

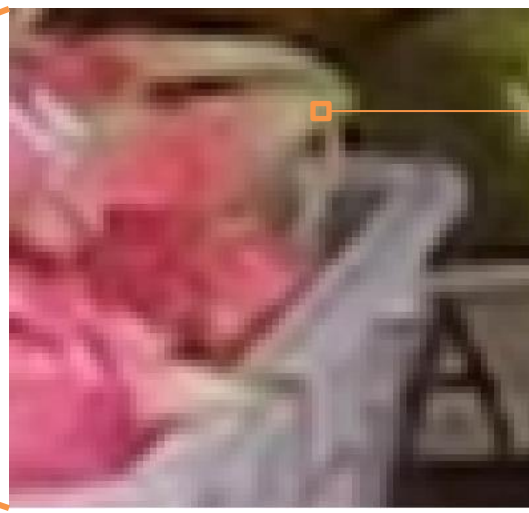


IMAGE FILTERING

HYUN SOO PARK

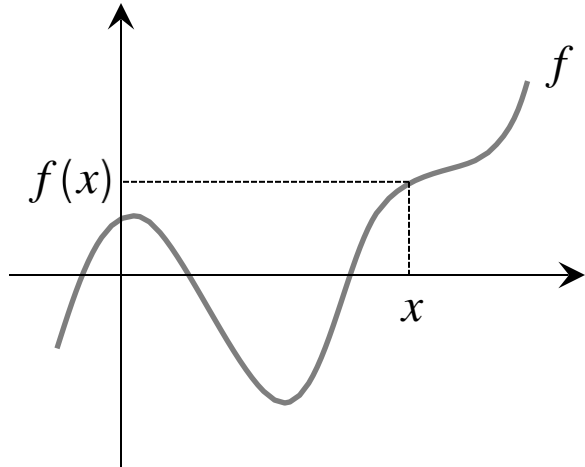


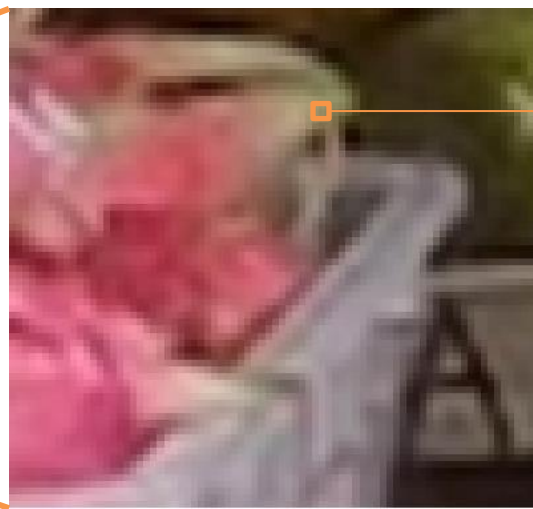




$I(x_1)$

Pixel value at x





$I(x_1)$

Pixel value at x

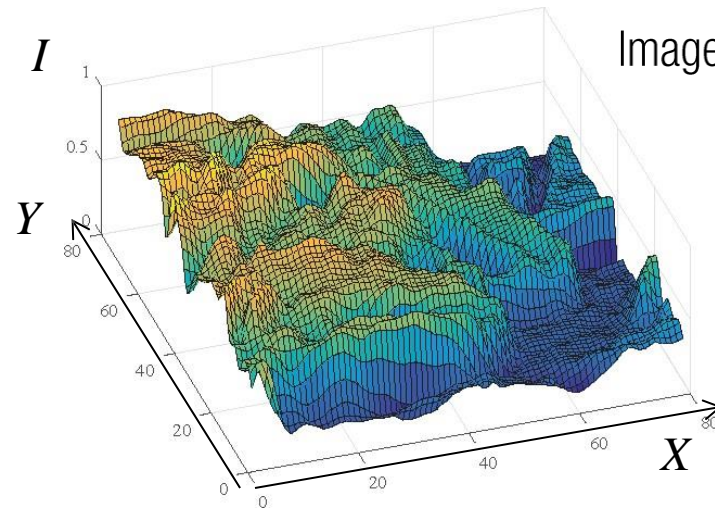
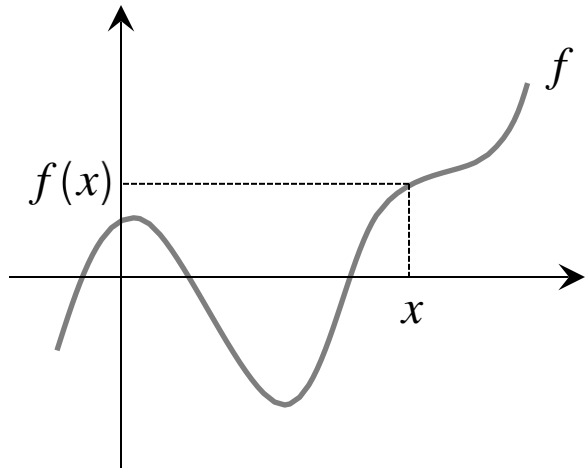
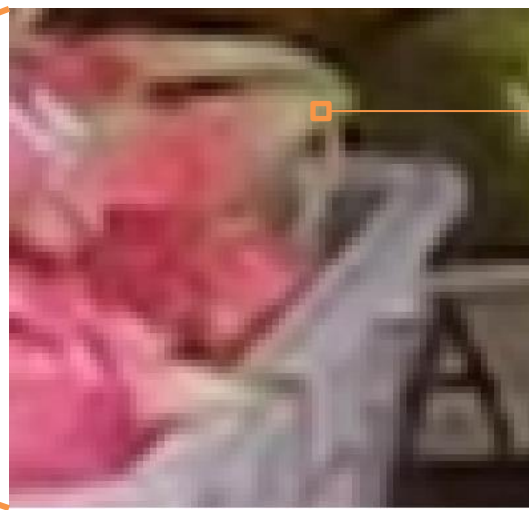


Image as a 2D function



$$I(x_1) = (212, 200, 221)$$

Pixel value at x

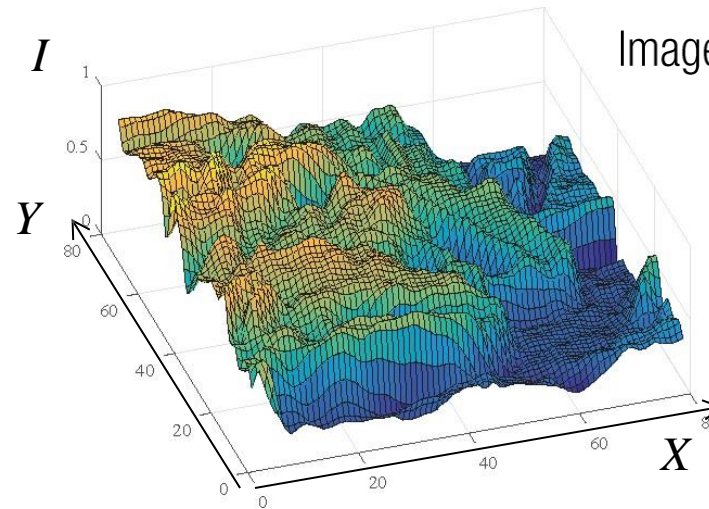
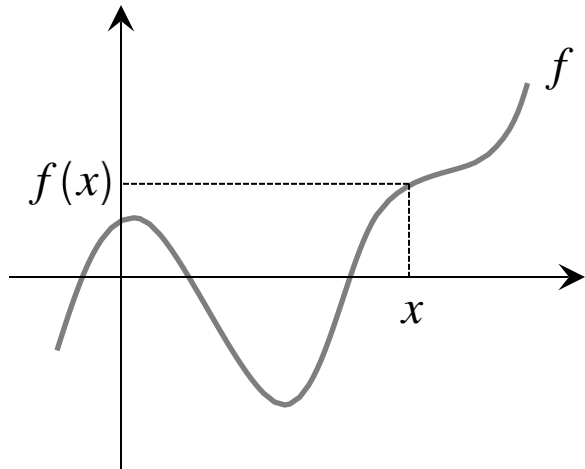
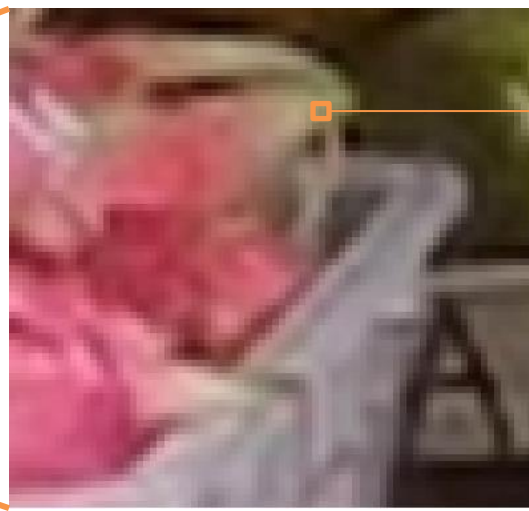


