Correlation and Convolution

❖Correlation and Convolution are basic operations that we will perform to extract information from images.

❖They are simple, they can be analyzed and understood very well, and they are also easy to implement and can be computed very efficiently.

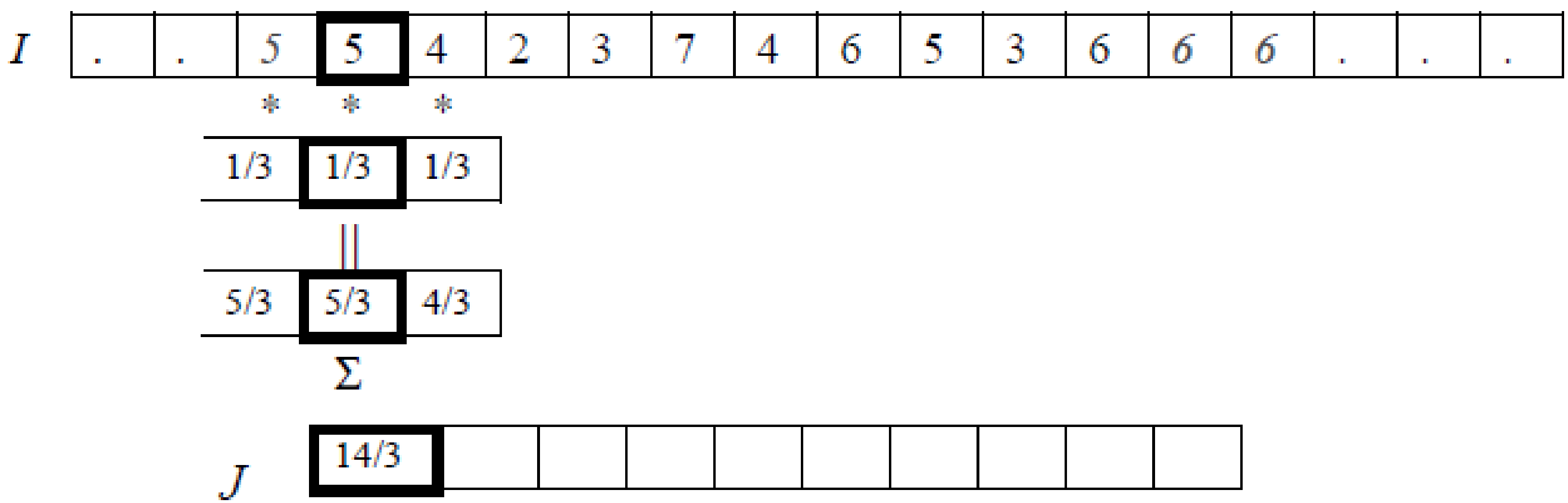
❖These operations have two key features: they are *shift-invariant, and they are linear.*

❖Shift-invariant means that we perform the same operation at every point in the image.

❖Linear means that this operation is linear, that is, we replace every pixel with a linear combination of its neighbors

One of the simplest operations that we can perform with correlation is local averaging.

5	4	2	3	7	4	6	5	3	6



Source : David Jacobs, Correlation and Convolution, http://www.cs.umd.edu/~djacobs/CMSC426/Convolution.pdf

$$F \circ I(x) = \sum_{i=-N}^{N} F(i) I(x+i)$$

$$F \circ I(x,y) = \sum_{j=-N}^{N} \sum_{i=-N}^{N} F(i,j) I(x+i, y+j)$$

3	7	5

3	2	4	1	3	8	4	0	3	8	0	7	7	7	1	2

40	43	39	34	64	85	52	27	61	65	59	84	105	75	38	27

Source : David Jacobs, Correlation and Convolution, http://www.cs.umd.edu/~djacobs/CMSC426/Convolution.pdf

$$\frac{\sum_{i=-N}^{N}\left(F(i)I(x+i)\right)}{\sqrt{\sum_{i=-N}^{N}\left(I(x+i)\right)^2}\sqrt{\sum_{i=-N}^{N}\left(F(i)\right)^2}}.$$

3	7	5

3	2	4	1	3	8	4	0	3	8	0	7	7	7	1	2

.946	.877	.934	.73	.81	.989	.64	.59	.78	.835	.61	.931	.95	.83	.57	.988

Source : David Jacobs, Correlation and Convolution, http://www.cs.umd.edu/~djacobs/CMSC426/Convolution.pdf

