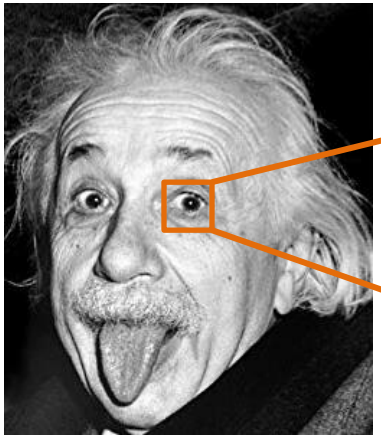
Image Convolution

Hyun Soo Park

IMAGE SPATIAL FILTERING ~ CORRELATION

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



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

I

Image

$\otimes$

a	b	c
d	e	f
g	h	i

z

Filter

$=$

	y			

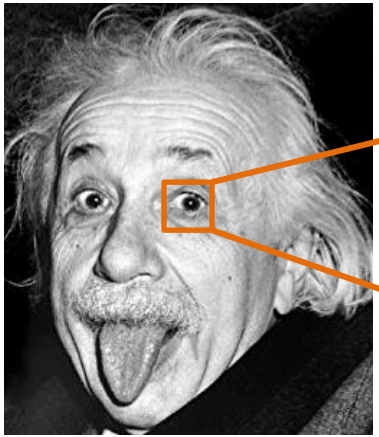
I_f

Filtered image

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

BOUNDARY

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



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

I

Image

$\otimes$

a	b	c
d	e	f
g	h	i

z

Filter

$=$

y				

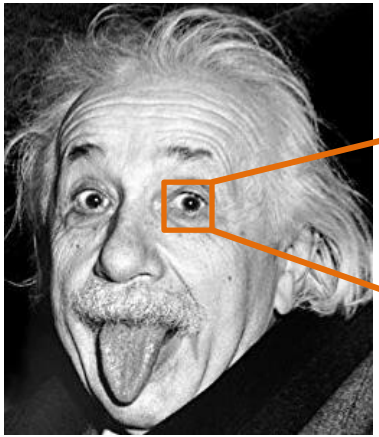
I_f

Filtered image

$y = ?$

BOUNDARY

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



0	0	0	0	0	0	0
0	2	1	4	4	7	0
0	1	2	2	3	6	0
0	3	3	5	8	9	0
0	5	2	2	6	7	0
0	8	3	2	1	3	0
0	0	0	0	0	0	0

I

Image

Zero padding

$\otimes$

a	b	c
d	e	f
g	h	i

z

Filter

$=$

y				

I_f

Filtered image

$$y = 2e + f + h + 2i$$

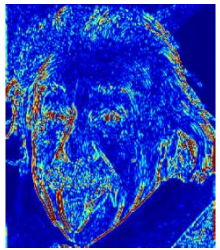
Correlation vs. Convolution

Image correlation:

$$I \otimes z = I_f$$

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

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

$\otimes$ **F** $=$ 

Correlation vs. Convolution

Image correlation:

$$I \otimes z = I_f$$

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

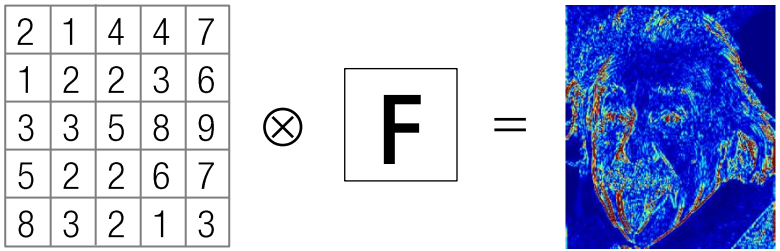
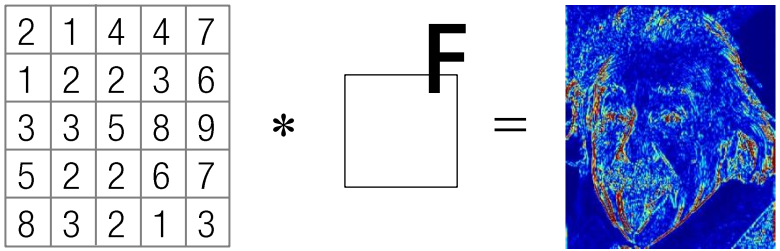


