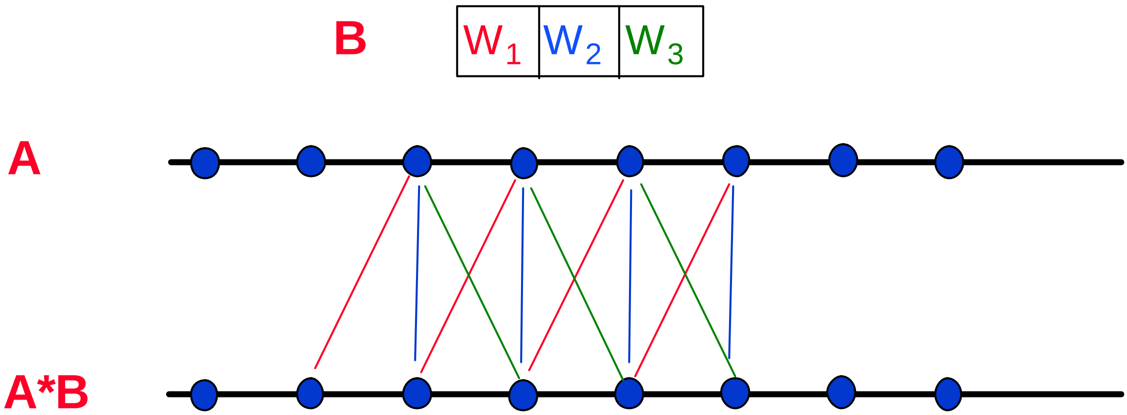
Convolution Properties

- Complexity:
  - Assume A is n x n and B is k x k then A*B takes $O(n^2k^2)$ operations.
- A*B = B*A
- (A*B)*C = A*(B*C)
  - If B and C are k x k then (A*B)*C takes $O(2n^2k^2)$ operations. However A*(B*C) takes $O(k^4+n^2k^2)$ operations, which is faster if k<<n .
- Separability
  - In some cases it is possible to decompose B (k x k) into B=C*D where C is 1 x k and D is k x 1. In such a case A*B takes $O(n^2k^2)$ while (A*C)*D takes $O(2n^2k)$.

$$\begin{bmatrix} 1 & -1 \end{bmatrix} * \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Mask Constraints

- Image Average
  - In order to preserve the overall average of A, the sum of B's elements should equal 1



If $W_1+W_2+W_3=1$ then Av(A)=Av(A*B)

$$d(x-x_0, y-y_0) = \begin{cases} 1 & \text{if } x=x_0 \text{ and } y=y_0 \\ 0 & \text{otherwise} \end{cases}$$

$d(x, y)$

0	0	0
0	1	0
0	0	0

$$A(x,y)*\delta(x-x_0, y-y_0) = A(x-x_0, y-y_0)$$

Convolution Masks - Example: The Delta Kernel

$$A(x,y)*\delta(x-x_0,y-y_0) = A(x-x_0,y-y_0)$$

$$d(x-x_0,y-y_0)=\begin{cases}1 & \text{if } x=x_0 \text{ and } y=y_0\\ 0 & \text{otherwise}\end{cases}$$

$$A(x,y)*\delta(x,y) = A(x,y)$$

d(x,y)

0	0	0
0	1	0
0	0	0

d(x-1,y-1)

0	0	1
0	0	0
0	0	0

A

1	2	3
4	5	6
7	8	9

*

d(x-1,y-1)

0	0	1
0	0	0
0	0	0

=

A(x-1,y-1)

0	4	5
0	7	8
0	0	0

(Zero padding)

=

A(x-1,y-1)

6	4	5
9	7	8
3	1	2

(Wrap around)

Grayscale Smoothing

Grayscale averaging = convolution with:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3 X 3

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

5 X 5

"Soft" Averaging: Convolution with a Gaussian

$$\frac{1}{2ps^{2}}\, e^{\frac{-(x^{2}+y^{2})}{2s^{2}}}$$

Discrete case:

(1/8) x

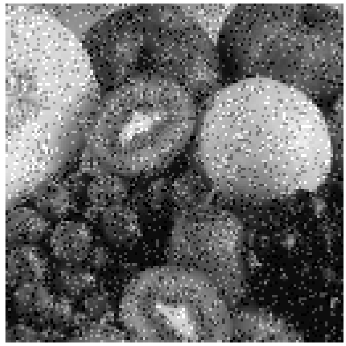
0	1	0
1	2	1
0	1	0

(1/81) x

1	2	3	2	1
2	4	6	4	2
3	6	9	6	3
2	4	6	4	2
1	2	3	2	1

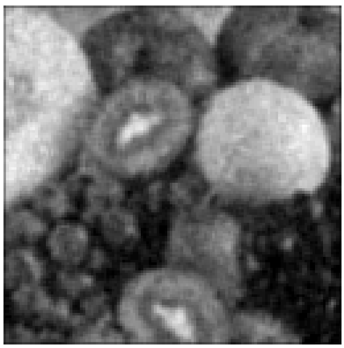
A seperable kernel

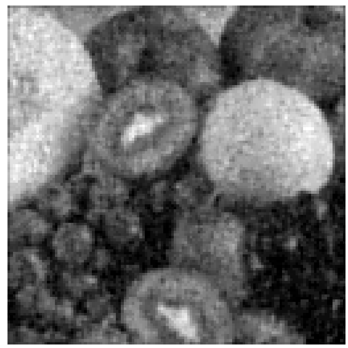
Normal vs Gaussian Grayscale Smoothing



Original
Noisy image

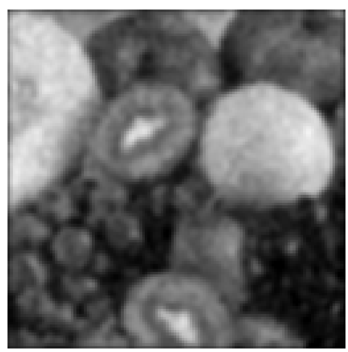
3 X 3
Average





3 X 3
Gaussian
Average

5 X 5
Average



5 X 5
Gaussian
Average

7 X 7
Average

7 X 7
Gaussian
Average

Median Filtering

S = neighborhood of pixel (x,y)

$$\text{New value at } (x,y) = \operatorname*{median}_{(x,y)\in S} \{I(x,y)\}$$

30	10	20
10	250	25
20	25	30

10, 10, 20, 20, 25, 25, 30, 30, 250

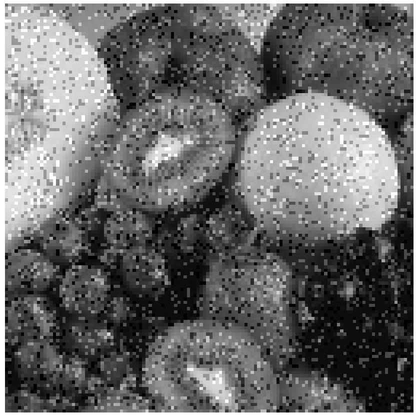
median

Median + Average: average the k central values.

