EE655: Computer Vision & Deep Learning

Lecture 02

Koteswar Rao Jerripothula, PhD
Department of Electrical Engineering
IIT Kanpur

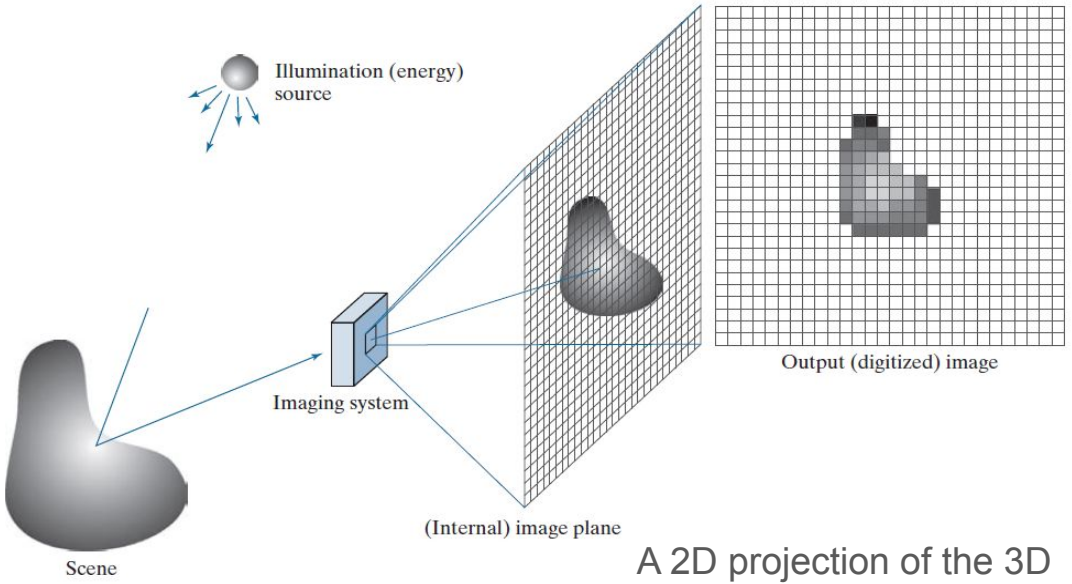
Lecture Outline

Image Formation 

Image Filtering

How is a digital image formed?

- A part of energy from an illumination (light) source is reflected by the scene.
- The imaging system collects the incoming energy and focuses it onto an image plane.
- A sensor array produces outputs proportional to the integral of the light received at each sensor.
- Digital and analog circuitry produces the output of final digital image.



A 2D projection of the 3D world is created and stored.

We will be referring to a “digital image” simply as an “image” in this course.

An image can be denoted as a 2D function:

Amount of light incident

$$f(x,y) = i(x,y)r(x,y)$$

Fraction of light reflected

where

$$0 \le i(x,y) < \infty$$

and

$$0 \le r(x,y) \le 1$$

Thus, $0 \le f(x,y) < \infty$

Typical values of illumination

	Illumination (lm/m^2)
Clear day	90,000
Cloudy Day	10,000
Full moon (clear evening)	0.1
Commercial Office	1,000

Typical values of reflectance

	Reflectance
Black velvet	0.01
Stainless Steel	0.65
Flat white wall paint	0.8
Silver plated metal	0.9
Snow	0.93

In we assume intensity of an image at coordinates (x,y) to be l

$$\ell = f(x, y)$$

$$L_{\min} \leq \ell \leq L_{\max}$$

$[L_{\min}, L_{\max}]$ is called the *intensity* (or *gray*) *scale*

$$L_{\min} = i_{\min} r_{\min} \text{ and } L_{\max} = i_{\max} r_{\max}$$

Real Numbers

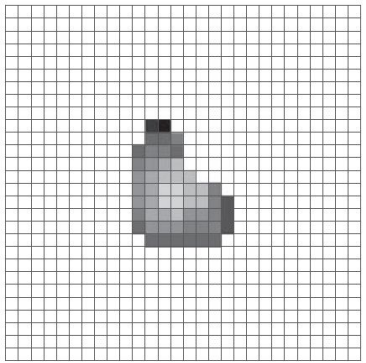
Whole Numbers

The intensity scale is shifted to either [0,1] or [0,C]

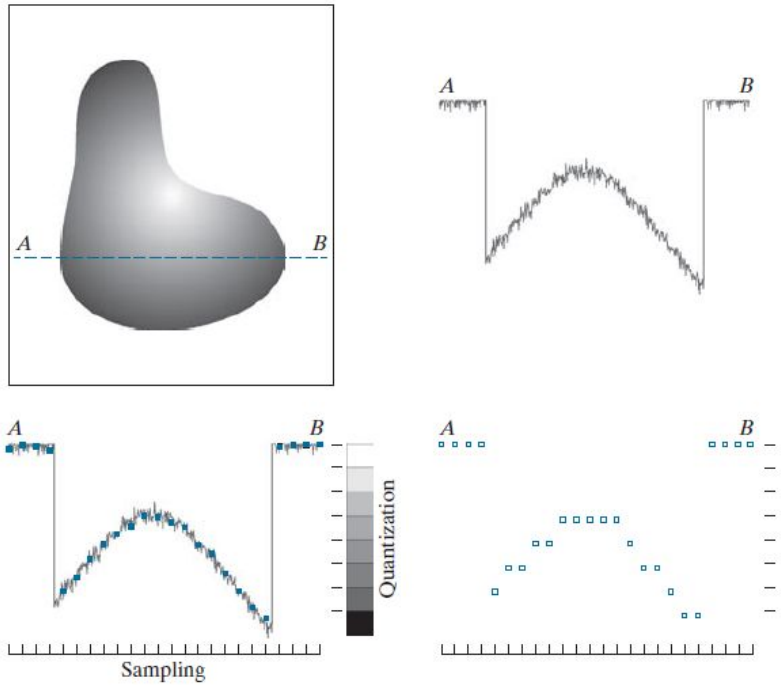
$\ell = 0$ represents Pure Black

$\ell = 1$ (or C) represents Pure White

Any value in-between represents a particular shade of gray



Quantization & Sampling



Quantization: Digitizing Amplitude

Sampling: Digitizing Coordinate Values

Pixels

- Camera stores the projection in the form of an 2D-array of pixels.
- A pixel is a tiny dot or square of color that is part of a digital image.
- A digital image (picture) consists of millions of these pixels or picture elements, which store color values.
- The word Pixel is a twist on the words "picture" and "element" combined.