The Filter Matrix

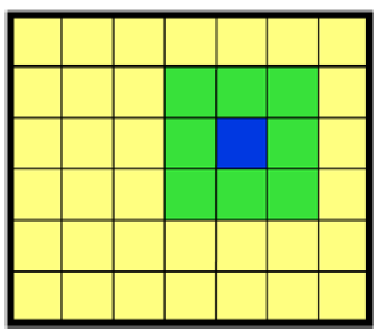
$$I'(u,v) \leftarrow \frac{p_0+p_1+p_2+p_3+p_4+p_5+p_6+p_7+p_8}{9}$$

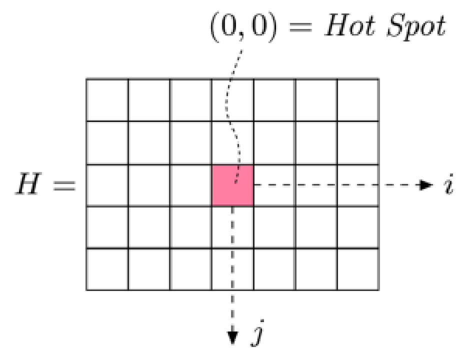
$$\begin{aligned} I'(u,v) \leftarrow \tfrac{1}{9}\cdot [\, & I(u-1,v-1) && + I(u,v-1) && + I(u+1,v-1) && + \\ & I(u-1,v) && + I(u,v) && + I(u+1,v) && + \\ & I(u-1,v+1) && + I(u,v+1) && + I(u+1,v+1) \,] \end{aligned}$$

$$H(i,j) = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

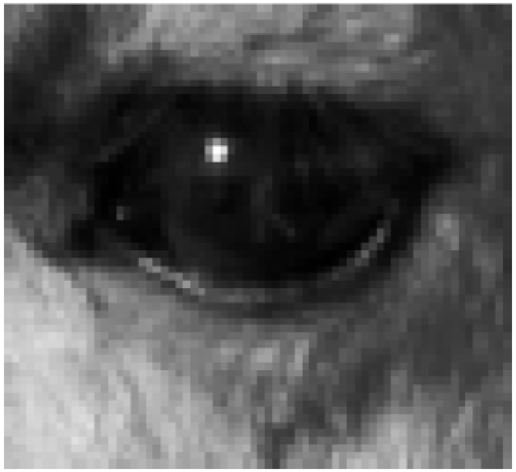
Filter operation can be expressed as a matrix
Example: averaging filter

Filter matrix also called filter mask ***H(i,j)***

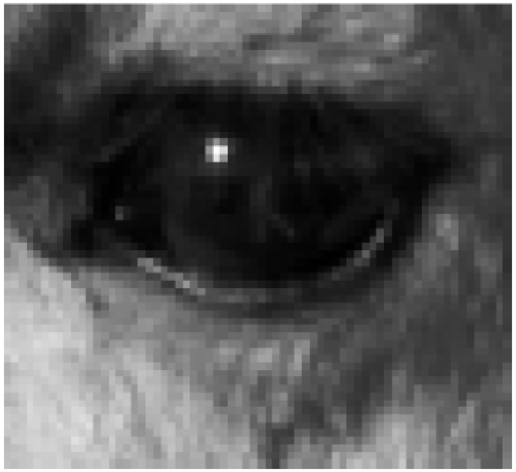




Example: What does this Filter Do?