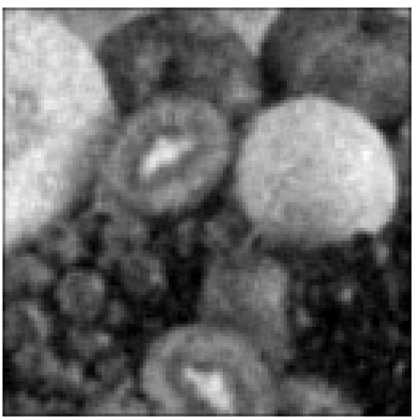
10, 10, 20, 20, 25, 25, 30, 30, 250

24

Median vs Average Filtering

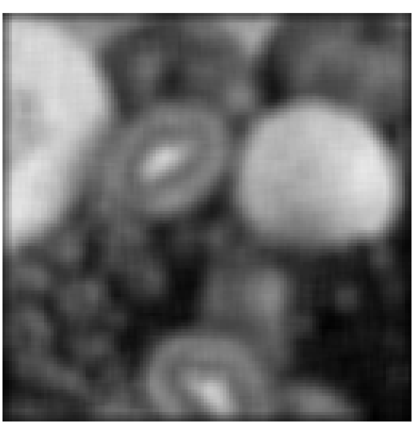


Salt & Pepper Noise



3 X 3 Average

5 X 5 Average



7 X 7 Average

Median

Multiple Median Filtering

Large Noise

Median

Median x 2

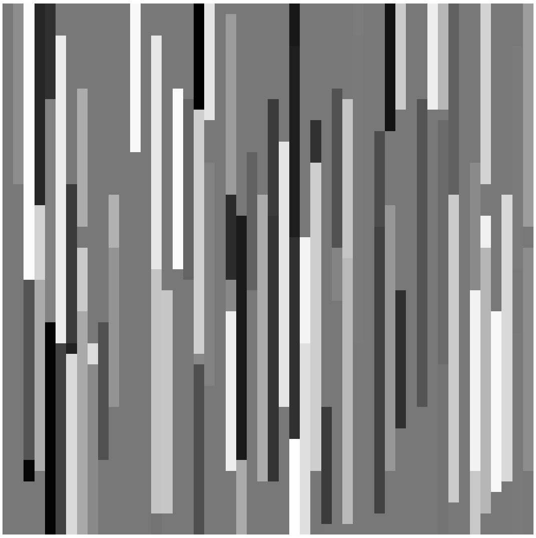
Median x 4

Median x 8

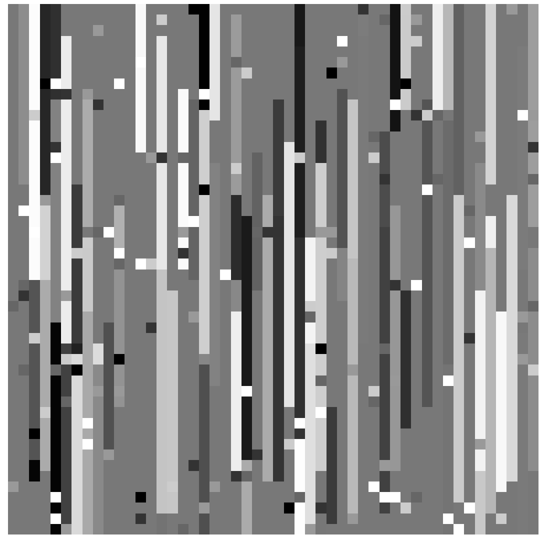
Median x 6

Median x 7

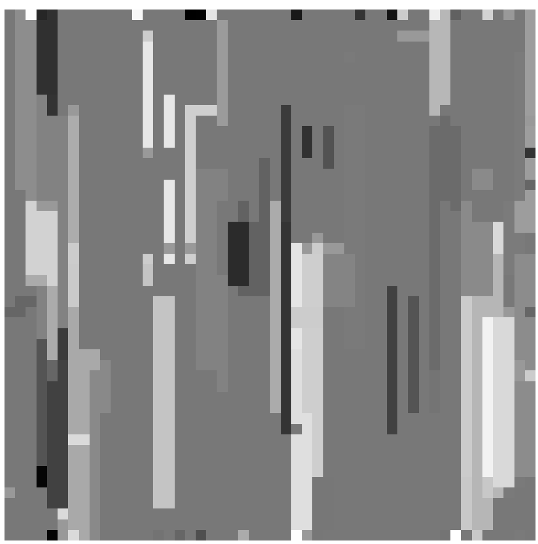