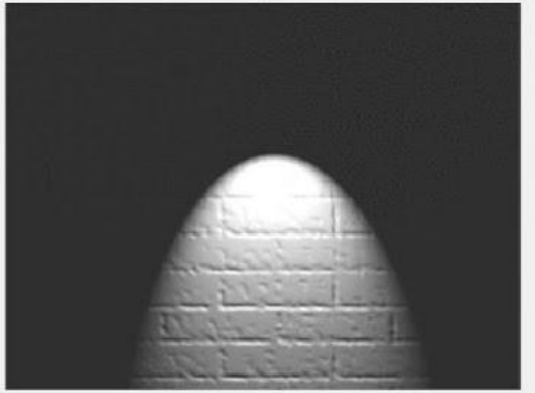
The more the pixels, the clearer it looks.

What all a pixel can store:

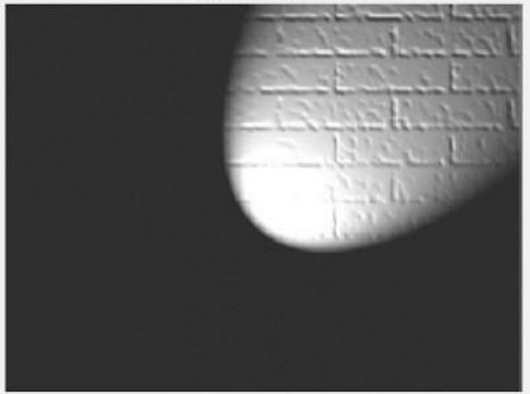
- A binary number to form a logical image C=1
- An 8-bit integer to form a grayscale image C=255
- Three 8-bit integers to form a color image

RGB Image

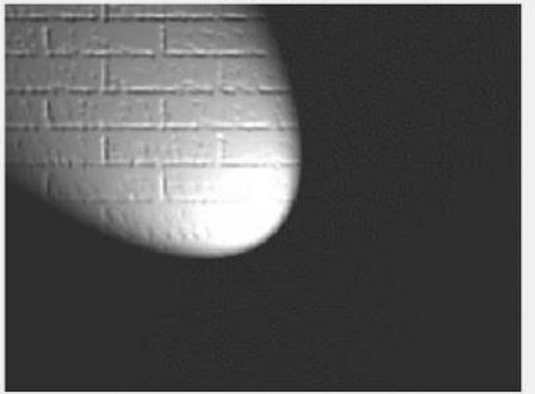
Red Channel



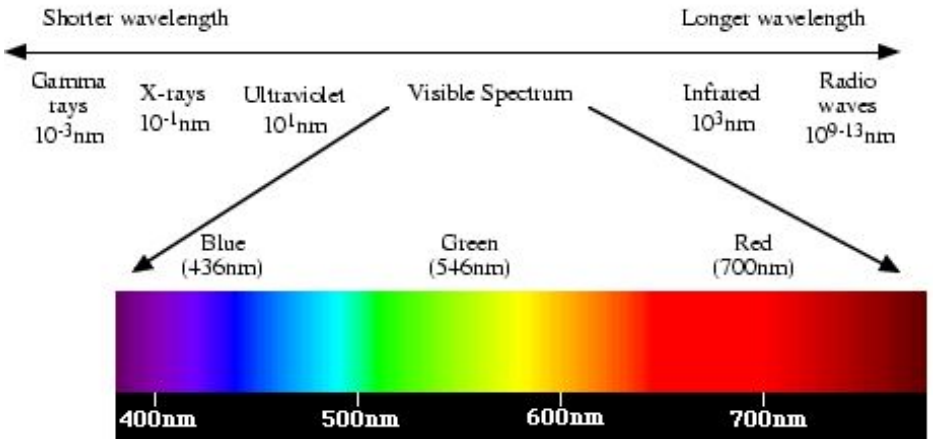
Green Channel



Blue Channel



The color is determined using the different wavelengths present in the light received by the sensor.



Lecture Outline

Image Formation

Image Filtering 

IMAGE FILTERING

$$f(I) = I_f$$

IMAGE FILTERING

$$f(I) = I_f$$

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

$$f(I(x_1)) = f(212,200,221)$$
$$= (242,152,135)$$

IMAGE FILTERING (PIXEL-WISE FILTERING)

$I_f = I + 100$

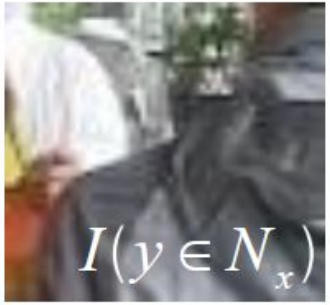
$I_f = I - 100$

$I_f = 255 - I$

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

IMAGE SPATIAL FILTERING

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



Local patch around x

N_x : Neighboring pixels of x

IMAGE SPATIAL FILTERING

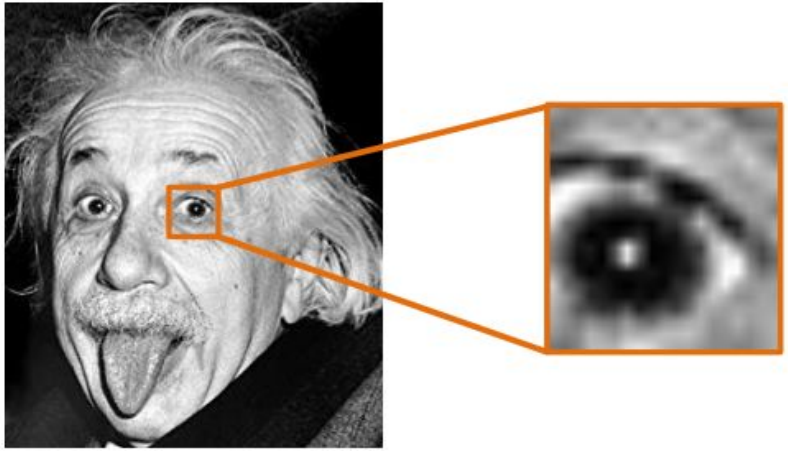
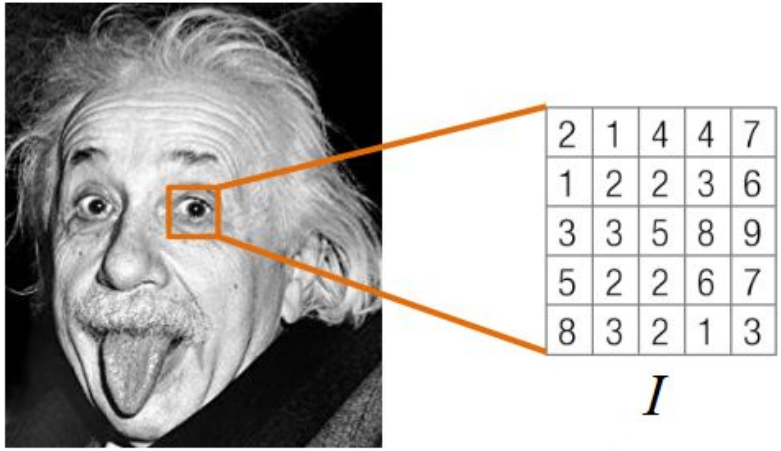