Image as a 2D function



$I(x_1) = (212, 200, 221)$

Pixel value at x

Select Color:

A color selection interface. It features a large color gradient square on the left, a vertical color bar on the right, and a panel on the right with input fields for color properties. The selected color is a dark red.

H: 0°

S: 57%

B: 69%

R: 175

G: 75

B: 75

AF4B4B



$I(x_1) = (212, 200, 221)$

Pixel value at x

$I(x_2) = (212, 20, 51)$

Select Color:

The interface includes a large color gradient area with a white circle, a vertical color bar, and several input fields:

- H: 0°
- S: 57%
- B: 69%
- R: 175
- G: 75
- B: 75
- # AF4B4B



$$I(x_1) = (212, 200, 221)$$

Pixel value at x

$$I(x_2) = (212, 20, 51)$$

$$I(x) \in 2^8 \times 2^8 \times 2^8$$

8-bit image



$$I(x_1) = (212, 200, 221)$$

Pixel value at x

$$I(x_2) = (212, 20, 51)$$

$I(x) \in 2^8 \times 2^8 \times 2^8$
8-bit image

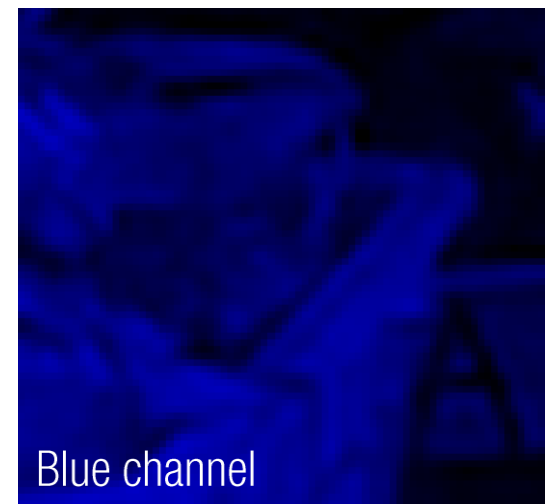
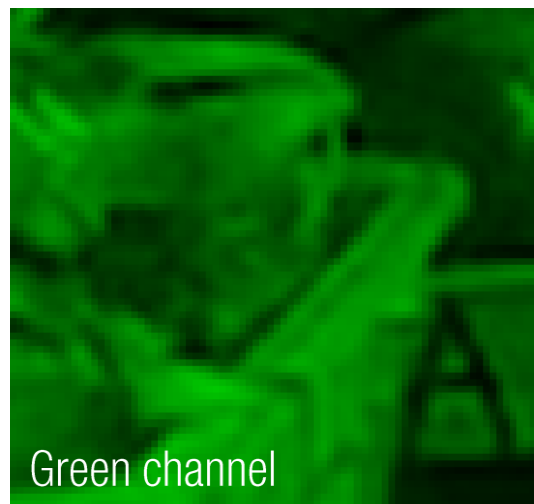
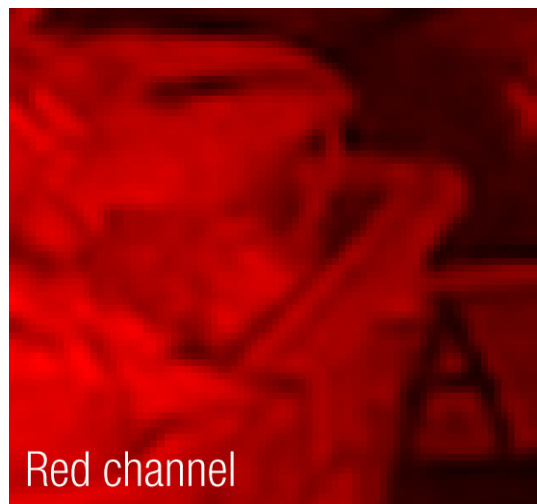


IMAGE FILTERING



$$f(I) = I_f$$

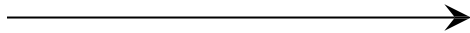


IMAGE FILTERING



$$I(x_1) = (212, 200, 221)$$

$$f(I) = I_f$$



$$\begin{aligned} f(I(x_1)) &= (212, 200, 221) \\ &= (242, 152, 135) \end{aligned}$$

IMAGE FILTERING (PIXEL-WISE FILTERING)



$$I_f = I + 100$$



$$I_f = I - 100$$



$$I_f = 255 - I$$

IMAGE FILTERING (PIXEL-WISE FILTERING)



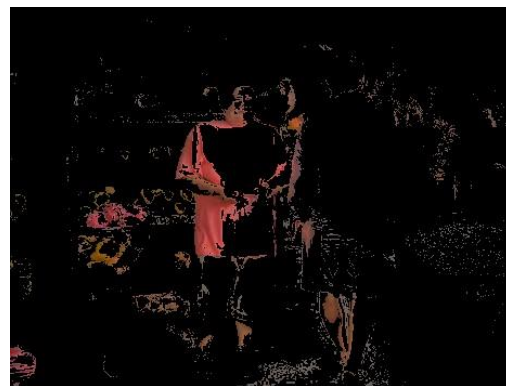
$$I_f = I + 100$$



$$I_f = I - 100$$



$$I_f = 255 - I$$



$$I_r > 100 \ \& \ I_g < 100 \ \& \ I_b < 100$$

IMAGE FILTERING (PIXEL-WISE FILTERING)



$$I_r > 100 \ \& \ I_g < 100 \ \& \ I_b < 100$$



Blue/black

White/gold





IMAGE FILTERING (PIXEL-WISE FILTERING)



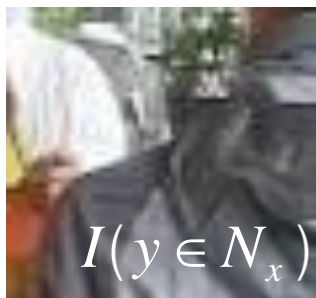
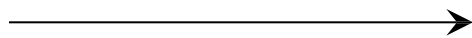
Color cannot be trusted.
Spatial cue can be useful.