Image convolution:

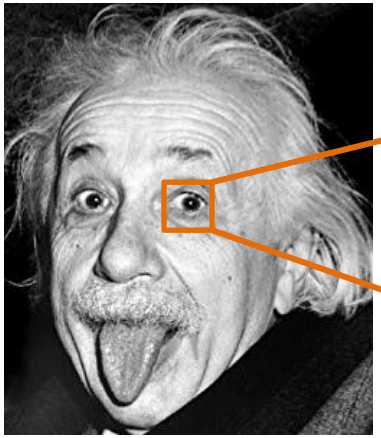
$$I * z = J$$

$$\sum_{k,l} I(i-k, j-l) z(k,l) = J(i,j)$$



Flip the filter in both dimension (bottom to top, right to left)

Correlation vs. Convolution



Correlation

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

$\otimes$

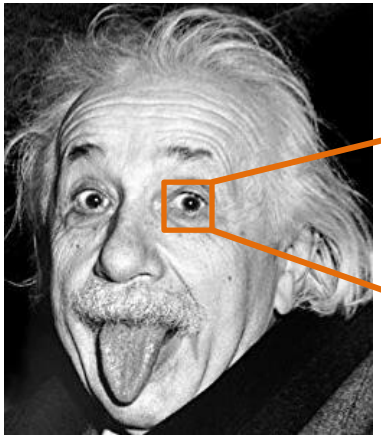
a	b	c
d	e	f
g	h	i

$=$

	y			

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

Correlation vs. Convolution



Correlation

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

$\otimes$

a	b	c
d	e	f
g	h	i

$=$

	y			

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

Convolution

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

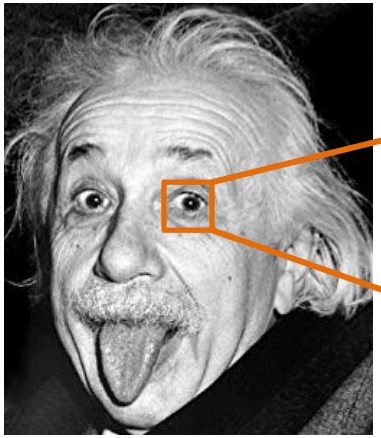
$*$

a	b	c
d	e	f
g	h	i

$=$

	z			

Correlation vs. Convolution



Correlation

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

$\otimes$

a	b	c
d	e	f
g	h	i

$=$

	y			

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

Convolution