IMAGE SPATIAL FILTERING



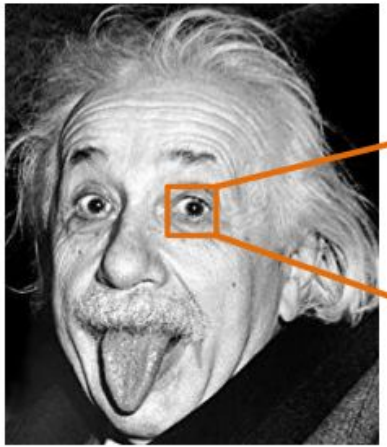
2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

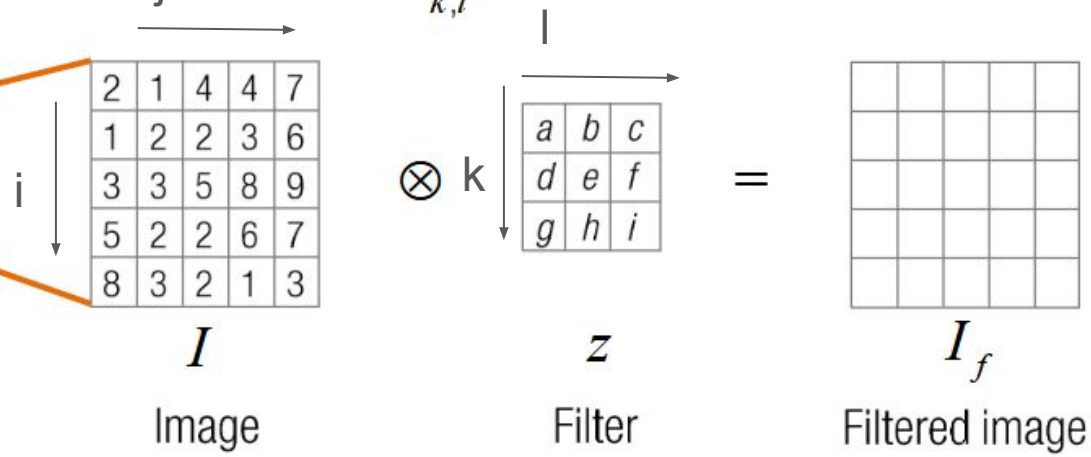
I

Image

IMAGE SPATIAL FILTERING

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

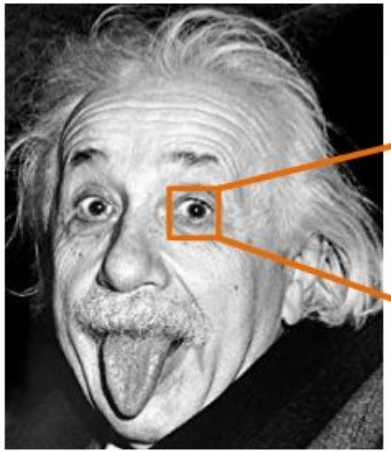




Range of i & j is [1,5]
Range of k & l is [-1,1]

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

$\otimes$

a	*b*	*c*
d	*e*	*f*
g	*h*	*i*

$=$

	y			

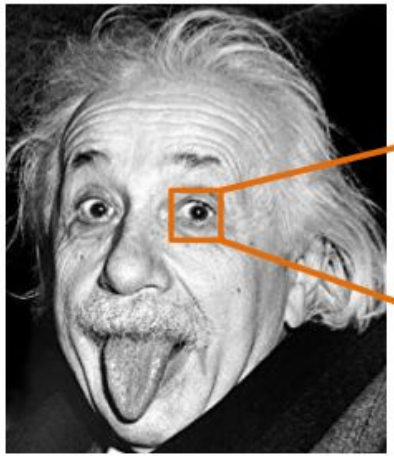
Image Filter Filtered image

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

@ i=2, j=2

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

$\otimes$

a	b	c
d	e	f
g	h	i

$=$

		y		

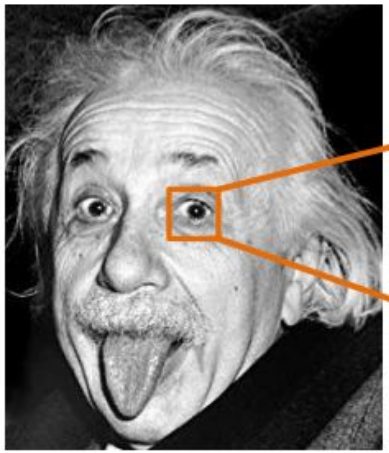
Image Filter Filtered image

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

@ i=2, j=3

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

$\otimes$

a	b	c
d	e	f
g	h	i

$=$

			y	

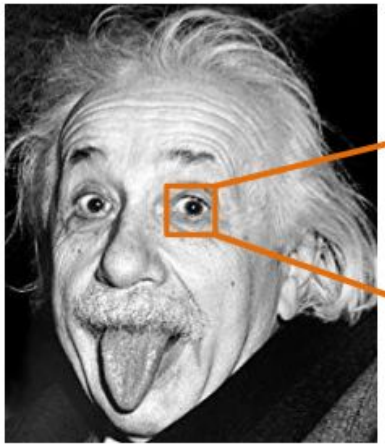
Image Filter Filtered image

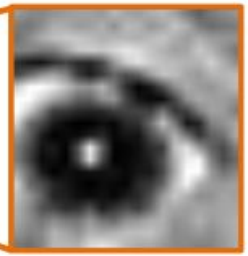
$$y = 5a + 8b + 9c + 2d + 6e + 7f + 2g + h + 3i$$