$$I_r > 100 \ \& \ I_g < 100 \ \& \ I_b < 100$$

IMAGE SPATIAL FILTERING



$$I_f(x) = f(I(y \in N_x))$$



Local patch around x

N_x : Neighboring pixels of x

IMAGE SPATIAL FILTERING

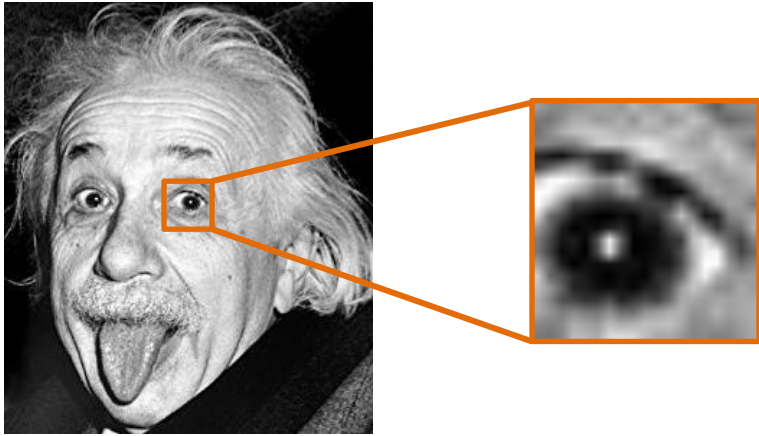
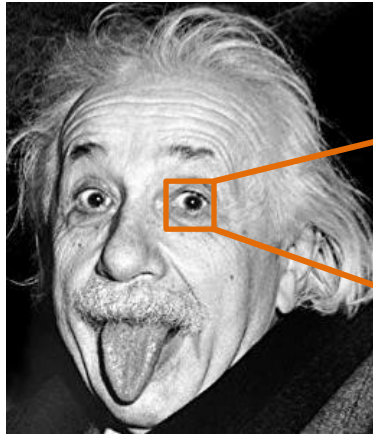