Median Filtering - Failure



Original

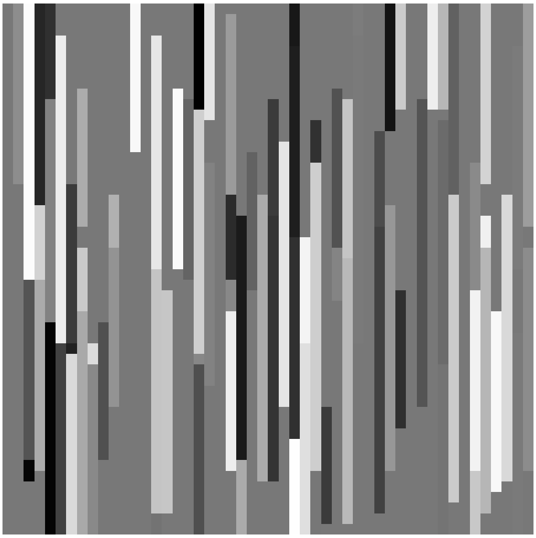


Salt & Pepper Noise

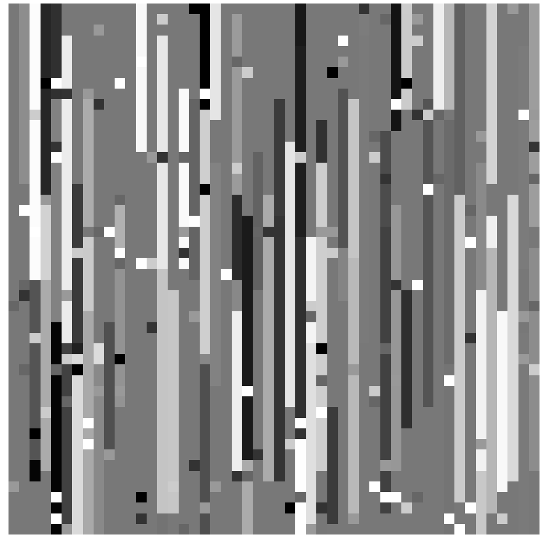


Median Filter

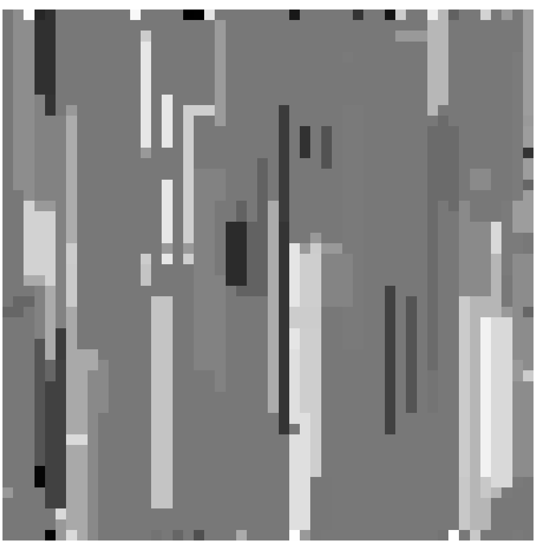
Oriented Median Filtering



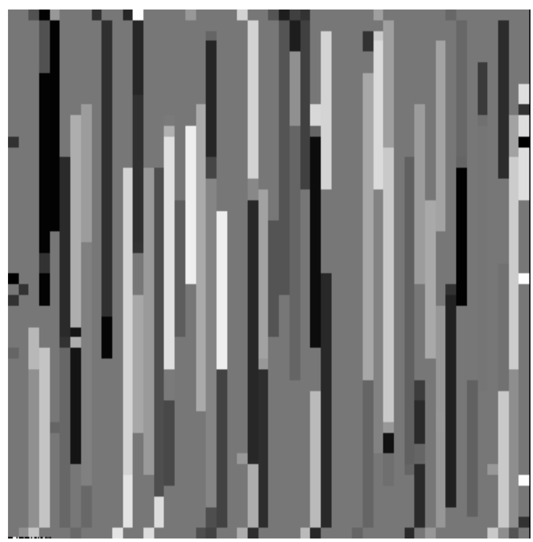
Original



Salt & Pepper Noise



Median Filter

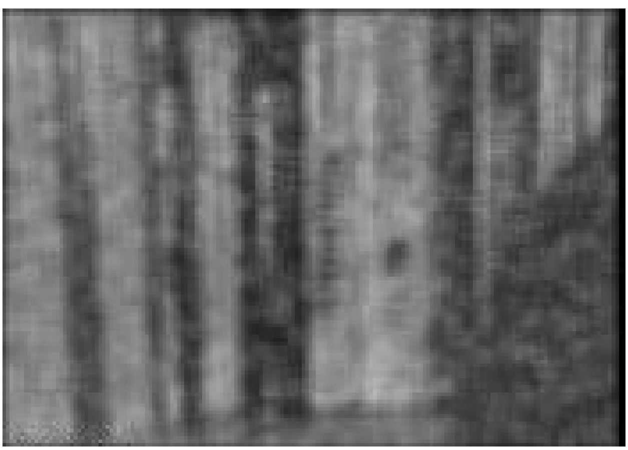


Oriented Median Filter

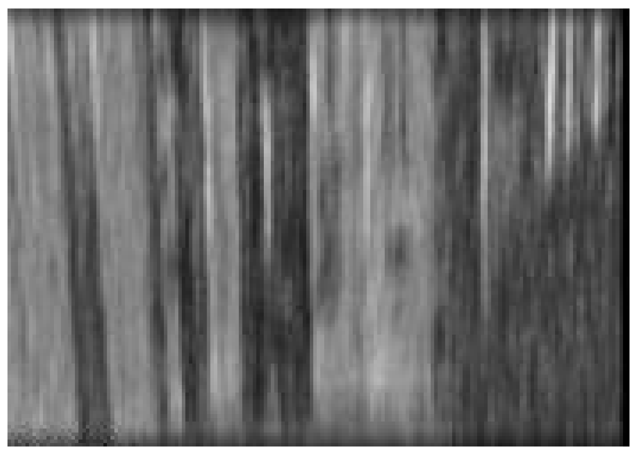
Oriented Filters

Salt & Pepper noise