@ i=4, j=4

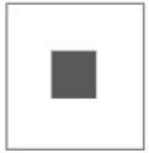
IDENTITY

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





$\otimes$



$=$

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

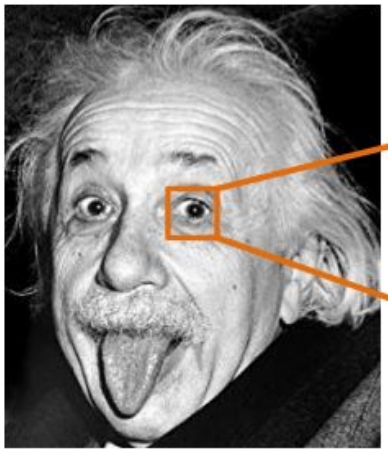
$\otimes$

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

⊗

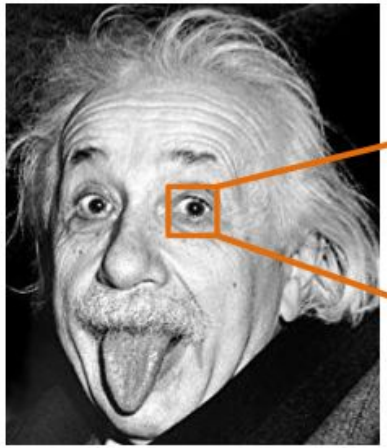
0	0	0
0	1	0
0	0	0

=

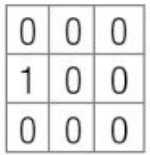
	2	2	3	
	3	5	8	
	2	2	6	

SHIFTING

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



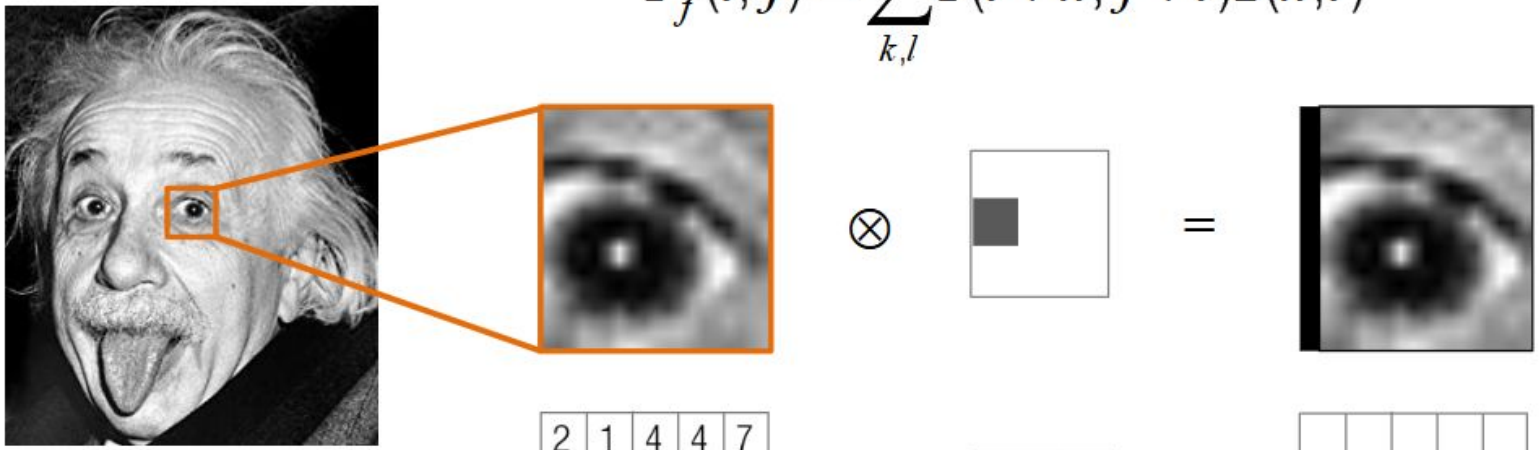
2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3



0	0	0
1	0	0
0	0	0

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

$\otimes$

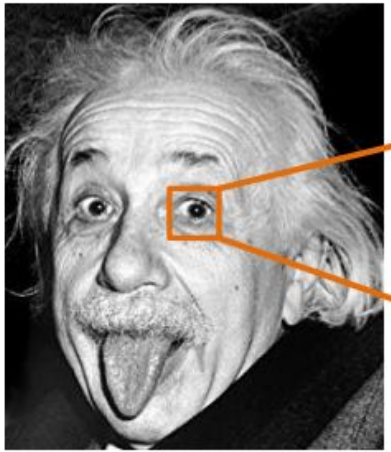
0	0	0
1	0	0
0	0	0

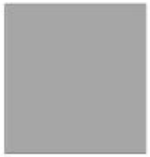
$=$

	1	2	2	
	3	3	5	
	5	2	2	

Box Filter (Moving Averaging)

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



$\otimes$ 

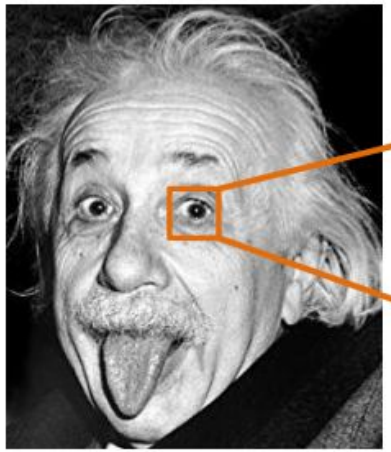
2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

$\otimes \frac{1}{9}$

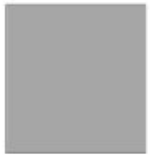
1	1	1
1	1	1
1	1	1

Box Filter (Moving Averaging)

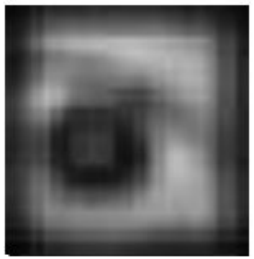
$$I_f(i,j) = \sum_{k,l} I(i+k, j+l) z(k,l)$$



$\otimes$



$=$



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

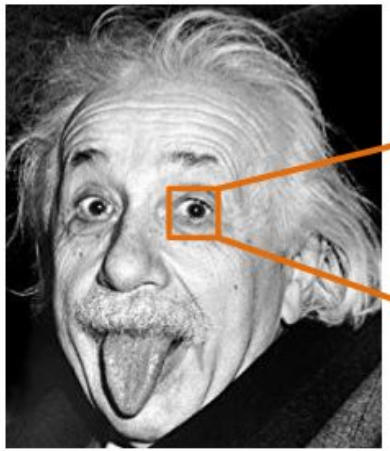
$\otimes$ $\frac{1}{9}$

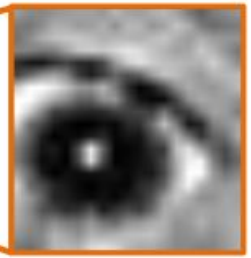
1	1	1
1	1	1
1	1	1

Average

DIRECTIONAL BLUR

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





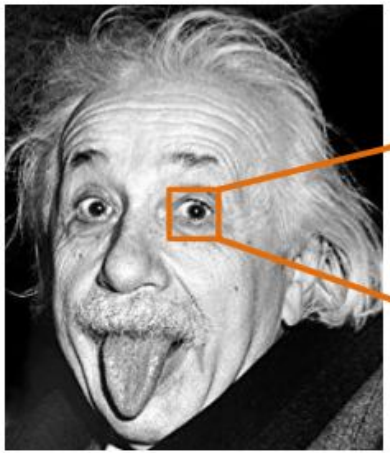
2	1	4	4	7
1	2	2	3	6
3	3	5	8	9
5	2	2	6	7
8	3	2	1	3