IMAGE SPATIAL FILTERING

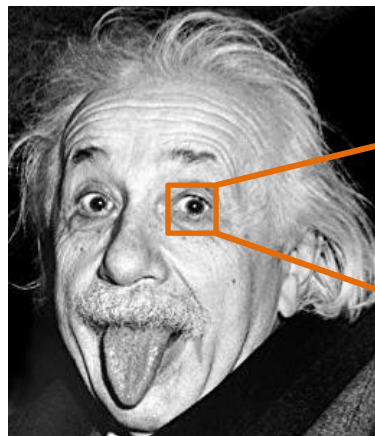


2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

I

Image

IMAGE SPATIAL FILTERING



2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

I

Image

$$I_f(i, j) = \sum_{k,l} I(i+k, j+l)z(k,l)$$

\otimes

a	b	c
d	e	f
g	h	i

z

Filter

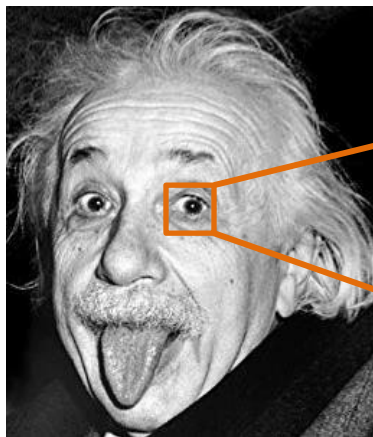
=

I_f

Filtered image

IMAGE SPATIAL FILTERING

$$I_f(i, j) = \sum_{k,l} I(i+k, j+l)z(k,l)$$



2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

Image

⊗

a	b	c
d	e	f
g	h	i

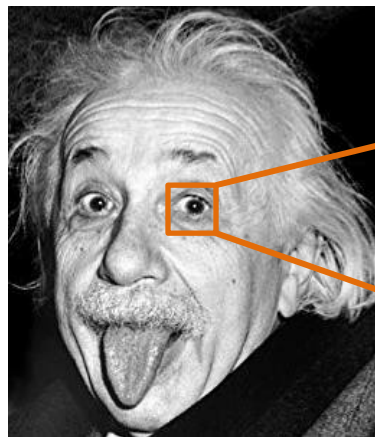
Filter

=

	y			

Filtered image

IMAGE SPATIAL FILTERING



2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

Image

$$I_f(i, j) = \sum_{k,l} I(i+k, j+l)z(k,l)$$

⊗

a	b	c
d	e	f
g	h	i

Filter

=

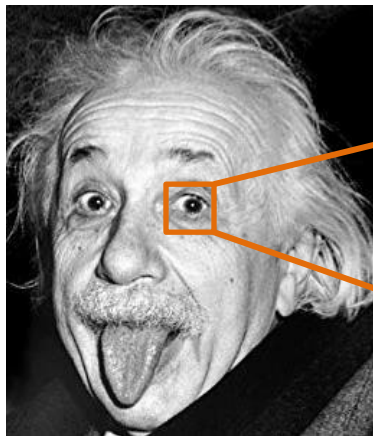
	y			

Filtered image

$$y = 2a + b + 4c + d + 2e + 2f + 3g + 3h + 5i$$

IMAGE SPATIAL FILTERING

$$I_f(i, j) = \sum_{k,l} I(i+k, j+l)z(k,l)$$



2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

Image

⊗

a	b	c
d	e	f
g	h	i

Filter

=

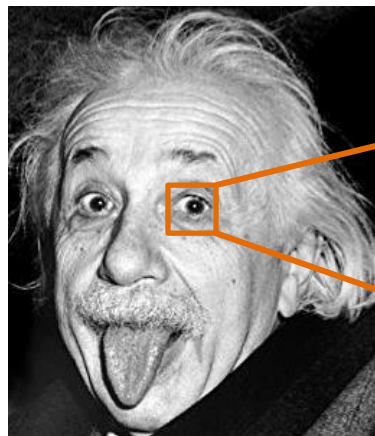
		y		

Filtered image

$$y = a + 4b + 4c + 2d + 2e + 3f + 3g + 5h + 8i$$

IMAGE SPATIAL FILTERING

$$I_f(i, j) = \sum_{k,l} I(i+k, j+l)z(k,l)$$



2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

Image

⊗

<i>a</i>	<i>b</i>	<i>c</i>
<i>d</i>	<i>e</i>	<i>f</i>
<i>g</i>	<i>h</i>	<i>i</i>

Filter

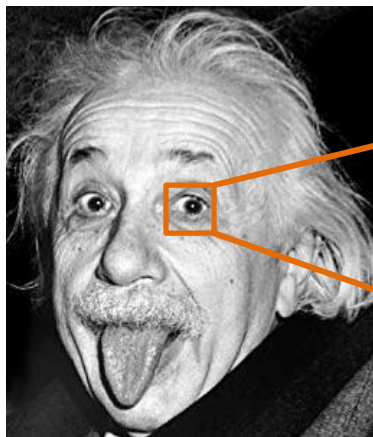
=

			<i>y</i>	

Filtered image

$$y = 5a + 8b + 9c + 2d + 6e + 7f + 2g + h + 3i$$

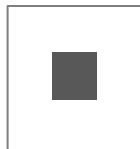
IDENTITY



2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

$$I_f(i, j) = \sum_{k, l} I(i + k, j + l) z(k, l)$$

⊗



=

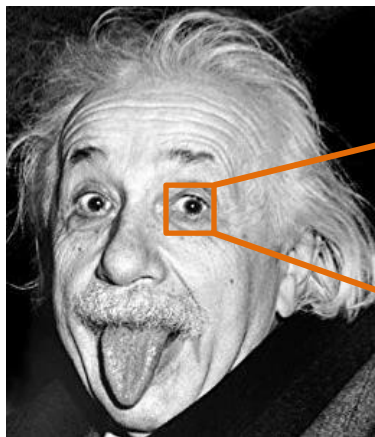
⊗

0	0	0
0	1	0
0	0	0

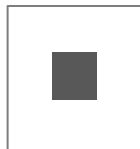
=

IDENTITY

$$I_f(i, j) = \sum_{k,l} I(i+k, j+l)z(k,l)$$



⊗



=



2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

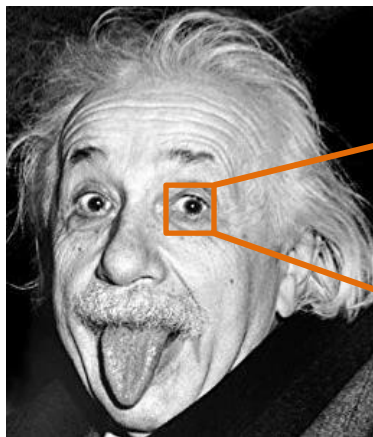
⊗

0	0	0
0	1	0
0	0	0

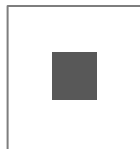
=

IDENTITY

$$I_f(i, j) = \sum_{k, l} I(i + k, j + l) z(k, l)$$



⊗



=



2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

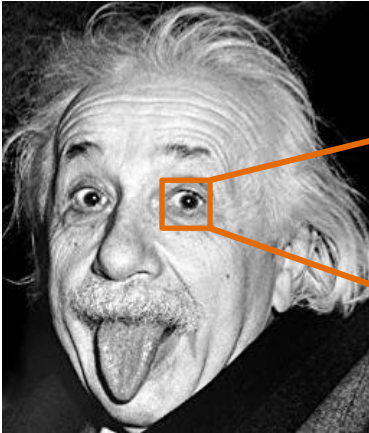
⊗

0	0	0
0	1	0
0	0	0

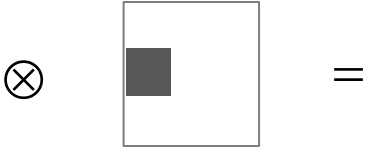
=

	2	2	3	
	3	5	8	
	2	2	6	

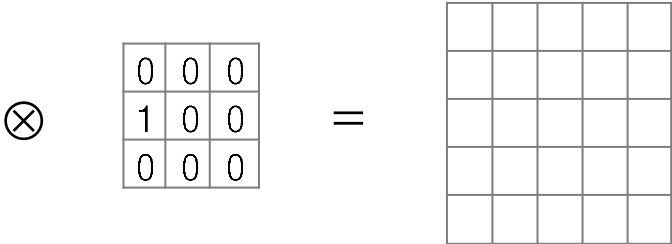
SHIFTING



$$I_f(i, j) = \sum_{k,l} I(i+k, j+l)z(k,l)$$

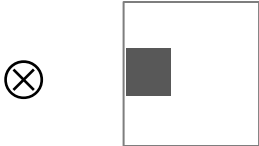
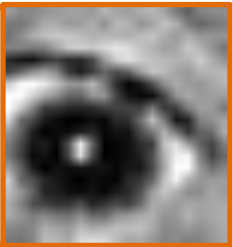
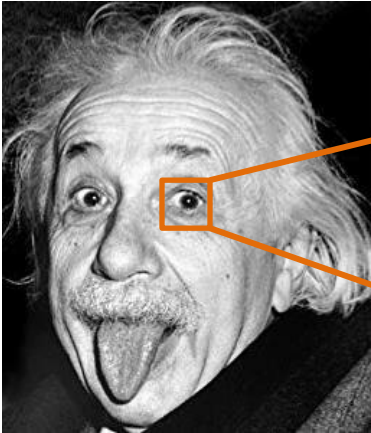


2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

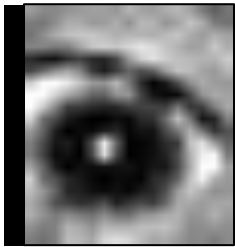


SHIFTING

$$I_f(i, j) = \sum_{k,l} I(i+k, j+l)z(k,l)$$



=



2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

⊗

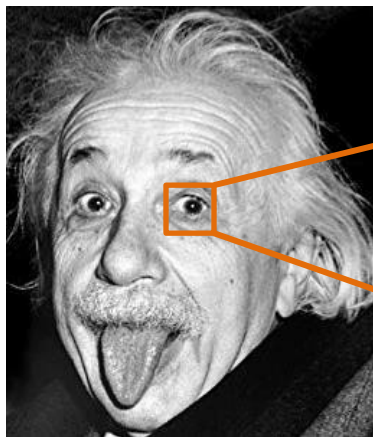
0	0	0
1	0	0
0	0	0

=

	1	2	2	
	3	3	5	
	5	2	2	

BOX FILTER (MOVING AVERAGING)

$$I_f(i, j) = \sum_{k,l} I(i+k, j+l) z(k,l)$$



⊗



2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

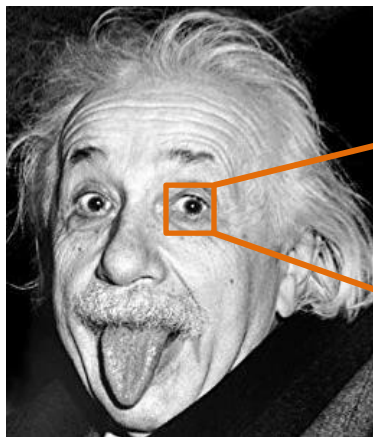
⊗

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

BOX FILTER (MOVING AVERAGING)

$$I_f(i, j) = \sum_{k,l} I(i+k, j+l) z(k,l)$$

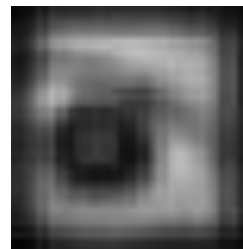


2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

⊗



=



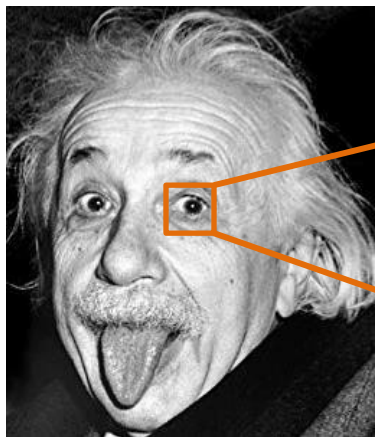
⊗

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Average

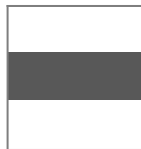
DIRECTIONAL BLUR



2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

$$I_f(i, j) = \sum_{k, l} I(i + k, j + l) z(k, l)$$

⊗

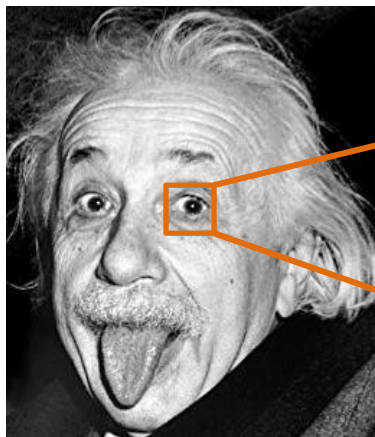


⊗

$\frac{1}{3}$

0	0	0
1	1	1
0	0	0

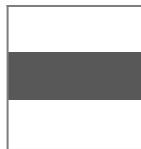
DIRECTIONAL BLUR



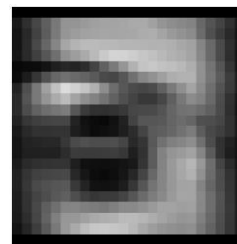
2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

$$I_f(i, j) = \sum_{k, l} I(i + k, j + l) z(k, l)$$

⊗



=



⊗

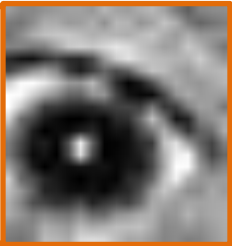
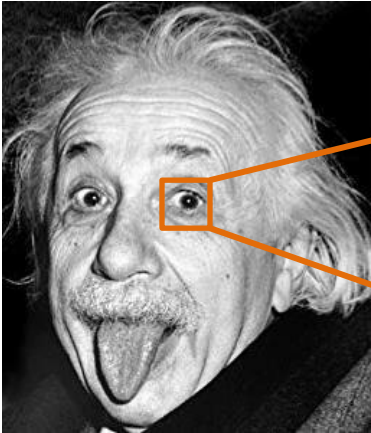
$\frac{1}{3}$