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

$\otimes$

Flip the filter

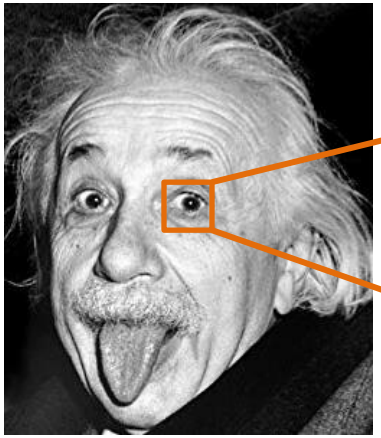
i	h	g
f	e	d
c	b	a

$=$

	z			

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

Correlation vs. Convolution



Correlation

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

$\otimes$

a	b	c
d	e	f
g	h	i

$=$

		y		

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

Convolution

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

$\otimes$

Flip the filter

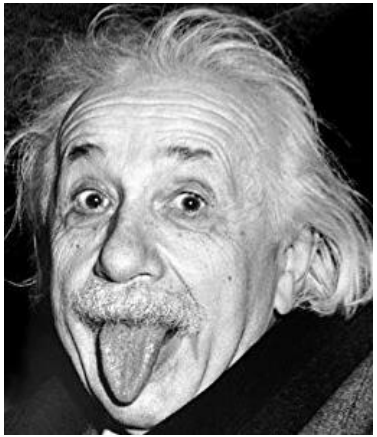
i	h	g
f	e	d
c	b	a

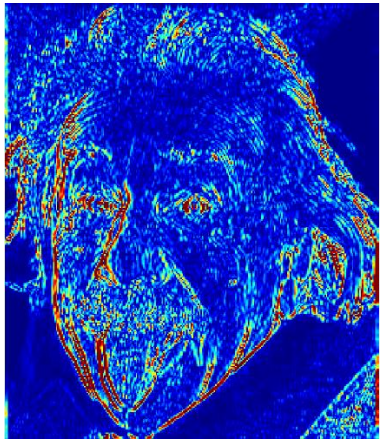
$=$

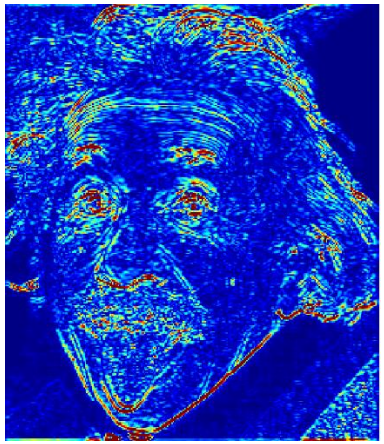
		z		

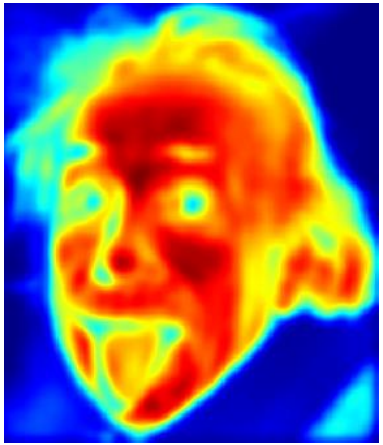
$$z = 2i + 2h + 3h + 3f + 5e + 8d + 2c + 3b + 6a$$

Convolution as a Feature Extraction

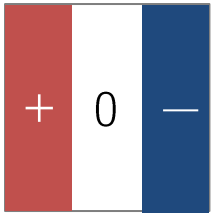


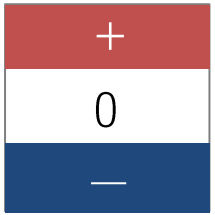


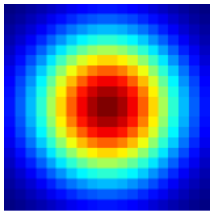




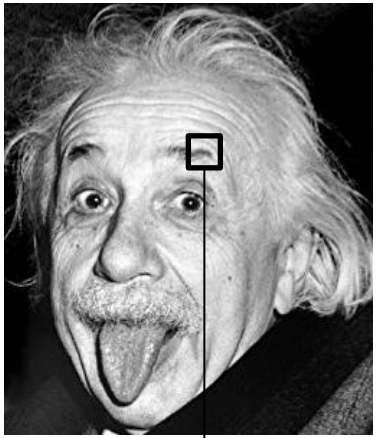
...