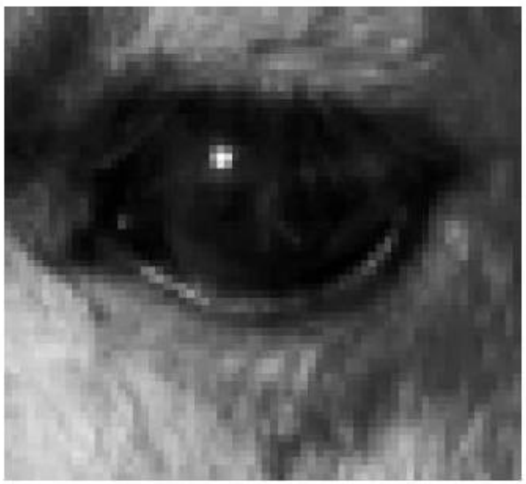


0	0	0
0	1	0
0	0	0

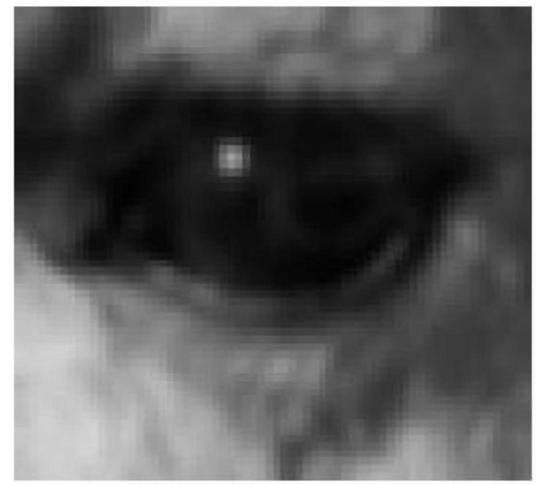


Identity function (leaves image alone)

What Does this Filter Do?

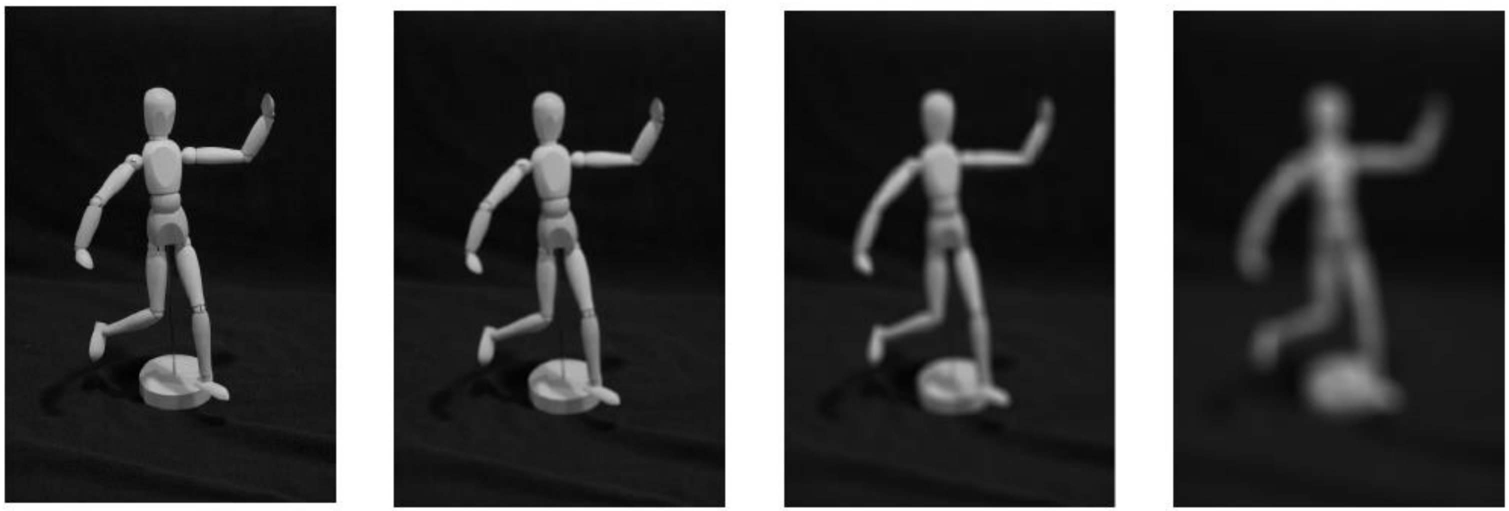


$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$



Mean (averages neighborhood)

Mean Filters: Effect of Filter Size



Original　　7×7　　15×15　　41×41

Separable Filters

Generally, 2D correlation is more expensive than 1D correlation because the sizes of the filters we use are larger.

If our filter is NxN in size, and our image contains MxM pixels, then the total number of multiplications we must perform is N^2M^2.

With an important class of filters, we can save on this computation.

$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & 0 \end{bmatrix} \circ \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 \end{bmatrix}$$

Convolution

$$F * I(x) = \sum_{i=-N}^{N} F(i) I(x-i)$$

$$F * I(x,y) = \sum_{j=-N}^{N} \sum_{i=-N}^{N} F(i,j) I(x-i, y-j)$$

Convolution is just like correlation, except that we flip over the filter before correlating.

Correlation and convolution are identical when the filter is symmetric.