0	0	0
1	1	1
0	0	0

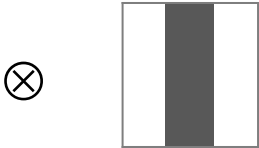
Horizontal blur

DIRECTIONAL BLUR

$$I_f(i, j) = \sum_{k,l} I(i+k, j+l)z(k,l)$$

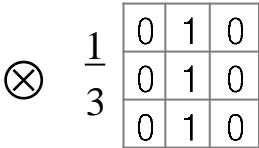
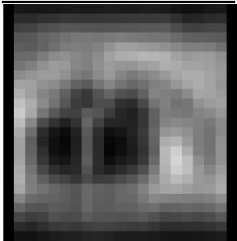


2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3



⊗

=

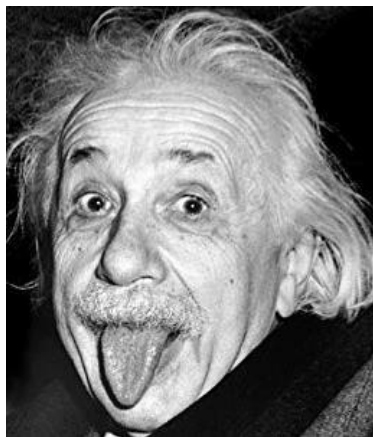


⊗

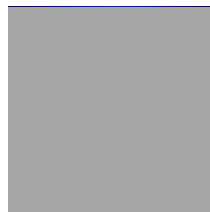
$\frac{1}{3}$

Vertical blur

BOX FILTER (MOVING AVERAGING)



\otimes

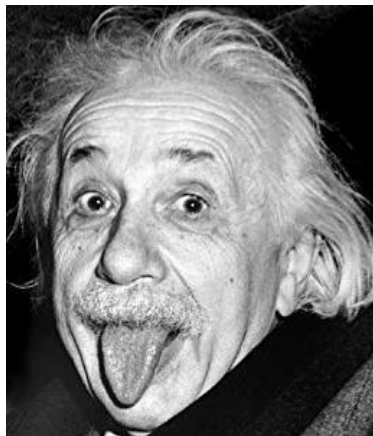


=

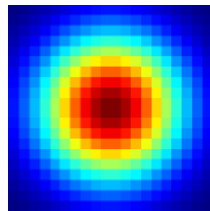
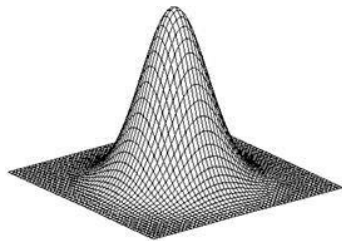


Pixel averaging

GAUSSIAN BLURRING (MOVING WEIGHTED AVERAGE)



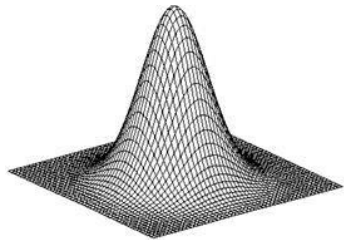
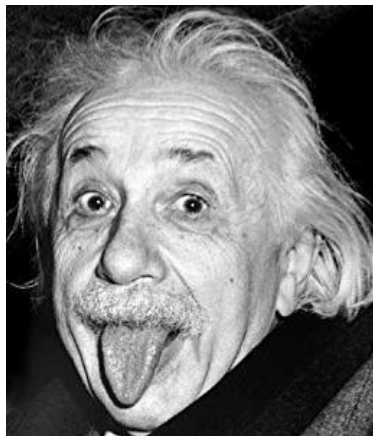
⊗



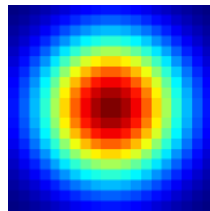
=

$$z(k,l) = \frac{1}{2\pi\sigma^2} e^{-\frac{k^2+l^2}{2\sigma^2}}$$

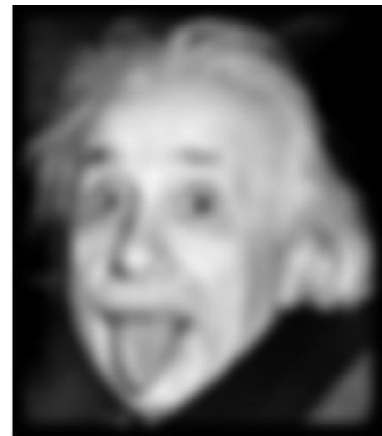
GAUSSIAN BLURRING (MOVING WEIGHTED AVERAGE)