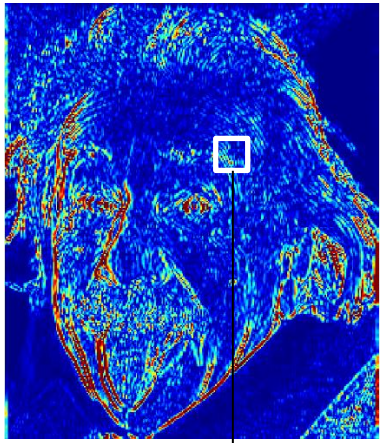


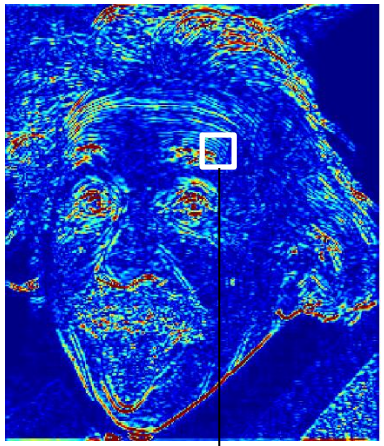


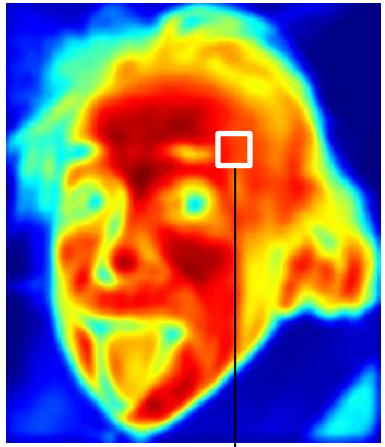


Convolution as a Feature Extraction









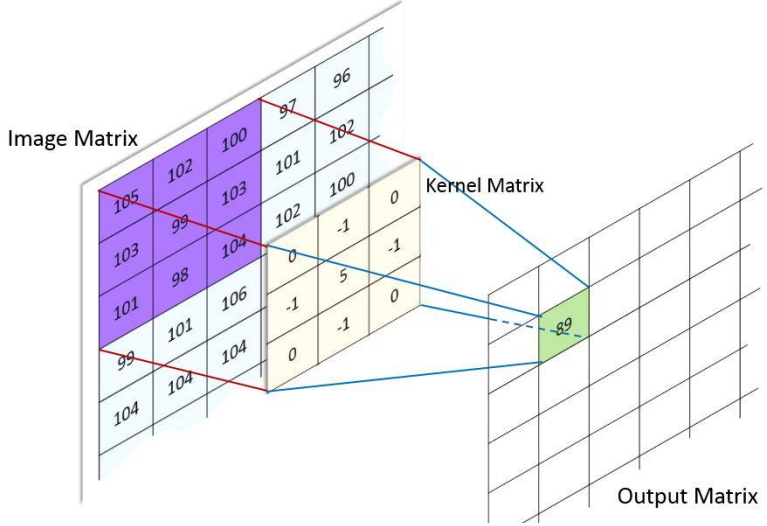
$I(x_1) = (212, 200, 221)$

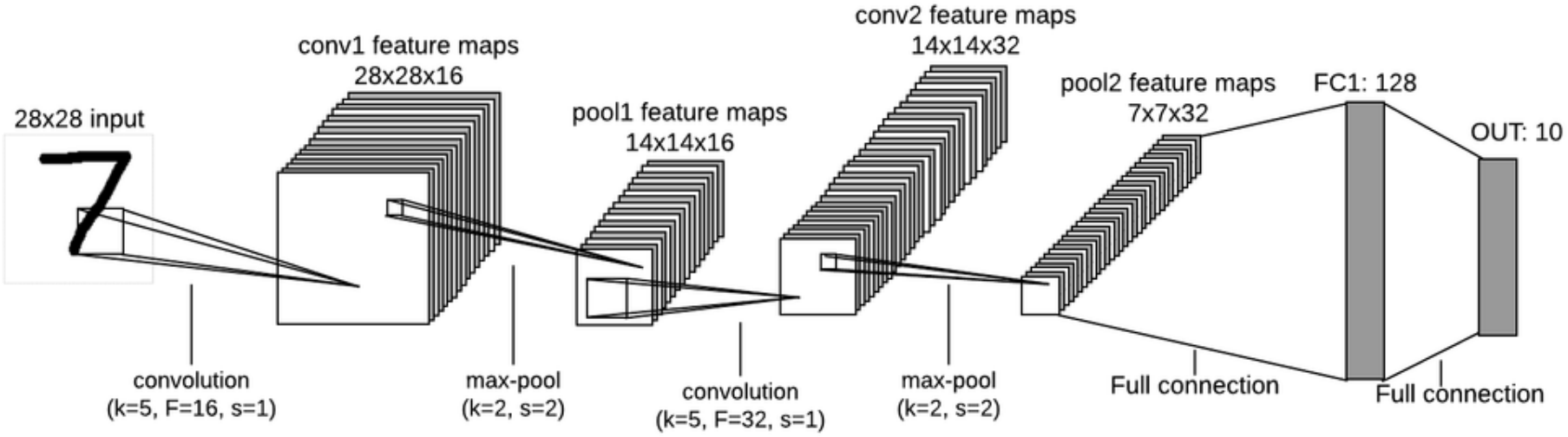
$f_1(x) = 0.3$

$f_2(x) = -0.1$

$f_3(x) = 0.9$

CONVOLUTIONAL NEURAL NETWORK

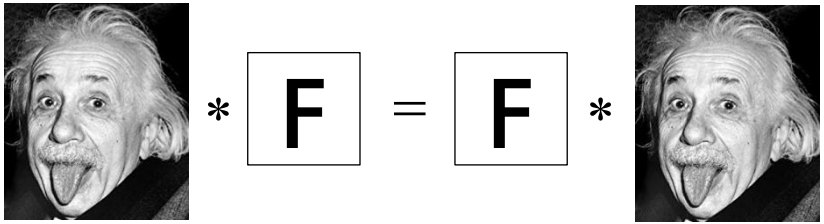




Properties of Convolution

Commutative:

$$f * g = g * f$$