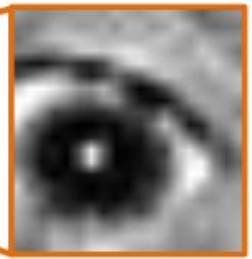
$\otimes$ $\frac{1}{3}$

0	0	0
1	1	1
0	0	0

DIRECTIONAL BLUR

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





$\otimes$

$=$

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

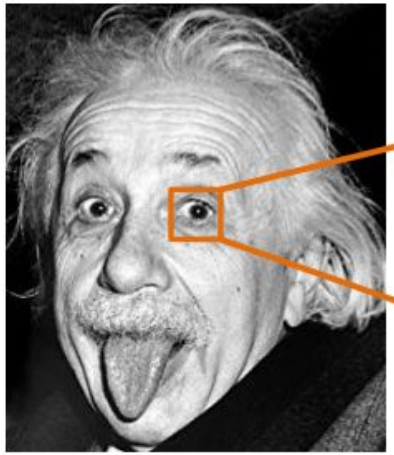
$\otimes$ $\frac{1}{3}$

0	0	0
1	1	1
0	0	0

Horizontal blur

DIRECTIONAL BLUR

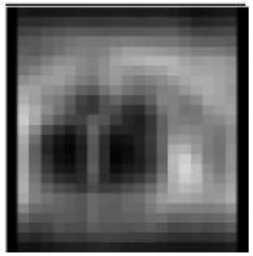
$$I_f(i,j) = \sum_{k,l} I(i+k, j+l) z(k,l)$$



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

$\otimes$

$=$

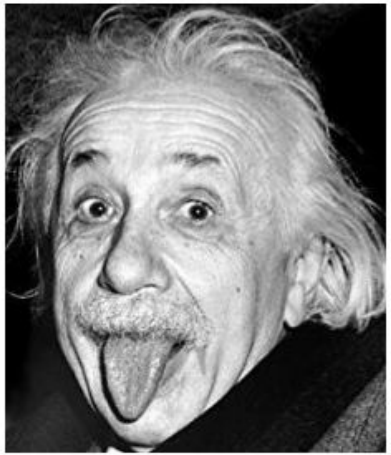


$\otimes \frac{1}{3}$

0	1	0
0	1	0
0	1	0

Vertical blur

Box Filter (Moving Averaging)

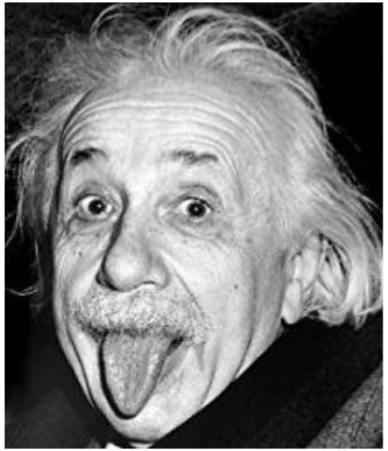


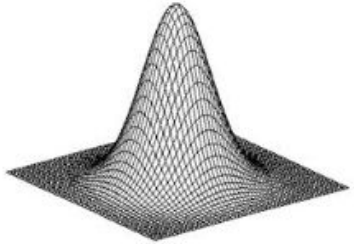
$\otimes$

$=$

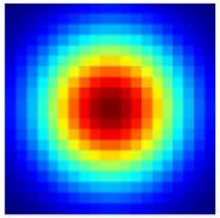
Pixel averaging

GAUSSIAN BLURRING (MOVING WEIGHTED AVERAGE)





$\otimes$

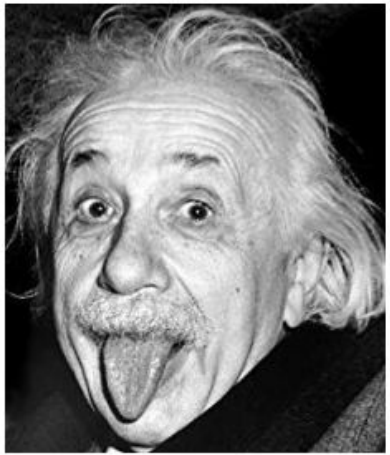


$=$

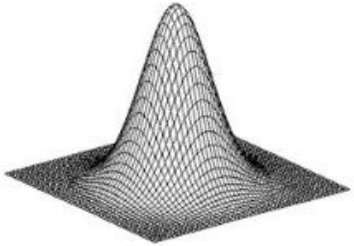
$$z(k,l) = \frac{1}{2\pi\sigma^2} e^{-\frac{k^2+l^2}{2\sigma^2}}$$

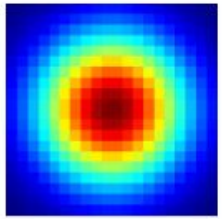
HIGH
LOW

GAUSSIAN BLURRING (MOVING WEIGHTED AVERAGE)