⊗



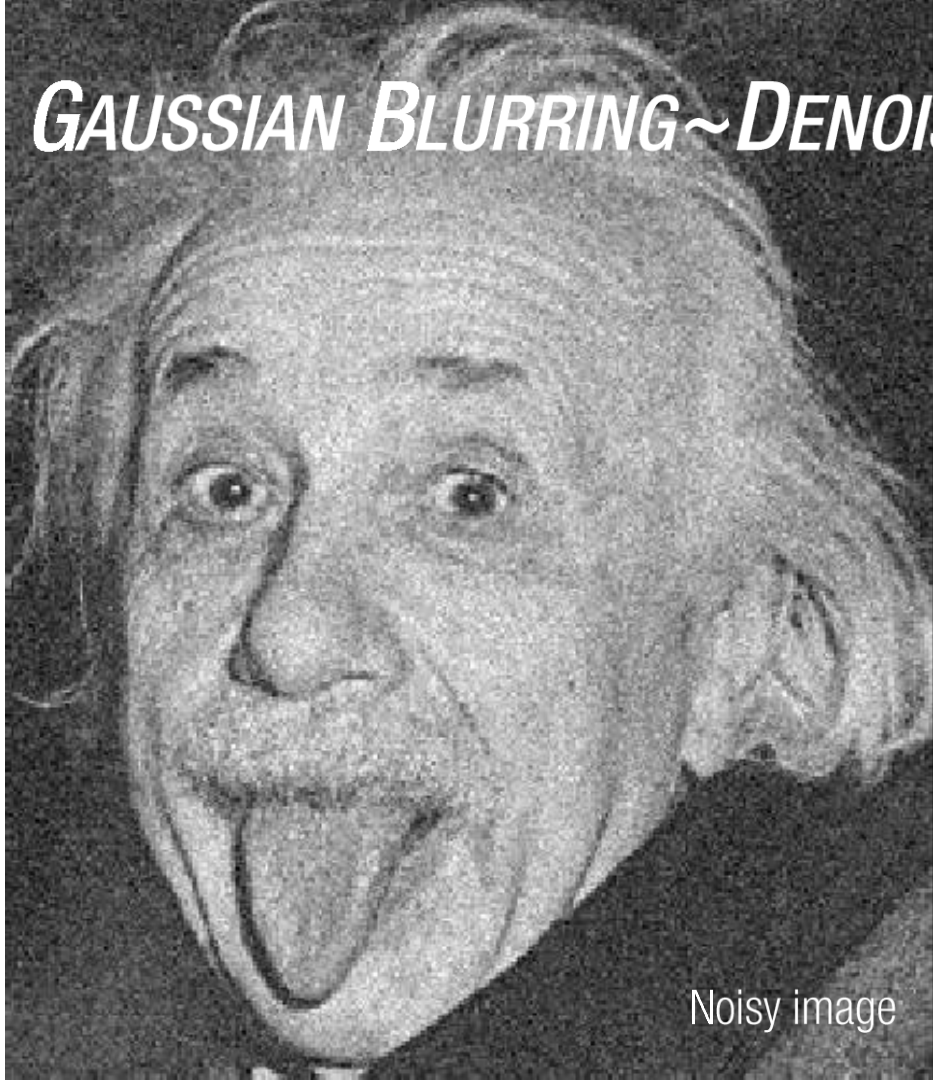
=



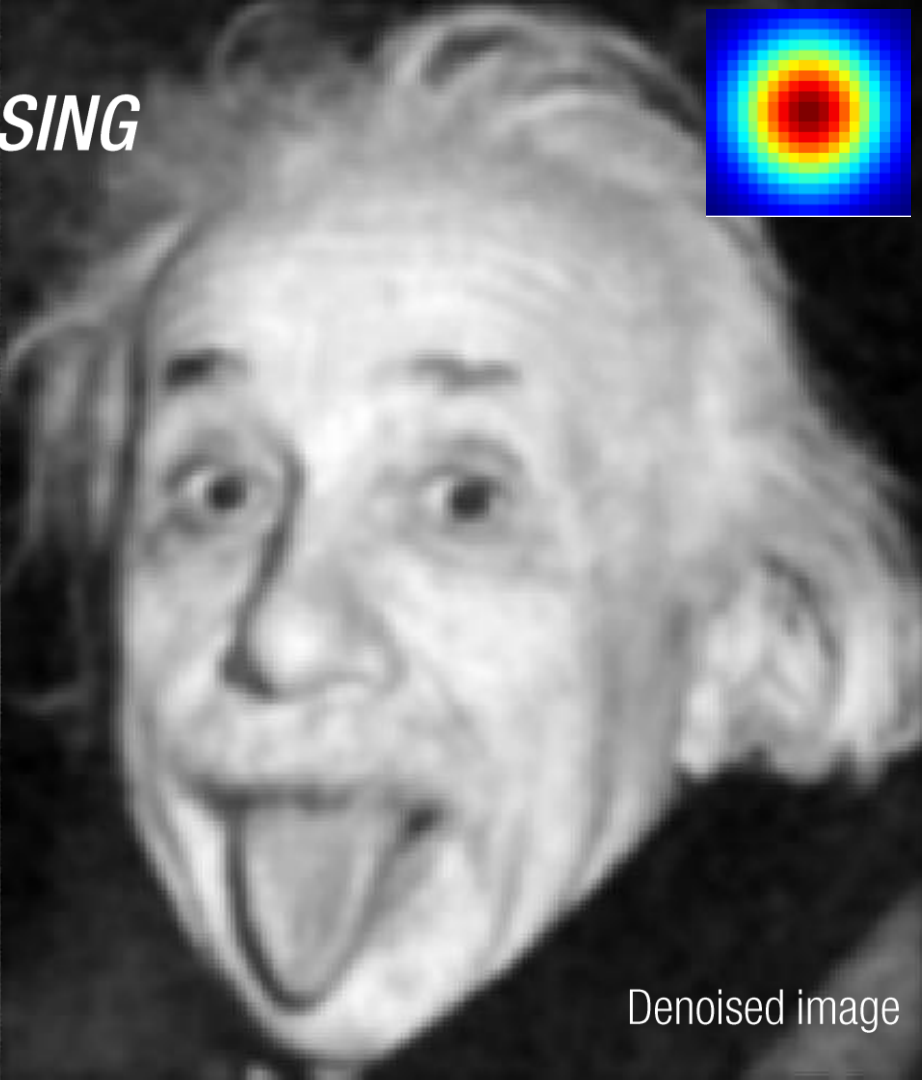
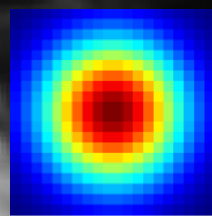
$$z(k,l) = \frac{1}{2\pi\sigma^2} e^{-\frac{k^2+l^2}{2\sigma^2}}$$

Gaussian blur

GAUSSIAN BLURRING ~ DENOISING



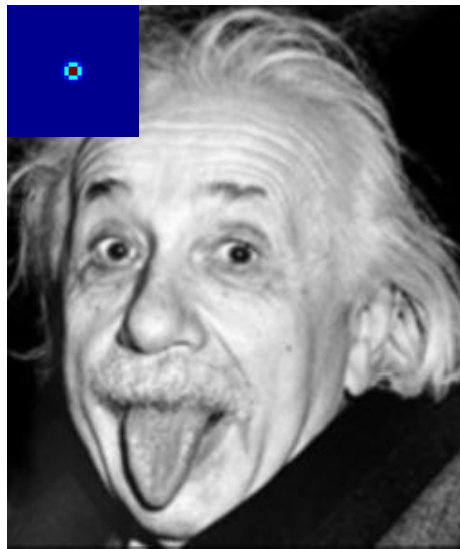
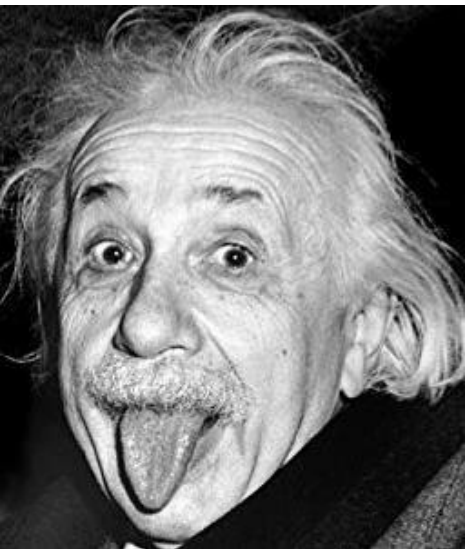
Noisy image



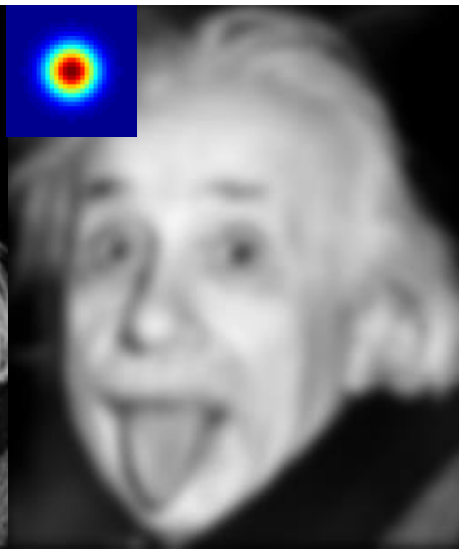
Denoised image

GAUSSIAN BLURRING

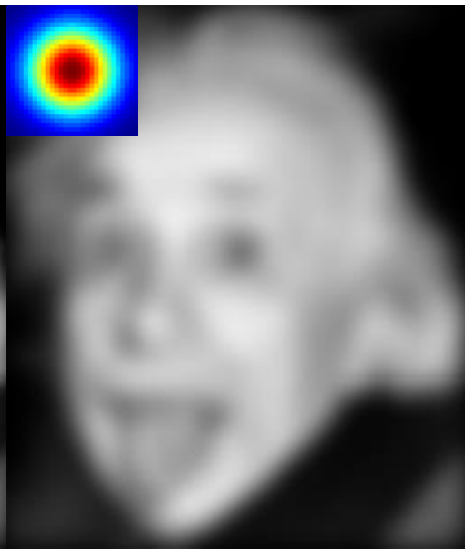
$$z(k,l) = \frac{1}{2\pi\sigma^2} e^{-\frac{k^2+l^2}{2\sigma^2}}$$



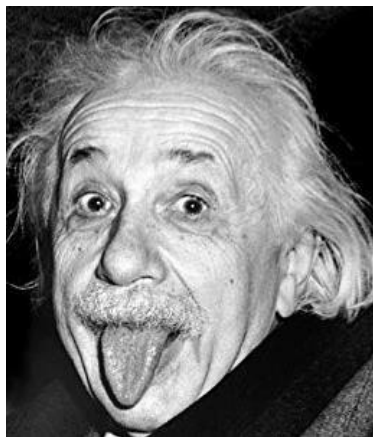
$\sigma=1$



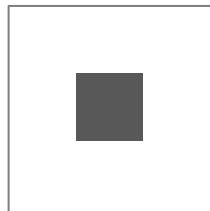
$\sigma=4$



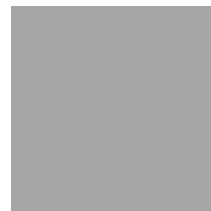
$\sigma=7$



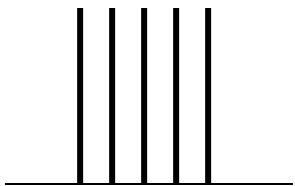
\otimes



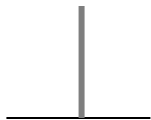
-



=



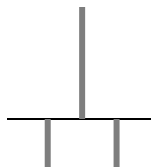
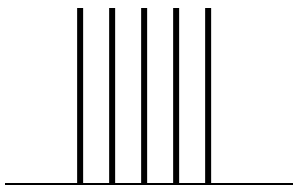
⊗