A filter can be filtered by an image.

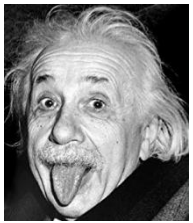
Properties of Convolution

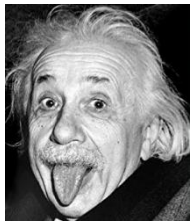
Commutative:

$$f * g = g * f$$

Associative:

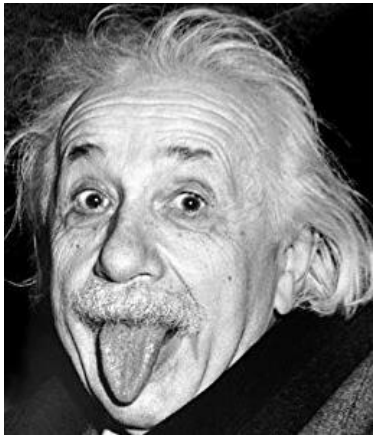
$$(f * g) * h = f * (g * h)$$

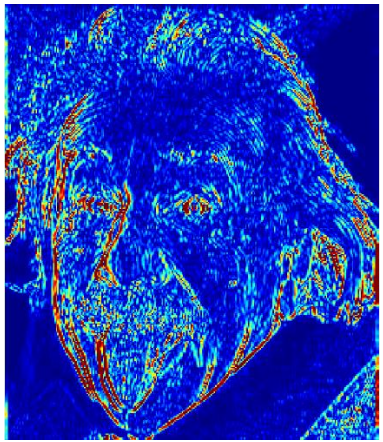
$$\left[\text{  } * \boxed{F} \right] * \boxed{G}$$

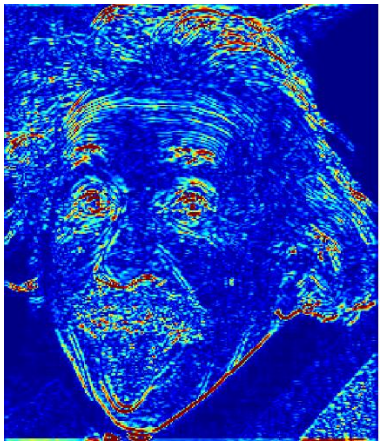
$$= \text{  } * \left[\boxed{F} * \boxed{G} \right]$$

A composition of convolutions can be pre-computed.