The key difference between the two is that convolution is associative.

$$F*(G*I) = (F*G)*I.$$

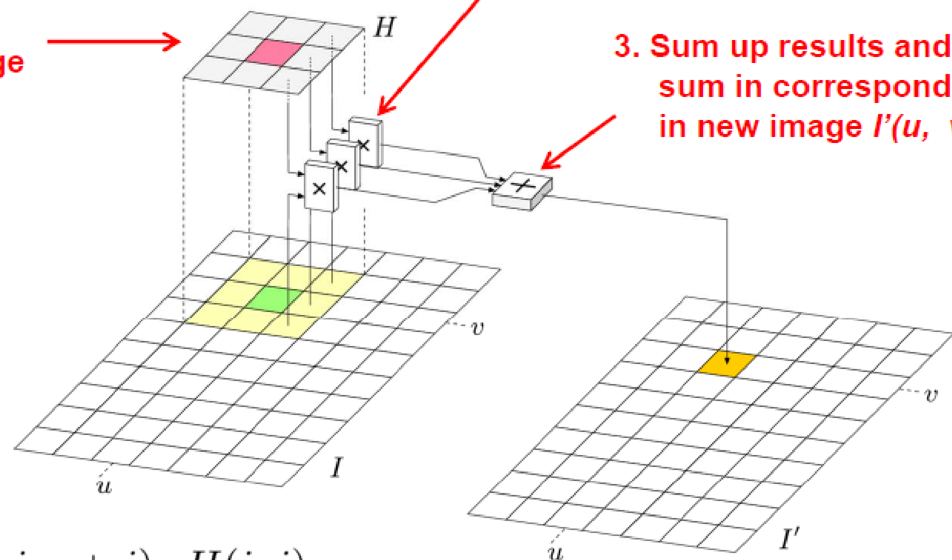
Applying Linear Filters: Convolution

For each image position *I(u,v)*:

1. Move filter matrix *H* over image such that *H(0,0)* coincides with current image position *(u,v)*

2. Multiply all filter coefficients *H(i,j)* with corresponding pixel *I(u + i, v + j)*

3. Sum up results and store sum in corresponding position in new image *I'(u, v)*



Stated formally:

$$I'(u,v) \leftarrow \sum_{(i,j)\in R_H} I(u+i, v+j) \cdot H(i,j)$$

R_H is set of all pixels Covered by filter.
For 3x3 filter, this is:

$$I'(u,v) \leftarrow \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} I(u+i, v+j) \cdot H(i,j)$$

Weighted Smoothing Filters

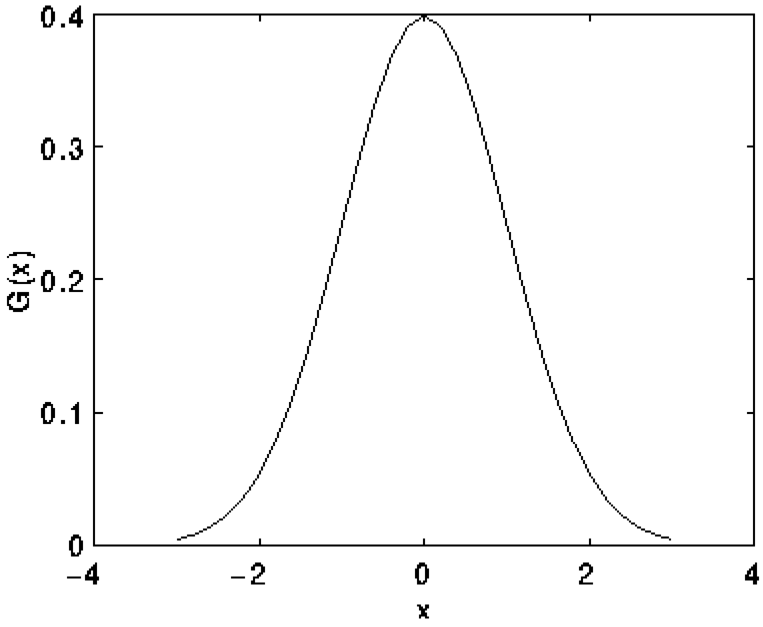
- More effective smoothing filters can be generated by allowing different pixels in the neighbourhood different weights in the averaging function
  - Pixels closer to central pixel more important
  - Often referred to as a *weighted averaging*