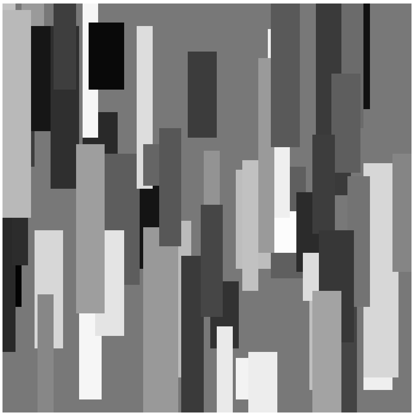
4x4 Average



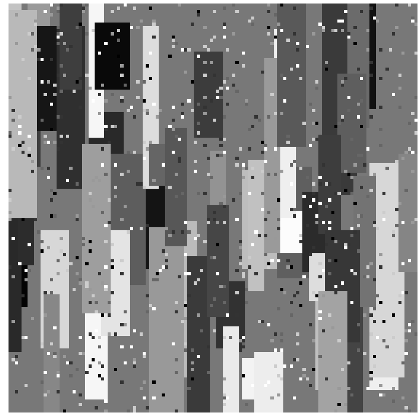
7x2 Average



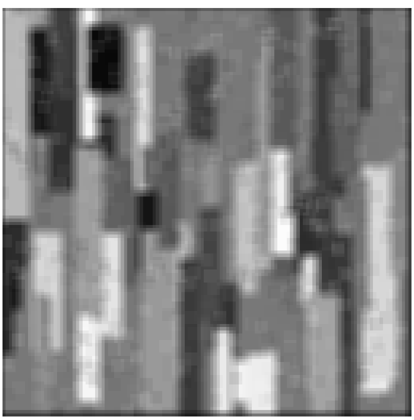
Oriented Filtering - Example



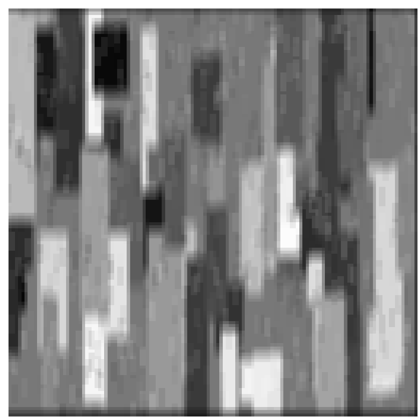
Original



Noisy Image



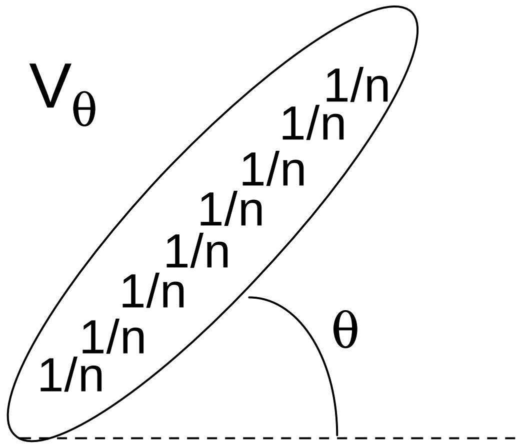
4x4 Average



Oriented 6x2 Average

Directional Smoothing

Define oriented masks:



Choose neighborhood with smallest variance and replace pixel value with the average of that neighborhood.