$\otimes$



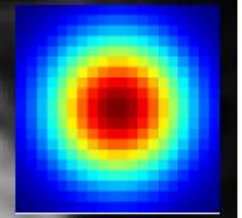


$=$

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

Gaussian blur

Gaussian Blurring~Denoising

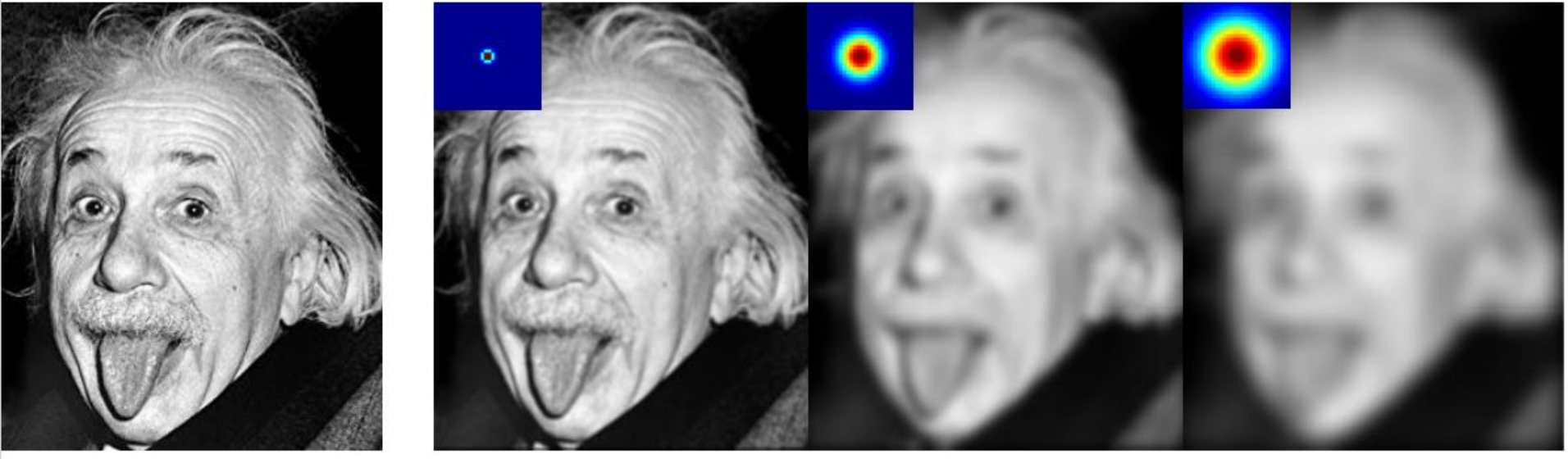


Noisy image

Denoised image

GAUSSIAN BLURRING

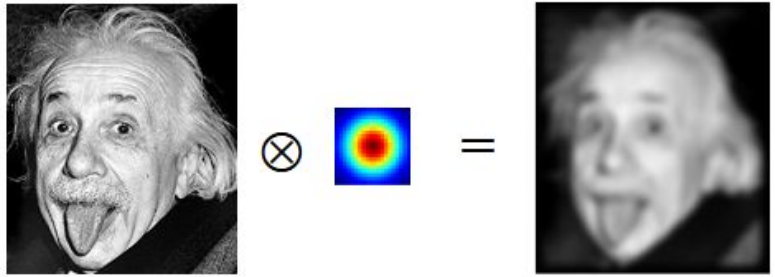
$$z(k,l) = \frac{1}{2\pi\sigma^2} e^{-\frac{k^2+l^2}{2\sigma^2}}$$



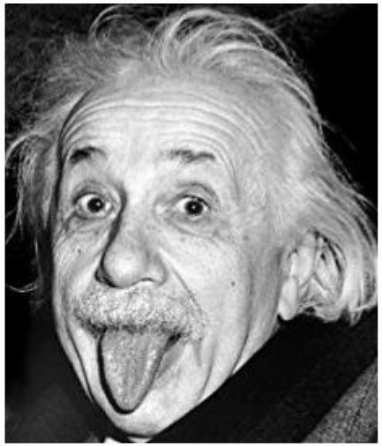
$\sigma = 1$ $\sigma = 4$ $\sigma = 7$

IMAGE SPATIAL FILTERING

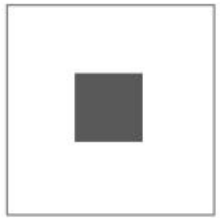
```
function im_f = Filtering(im, filter)

% filtered image initialization
im_f = zeros(size(im));
center_k = floor(size(filter,1)/2)+1;
center_l = floor(size(filter,2)/2)+1;
for i = 1 : size(im,1)
    for j = 1 : size(im,2)
        % filtering
        v = 0;
        for k = 1 : size(filter,1)
            for l = 1 : size(filter,2)
                i1 = i + k - center_k;
                j1 = j + l - center_l;
                if i1 <= 0 || i1 > size(im_f,1) || j1 <= 0 || j1 > size(im_f,2)
                    continue;
                end
                v = v + im(i1,j1)*filter(k,l);
            end
        end
        im_f(i,j) = v;
    end
end
```