$^{1}/_{16}$	$^{2}/_{16}$	$^{1}/_{16}$
$^{2}/_{16}$	$^{4}/_{16}$	$^{2}/_{16}$
$^{1}/_{16}$	$^{2}/_{16}$	$^{1}/_{16}$

Weighted averaging filter

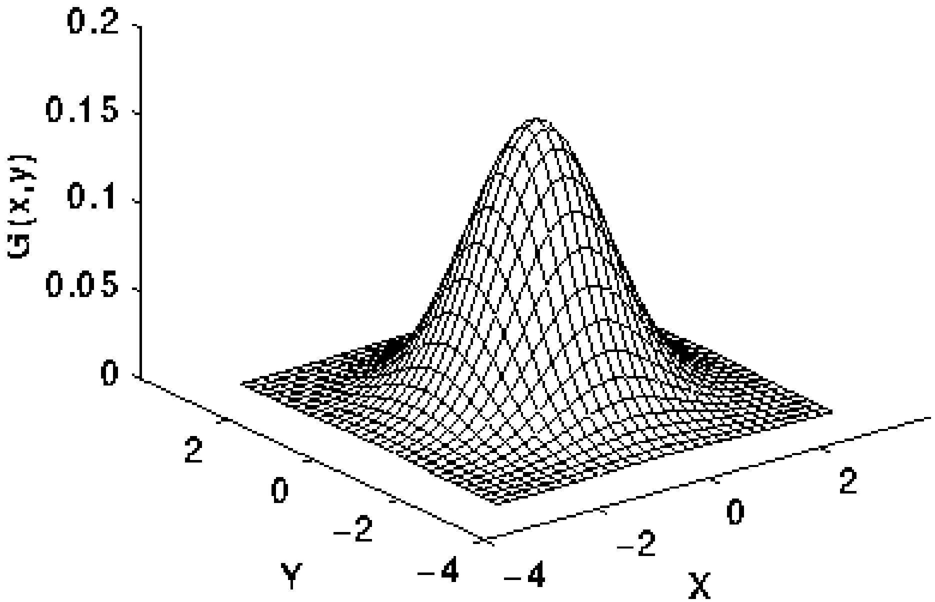
Gaussian Smoothing

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$



1-D Gaussian distribution with mean 0 and =1

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

2-D Gaussian distribution with mean (0,0) and =1

Properties of Convolution

- Commutativity

Same result if we convolve image with filter or vice versa

$$I * H = H * I$$

- Linearity

If image multiplied by scalar Result multiplied by same scalar

$$(s \cdot I) * H \;=\; I * (s \cdot H) \;=\; s \cdot (I * H)$$

$$(I_1 + I_2) * H \;=\; (I_1 * H) + (I_2 * H)$$

If 2 images added and convolve result with a kernel *H*, Same result if each image is convolved individually + added

(notice) $$(b + I) * H \;\neq\; b + (I * H)$$

- Associativity

$$A * (B * C) = (A * B) * C$$

Order of filter application irrelevant Any order, same result

Properties of Convolution

- Separability

$$H = H_1 * H_2 * \ldots * H_n$$

$$\begin{aligned} I * H &= I * (H_1 * H_2 * \ldots * H_n) \\ &= \left(\ldots ((I * H_1) * H_2) * \ldots * H_n \right) \end{aligned}$$

- If a kernel H can be separated into multiple smaller kernels

Applying smaller kernels H_1 H_2 … H_N H one by one computationally cheaper than apply 1 large kernel H

$$H = H_1 * H_2 * \ldots * H_n$$

Computationally More expensive

Computationally Cheaper

Separability in *x* and *y*

- Sometimes we can separate a kernel into **"vertical"** and **"horizontal"** components
- Consider the kernels

$$H_x = [1\ 1\ 1\ 1\ 1], \quad \text{and} \quad H_y = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Then

$$H = H_x * H_y = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Gaussian Kernel

- 1D

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

- 2D

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Separability of 2D Gaussian

- 2D gaussian is just product of 1D gaussians:

$$
\begin{aligned}
G_\sigma(x, y) &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \\
&= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \cdot \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right) \\
&= g_\sigma(x) \cdot g_\sigma(y)
\end{aligned}
$$

Separable!

Separability of 2D Gaussian

- Consequently, convolution with a gaussian is separable

$$I * G = I * G_x * G_y.$$

- Where *G* is the 2D discrete gaussian kernel;
- G_x is “horizontal” and G_y is “vertical” 1D discrete Gaussian kernels