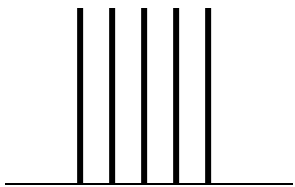
-



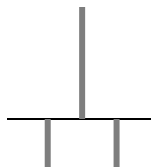
=



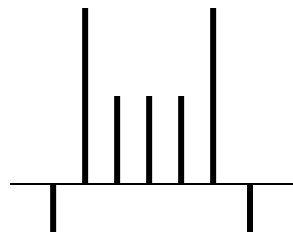
=



⊗

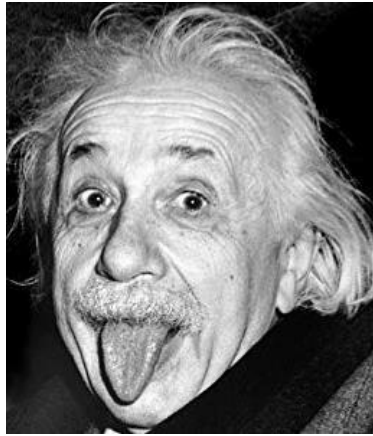


=

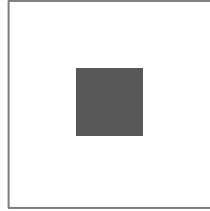


High response at edge

IMAGE SHARPENING



\otimes

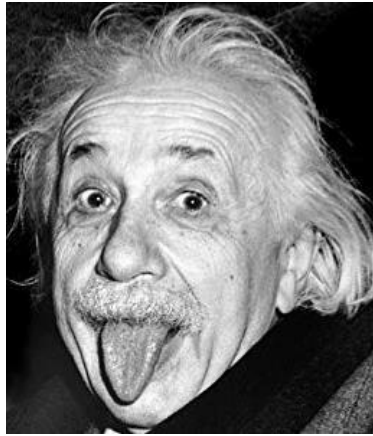


-

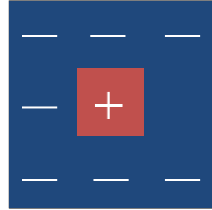


=

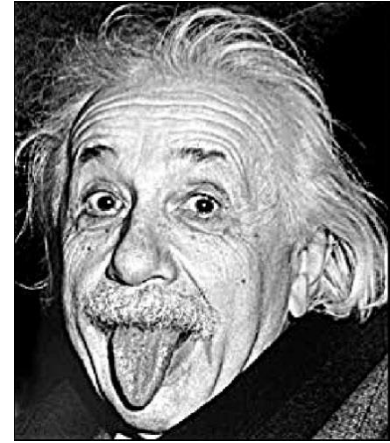
IMAGE SHARPENING



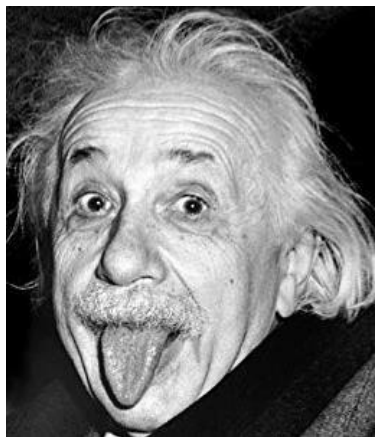
\otimes



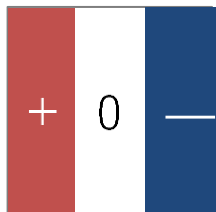
=



Sharpening

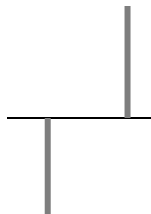
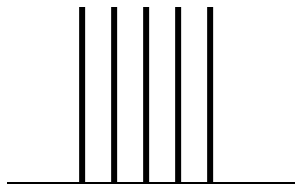


⊗

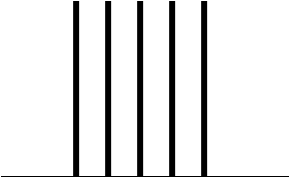


=

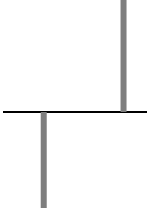
DIFFERENTIATION



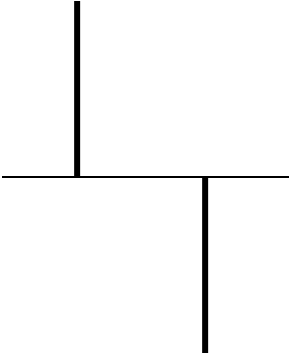
DIFFERENTIATION



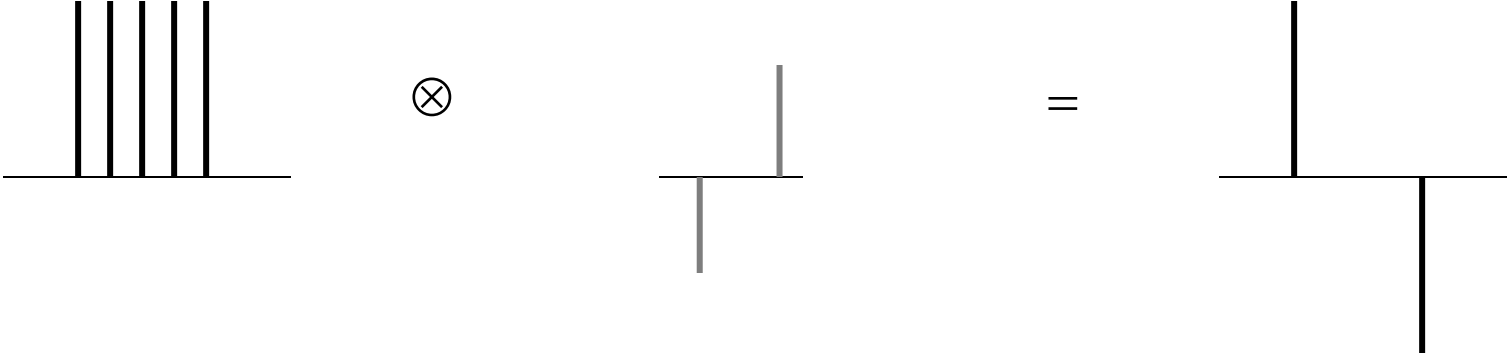
⊗



=



DIFFERENTIATION



$$\frac{df}{du} = \lim_{h \rightarrow 0} \frac{f(u+h) - f(u-h)}{2h} \longrightarrow I \otimes z = \frac{\partial I}{\partial u}$$