Recall: Edge Response







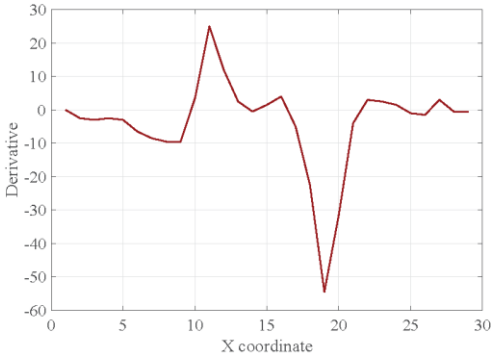
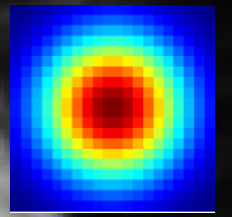


Image differentiation can be used for edge detection but it is VERY noisy.

How to detect edge reliably?

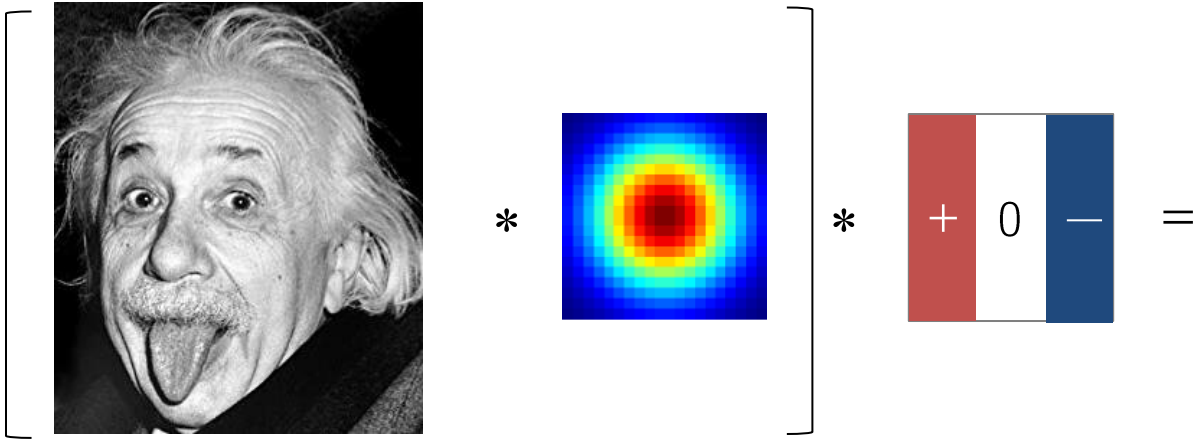
Recall: Gaussian Blurring~Denoising



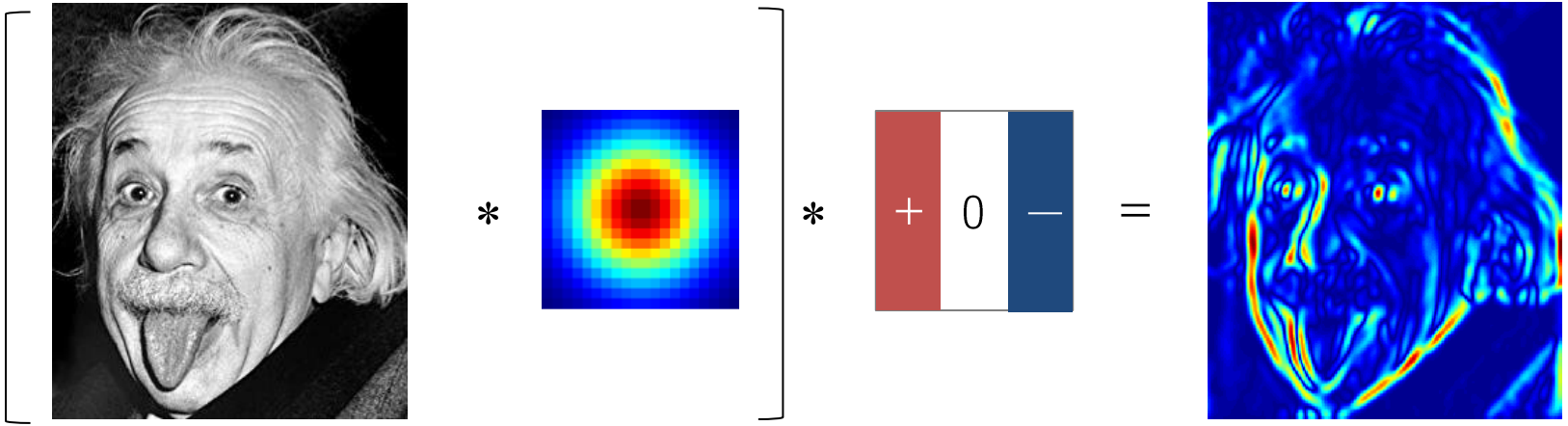
Noisy image

Denoised image

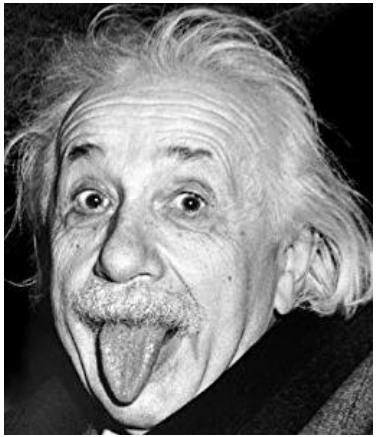
Strategy: Denoise and Differentiate



