Simple operations on pixels.

$J(x, y)$ – brightness at the point (x,y)

$J_{Out}(x, y)$ – output brightness at the point (x,y)

$\boldsymbol{J_{Out}(x, y) = aJ(x, y) + b}$ **– linear operations**

1. $J_{Out}(x, y) = J(x, y) \pm b$ – brightening or darkening
2. $J_{Out}(x, y) = aJ(x, y)$ – contrast editing
3. $J_{Out}(x, y) = 255 - J(x, y)$ – negation
4. $J_{Out}(x, y) = \big(J_R(x, y) + J_G(x, y) + J_B(x, y)\big)/3$ – conversion to grayscale
5. $J_{Out}(x, y) = 255 \cdot \frac{J(x,y) - J_{min}}{J_{max} - J_{min}}$ – normalization

$\boldsymbol{J_{Out}(x, y) = \varphi[J(x, y)]}$ **– nonlinear operations**

1. $J_{Out}(x, y) = J(x, y)^{\alpha}, \alpha > 0$ – contrast editing, requires normalization
   $J_{Out}(x, y) = 255\left(\frac{J(x,y)}{J_{max}}\right)^{\alpha}$ – normalization
   a. $\alpha > 1$ – exponentiation, darkening with greater variation of bright parts
   b. $\alpha < 1$ – roots, brightening with greater variation of dark parts – gamma correction
2. $J_{Out}(x, y) = 255\frac{\log(1+J(x,y))}{\log(1+J_{max})}$ – logarithm with normalization, brightening with greater variation of dark parts

Histogram – graph showing the number of pixels of a given brightness.

Projections – Vertical and horizontal projection of points number in a given direction. Taken in binary image.

Thresholding – Binaryzation of image. Above the fixed brightness is assumed to be all pixels white, below black.

Convolution operations - filters

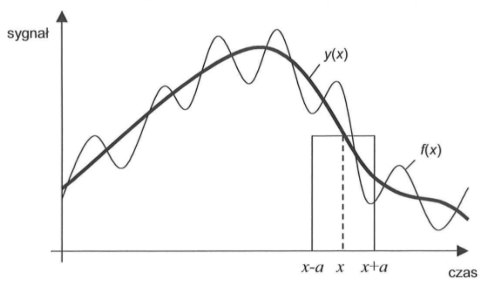
Convolution in one-dimensional case:

$$y(x) = \int_{-\infty}^{+\infty} f(x-t)h(t)dt = f(t)^{\circ}h(t)$$

The function h(t) – filter, moves across the whole domain of the function f(t).

For the filter $h(t) = \begin{cases} \frac{1}{2a}, & dla\ t \in (-a,\ a) \\ 0, & dla\ t \notin (-a,\ a) \end{cases}$ – averaging filter

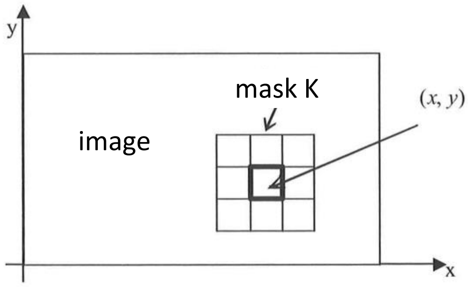
Convolution is: $y(x) = \frac{1}{2a}\int_{-a}^{a} f(x-t)dt$



In image analysis, the domain of brightness is two-dimensional and discrete. Convolution greatly simplifies.

$J_{Out}(x,y) = \sum_{i,j\in K} J(x-i, y-j)w(i,j)$

$$K - neighbourhood\ of\ the\ pixel - mask$$
$$w(i,j) - weights\ of\ the\ neighbourhood\ pixels$$



- Masks: 2x2, 3x3, and greater
- Often needed normalization

$$J_{Out}(x,y) = \frac{\sum_{i,j\in K} J(x-i, y-j)w(i,j)}{\sum_{i,j\in K} w(i,j)}$$

1. Low-pass filters

   Masks: $\begin{bmatrix} 1 & 1 & 1 \\ 1 & a & 1 \\ 1 & 1 & 1 \end{bmatrix}, a = 0, 1, 2, 4, 12, \ldots$

2. Gaussian filters

   Masks: $\begin{bmatrix} 1 & b & 1 \\ b & b^2 & b \\ 1 & b & 1 \end{bmatrix}, b = 1, 2, 3, 4, \ldots$ e.g. $\begin{bmatrix} 1 & 4 & 1 \\ 4 & 12 & 4 \\ 1 & 4 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 3 & 1 \\ 3 & 16 & 3 \\ 1 & 3 & 1 \end{bmatrix}, \ldots$

3. High-pass filters – Laplacians

   Laplacian - sum of second partial derivatives.

$$\frac{dJ}{dx^2} \approx [J(x+1,y) - J(x,y)] - [J(x,y) - J(x-1,y)]$$
$$= J(x+1,y) - 2J(x,y) + J(x+1,y)$$

$$\frac{dJ}{dy^2} \approx [J(x,y+1) - J(x,y)] - [J(x,y) - J(x,y-1)]$$
$$= J(x,y-1) - 2J(x,y) + J(x,y+1)$$

Mask: $\begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

Other masks:

Derivatives on the diagonals, or all derivatives, e.g.: $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

Derivatives for three parallel lines in vertical and horizontal: $\begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$

4. Edge detection filters

First and second derivatives. On the edges where there is a large change in brightness give a large value.

a. Roberts operators

$|\nabla f| = |f(x+1,y) - f(x,y)| + |f(x,y+1) - f(x,y)|$, gives: $\begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}$

Another masks: $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$

b. Prewitt operators

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \ldots$$

(Rotation to get others)

c. Sobel operators

$$\begin{matrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{matrix} \qquad \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix} \qquad \begin{matrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{matrix}$$

135° 90° 45°

$$\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix} \qquad \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

180° 0°

$$\begin{matrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{matrix} \qquad \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix} \qquad \begin{matrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{matrix}$$

225° 270° 315°

For output pixels with negative values, normalization should be done, scaling of the brightness or taking an absolute value.