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



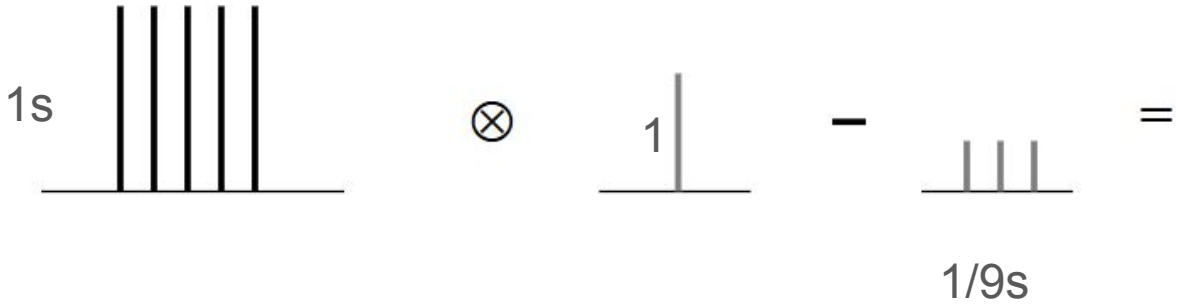
$\otimes$



$-$

$=$

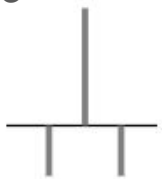
Equivalence in 1D



1s

$\otimes$

8/9



-1/9s

=

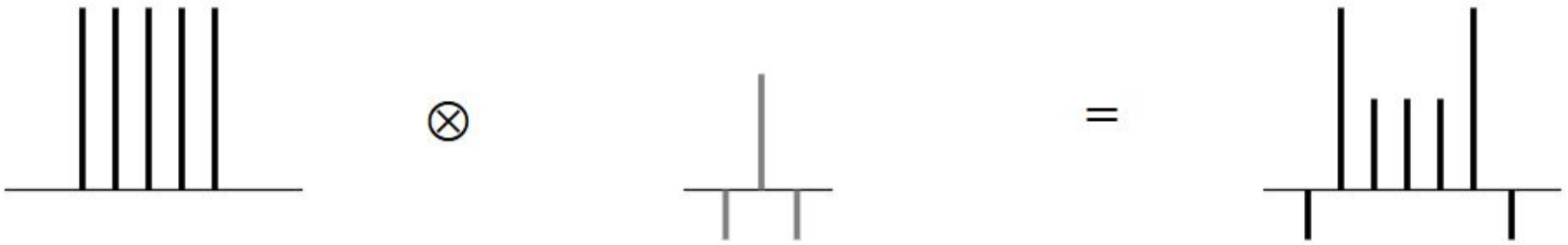
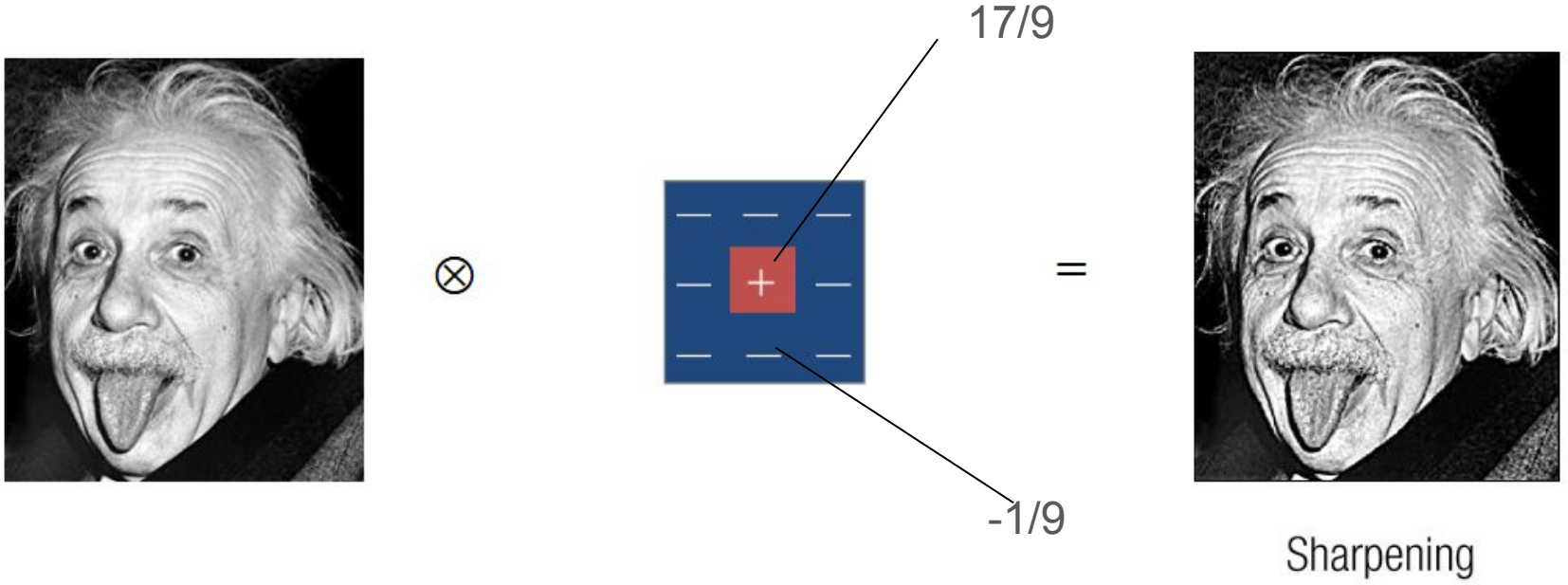
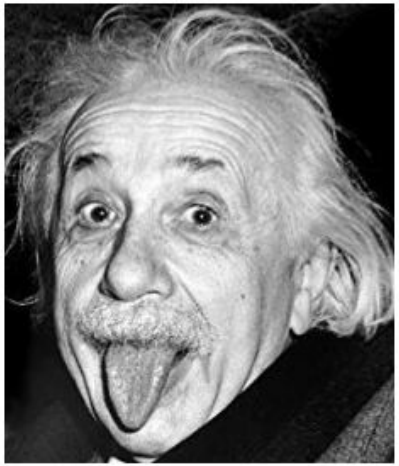

High response at edge

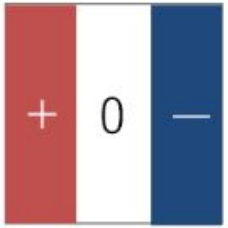
IMAGE SHARPENING



[**Note:** Need to add the original image to get the sharpened image, so the filter is 2*identity - box]