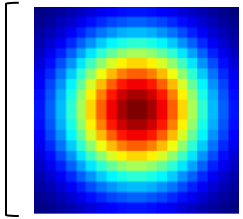
Strategy: Denoise and Differentiate



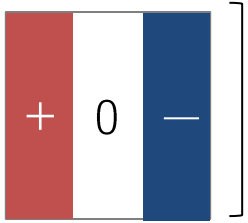
ASSOCIATIVITY



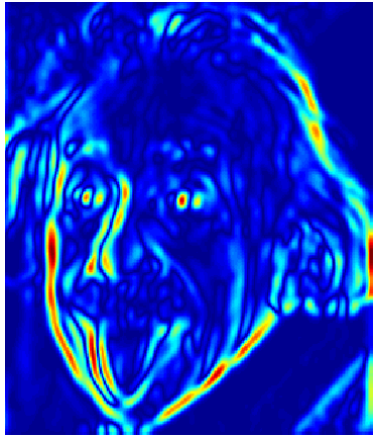
$*$



$*$

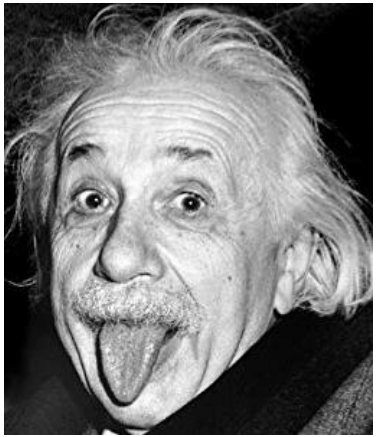


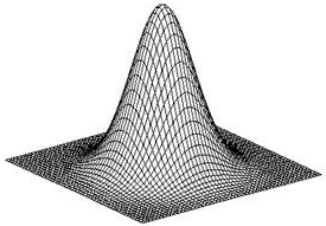
$=$



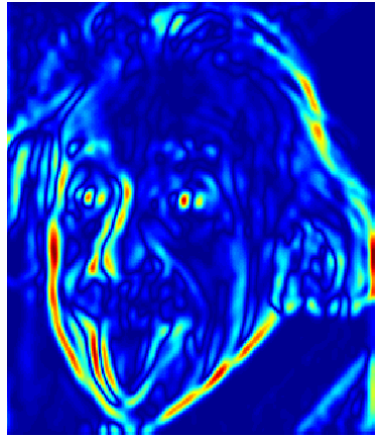
$$\frac{1}{2\pi\sigma^2} e^{-\frac{k^2+l^2}{2\sigma^2}} \qquad \frac{\partial}{\partial u}$$