DIFFERENTIATION

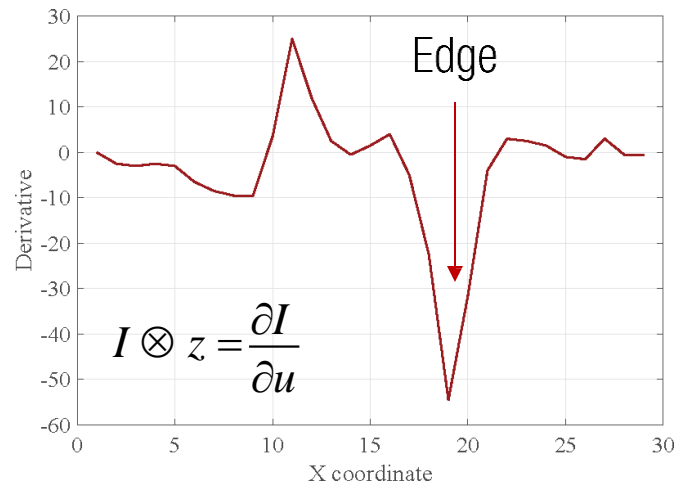
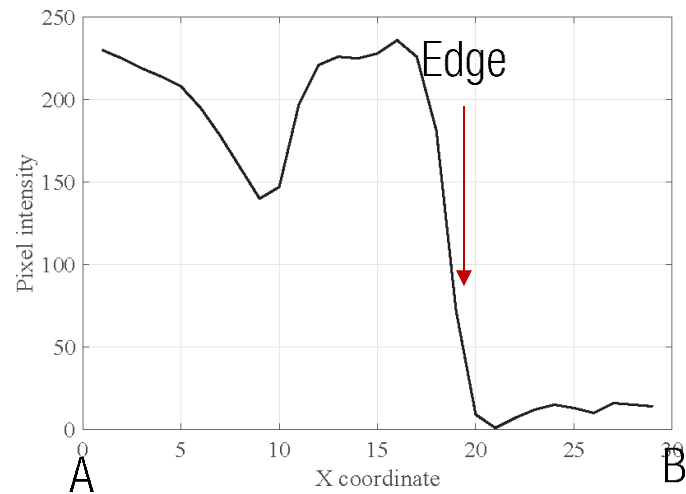
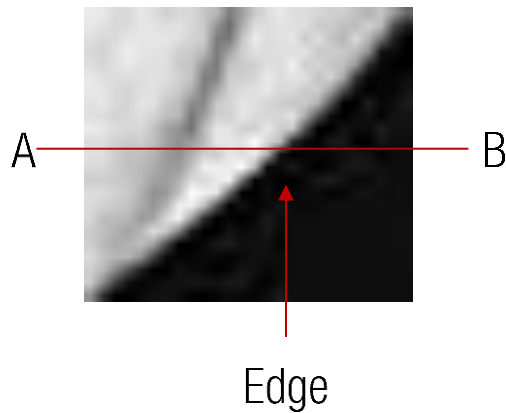
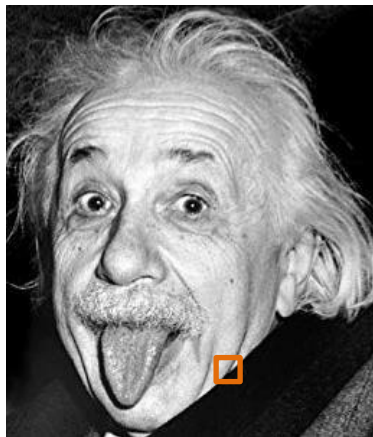
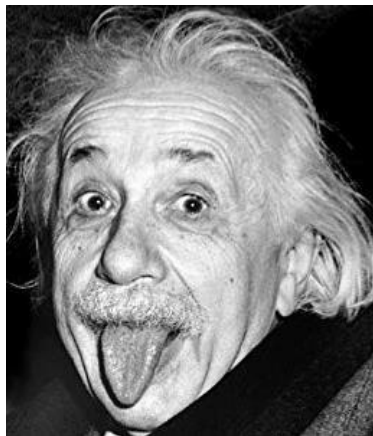
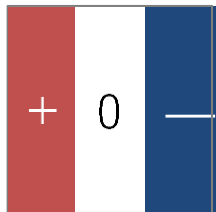


IMAGE DIFFERENTIATION

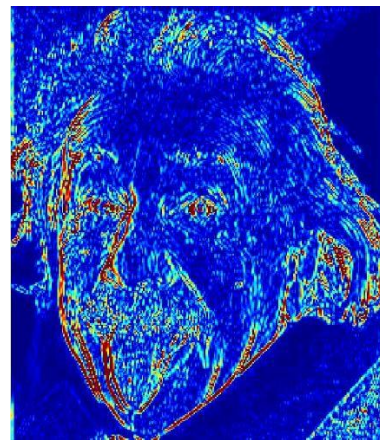


\otimes



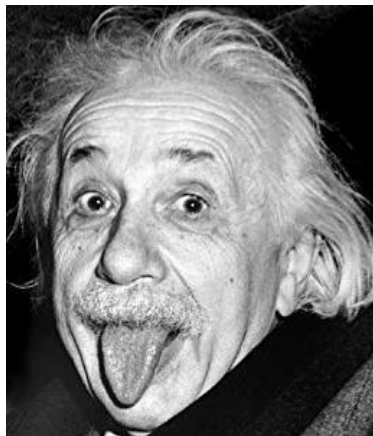
=

$$I \otimes z = \frac{\partial I}{\partial u}$$

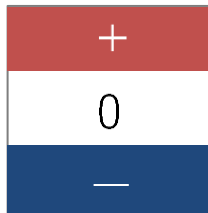


Differentiation

IMAGE DIFFERENTIATION

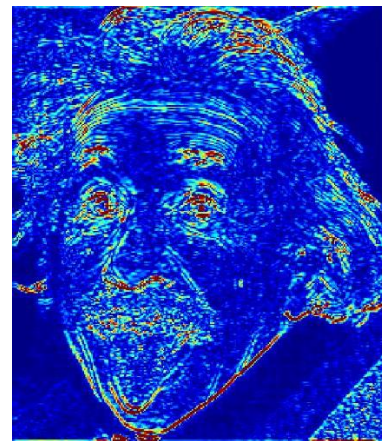


\otimes



=

$$I \otimes z = \frac{\partial I}{\partial v}$$



Differentiation

EDGE RESPONSE

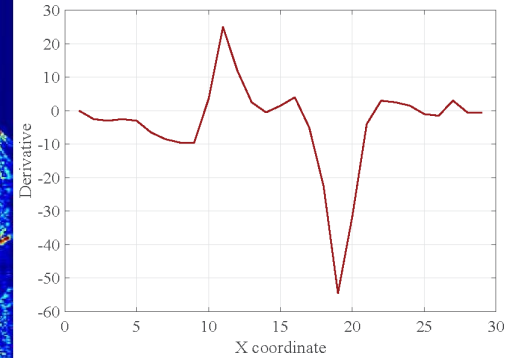
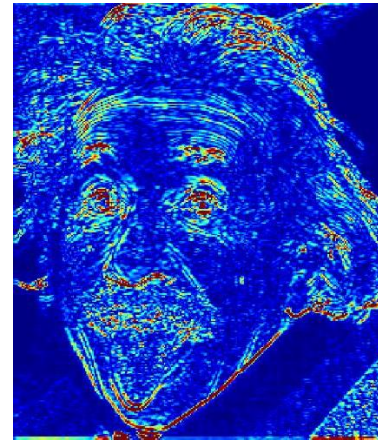
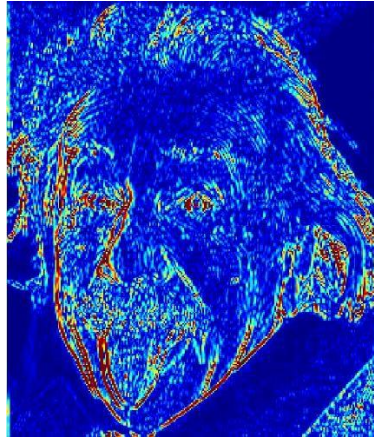
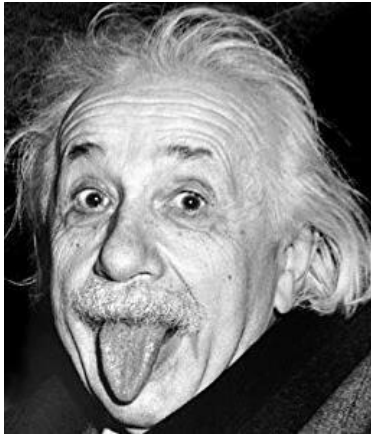


Image differentiation can be used for edge detection but it is VERY noisy.

NONLINEAR FILTER