$\otimes$

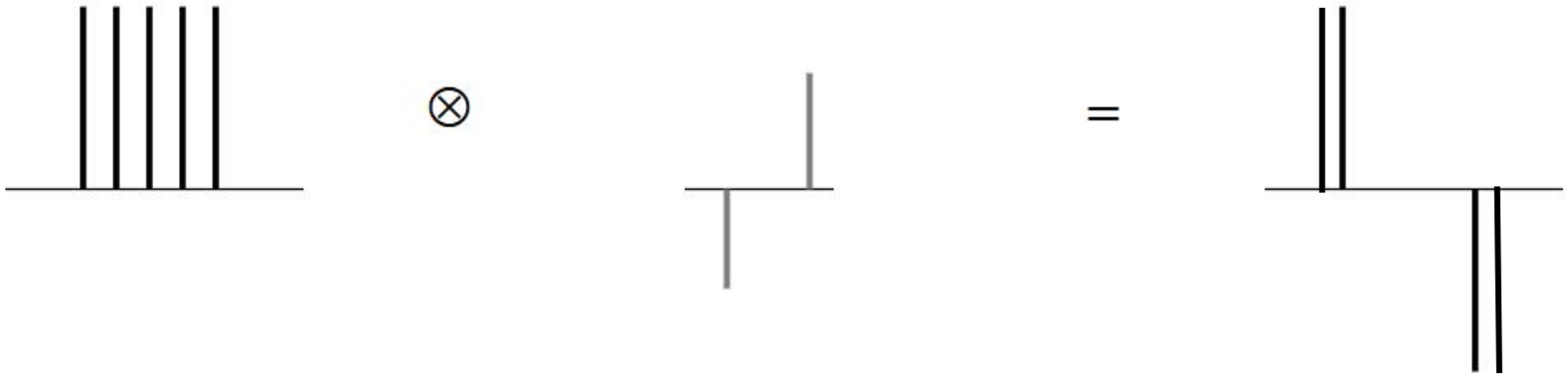


+1 -1

=

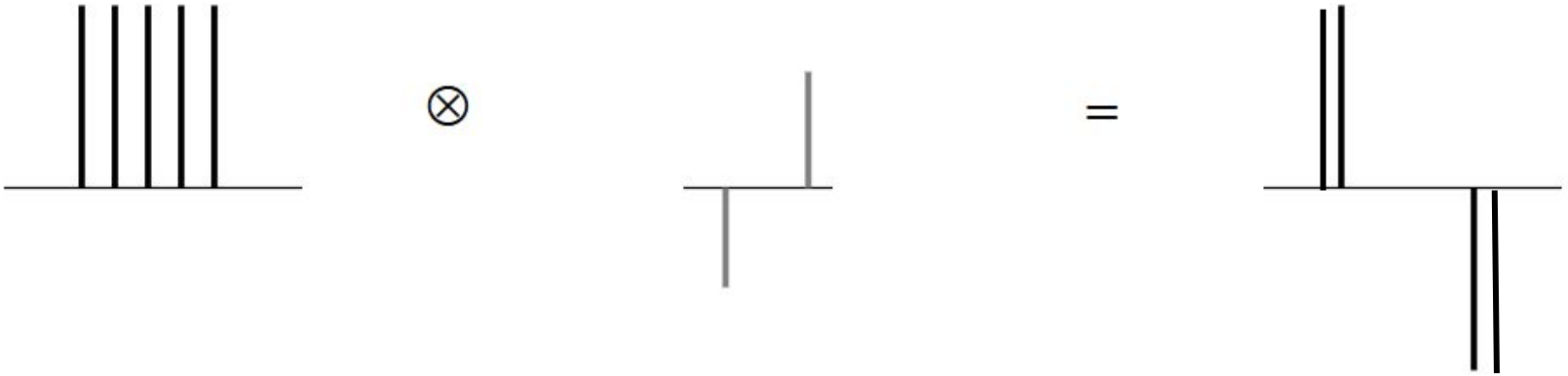
DIFFERENTIATION

DIFFERENTIATION



+ve or -ve spikes are observed at edges

DIFFERENTIATION

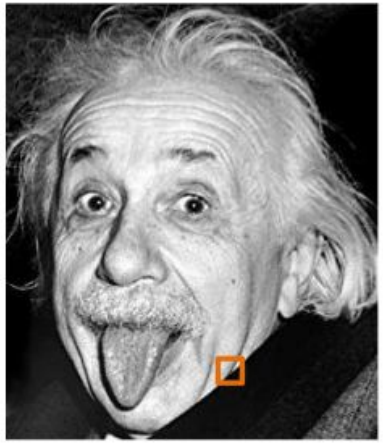


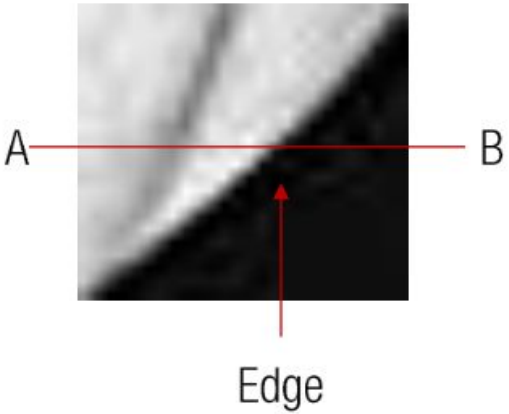
$$\frac{df}{du} = \lim_{h \to 0} \frac{f(u+h) - f(u-h)}{2h} \longrightarrow I \otimes z = \frac{\partial I}{\partial u}$$

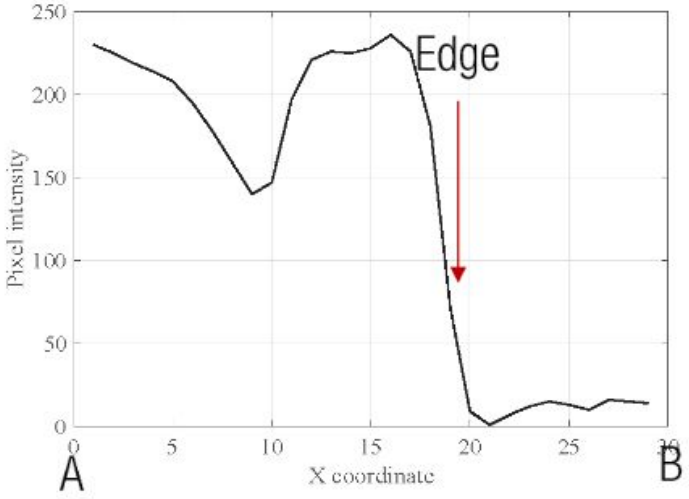
In discrete domain such as an image, h=1 (the smallest change)

So, 0.5*f(u+1)+0*f(u)+(-0.5)*f(u-1). The filter turns out to be [-0.5,0,0.5]

DIFFERENTIATION







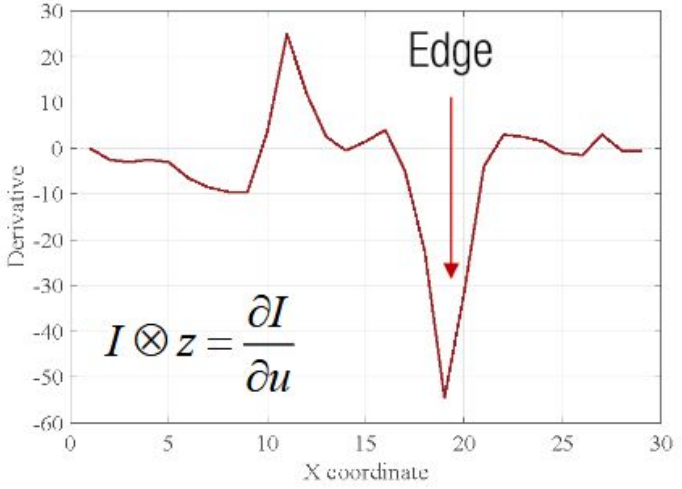
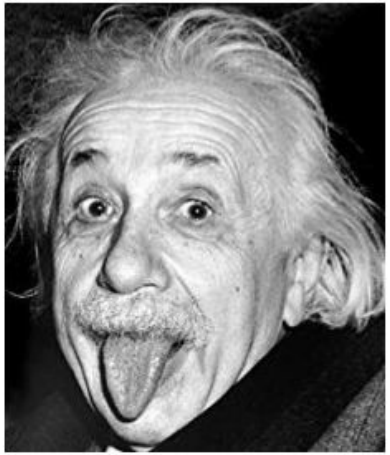
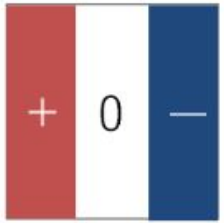


IMAGE DIFFERENTIATION

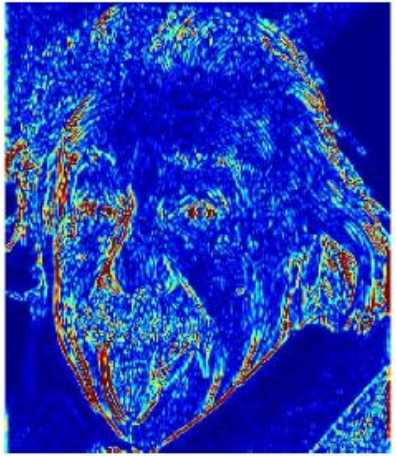


$\otimes$



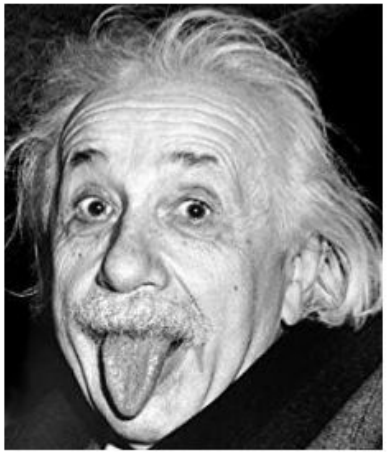
$$I \otimes z = \frac{\partial I}{\partial u}$$

=



Differentiation

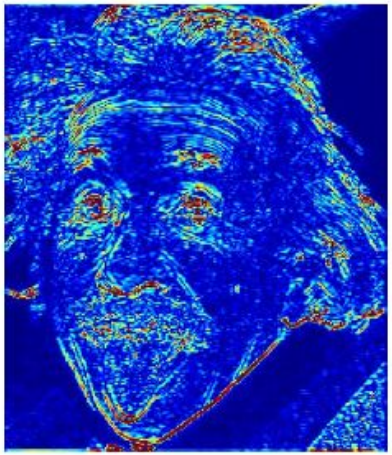
IMAGE DIFFERENTIATION



$\otimes$

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

=



Differentiation