Directional Smoothing - Example

Original + Noise

3x3 Average

Directional Smoothing
(2x5, 5x2, diagonalx2)

Directional Smoothing - Example

Original + Noise

3x3 Average

Directional Smoothing
(2x5, 5x2, diagonalx2)

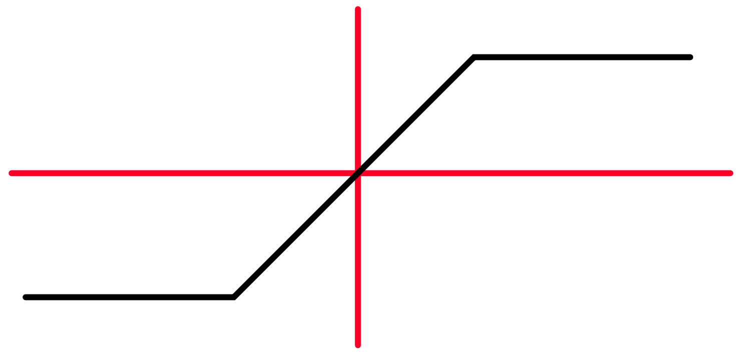
Sharpening

- A sharpening filter is applied in order to enhance edges and fine details (high frequency) in an image:
  - Assume $A*G$ is a smoothing filtering.
  - $A*(\delta - G)$ contains the fine details of the image.
  - $A+\lambda A*(\delta - G) = A*((1+\lambda)\delta-\lambda G)=A*S(\lambda)$ amplifies fine details in the image.
  - The parameter λ controls the amount of amplification.

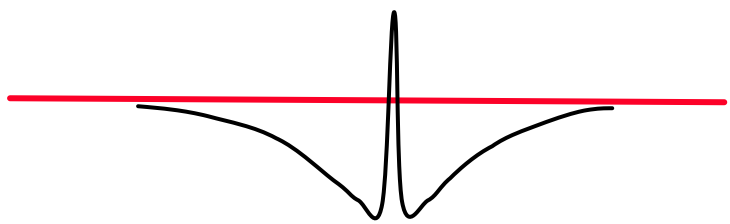
$$G = \begin{pmatrix} 0 & \frac{1}{8} & 0 \\ \frac{1}{8} & \frac{1}{2} & \frac{1}{8} \\ 0 & \frac{1}{8} & 0 \end{pmatrix} \qquad S(1) = \begin{pmatrix} 0 & -\frac{1}{8} & 0 \\ -\frac{1}{8} & \frac{3}{2} & -\frac{1}{8} \\ 0 & -\frac{1}{8} & 0 \end{pmatrix}$$

Ringing effect in edge enhancement

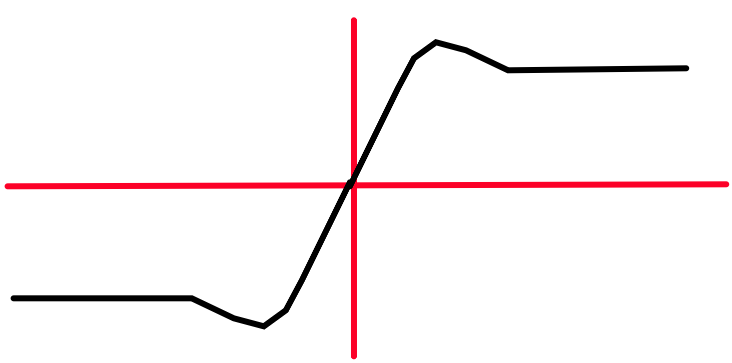
A



d-**G**



A*S

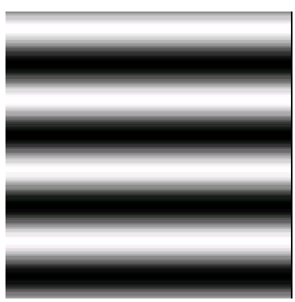


How can we enhance such an image?