COMMUTATIVITY



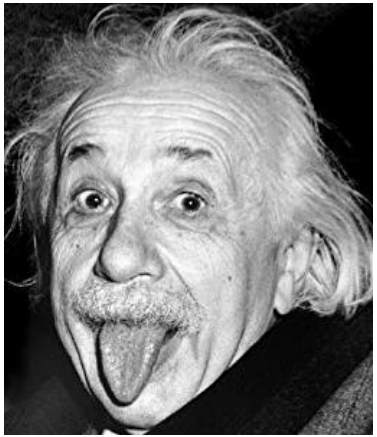


$$* \left[\begin{array}{|c|c|c|} \hline + & 0 & - \\ \hline \end{array} \right] * \left[\begin{array}{|c|} \hline \\ \hline \end{array} \right] =$$

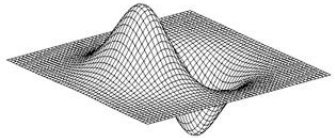


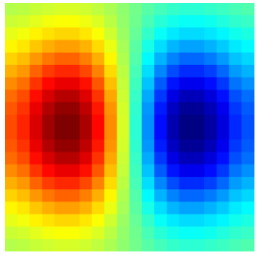
$$\frac{\partial}{\partial u} \qquad \frac{1}{2\pi\sigma^2} e^{-\frac{k^2+l^2}{2\sigma^2}}$$

SOBEL FILTER



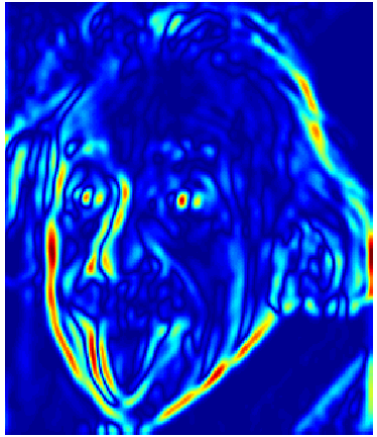
*





$$\frac{\partial}{\partial u}\frac{1}{2\pi\sigma^2}e^{-\frac{k^2+l^2}{2\sigma^2}}$$

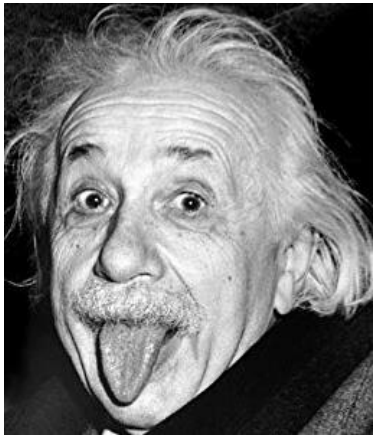
=



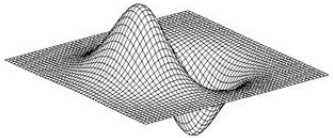
Sobel filter: derivative of Gaussian filter, e.g.,

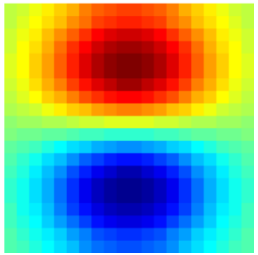
$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

SOBEL FILTER



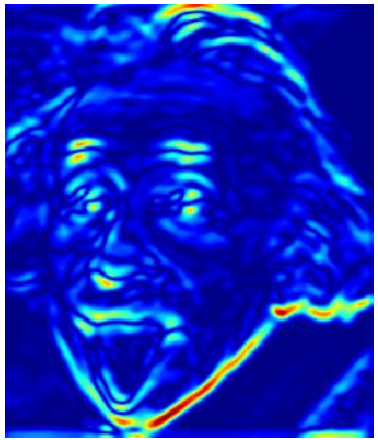
*





$$\frac{\partial}{\partial v}\frac{1}{2\pi\sigma^2}e^{-\frac{k^2+l^2}{2\sigma^2}}$$

=



Sobel filter: derivative of Gaussian filter, e.g.,

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$