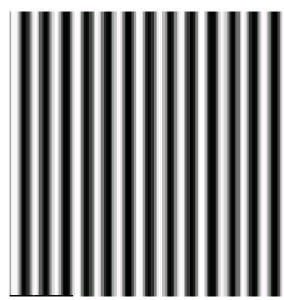
Solution: Image Representation

$$\begin{bmatrix} 2 & 1 & 3 \\ 5 & 8 & 7 \\ 0 & 3 & 5 \end{bmatrix} = 2\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 1\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \; + 3\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 5\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \ldots$$

= 3

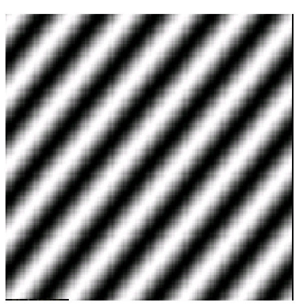


+ 5

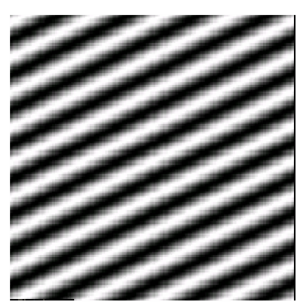


+

+ 10



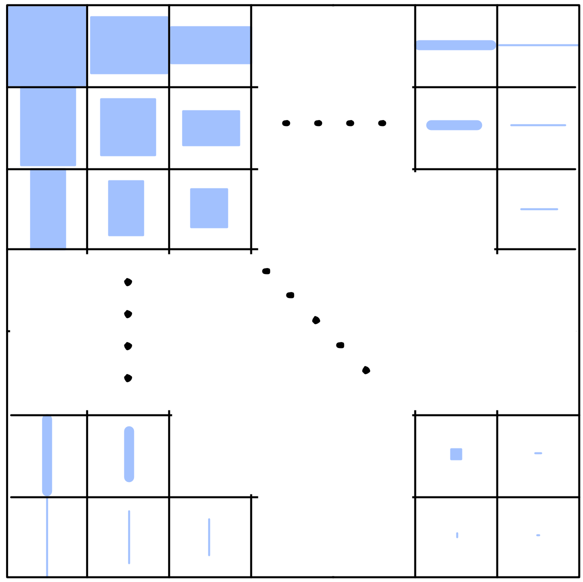
+ 23



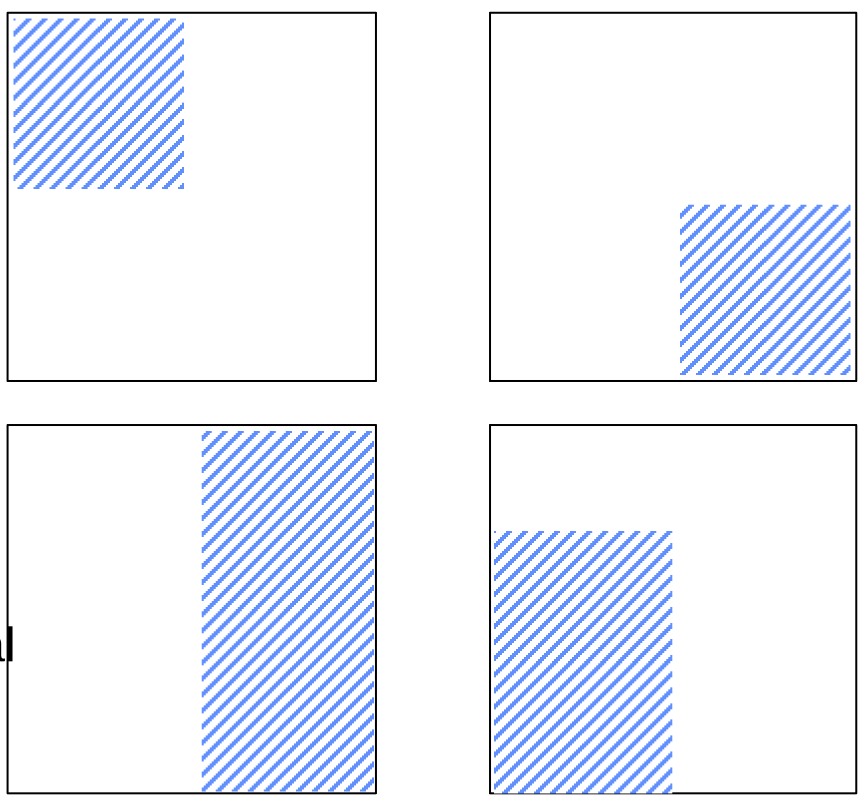
+ ...

Frequency Domain

Map of “Sizes and Orientations”



Evaluating an Image in terms of “sizes”:



Large

Small

Thin &
Horizontal

Vertical