Image Processing Lecture-4

Pixel neighbor operations

Hazırlayan: M. Kemal GÜLLÜ

Image mean and variance

- Image mean:

$$\mu = \mathrm{E}(X) = \bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

- Image variance:

$$\sigma^2 = Var(X) = \mathrm{E}\left((X-\mu)^2\right)$$
$$= \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

- In MATLAB, the built-in **mean2** function enables computation of 2D matrix mean.
- For the computation of variance, 2D standard deviation can be computed using built-in **std2** function first. Next, the variance is computed as its square.

Image mean and variance

- In MATLAB, divide an image into overlapping blocks (n x n) and compute local mean and standard deviation of for each pixel position.
- Create a standard deviation image using the above described approach.

Pixel neighbor operations

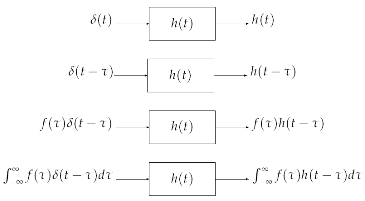
- A new value is computed for each pixel.
- The new pixel value is computed using neighbor pixels.
- The weights of pixel in the computed is decided according to operation carried out.
- Pixel operations used in edge detection, noise reduction, image sharpening, image smoothing, etc.
- Computation load can be significantly higher than the point operations where neighbors are not used.

Convolution

- In 1-D signals:

$$f * g = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

- Used extensively in image processing.



Convolution

- In 2-D

$$g(x,y) = k * f$$
$$= \sum_{i=-m}^{m}\sum_{j=-n}^{n} k(i,j)\, f(x-i, y-j)$$

k, convolution kernel
f, input image
g, output image
(x,y), pixel position
$(2m+1, 2n+1)$, width and height of kernel

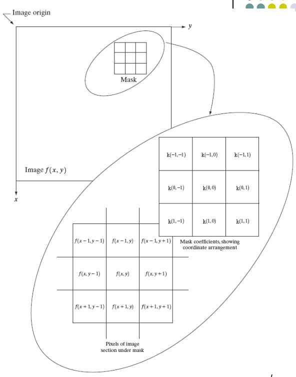
- The kernel is also called as convolution mask and convolution window.

Convolution

$$g(x,y)=k*f$$
$$=\sum_{i=-m}^{m}\sum_{j=-n}^{n}k(i,j)f(x-i,y-j)$$

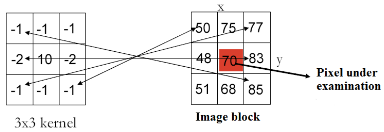
$$g(x,y)=k(-m,-n)f(x+m,y+n)$$
$$+k(-m+1,-n+1)f(x+m-1,y+n-1)$$
$$+\ldots$$
$$+k(m,n)f(x-m,y-n)$$

$$g(x,y)=k(-1,-1)f(x+1,y+1)$$
$$+k(-1,0)f(x+1,y)$$
$$+\ldots$$
$$+k(1,1)f(x-1,y-1)$$



Convolution

$$g(x,y)=k*f$$
$$=\sum_{i=-m}^{m}\sum_{j=-n}^{n}k(i,j)f(x-i,y-j)$$

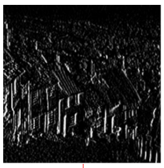


g(x,y)=k(-1,-1)f(x+1,y+1) + k(0,-1)f(x,y+1) + k(1,-1)f(x-1,y+1) + k(-1,0)f(x+1,y)
+ k(0,0)f(x,y) + k(1,0)f(x-1,y) + k(-1,1)f(x+1,y-1) + k(0,1)f(x,y-1)
+k (1,1)f(x-1,y-1)

g(x,y)=(-1x85) + (-1x68) + (-1x51) + (-2x83) + (10x70) + (-2x48) + (-1x77)
+ (-1x75) + (-1x50) = 32

Evrişim (Convolution)

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



Input image

Kernel

Output image

- In MATLAB, built-in **conv2** function performs 2-D convolution.
- It is also possible to utilize **imfilter** function.

Convolution

How to solve problems at the image borders :

- Neglecting the image borders by not processing them
- Cropping the image borders
- Cropping the kernel at the borders
- Copying the pixel values at the border (image becomes larger),
- Copying the pixel values at the border by mirroring (image becomes larger)...

Computational complexity:

- An convolution kernel size of (m,n) results in number of computations given below for each pixel

$$(m\times n)[\varsigma arpma]+(m\times n-1)[toplama]$$

Convolution

$$\begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}$$ Delta function

$$\begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & -1\end{bmatrix}$$ Shift and subtract

Convolution

$$\begin{bmatrix}-1/8 & -1/8 & -1/8\\ -1/8 & 1 & -1/8\\ -1/8 & -1/8 & -1/8\end{bmatrix}$$ Edge detection

$$\begin{bmatrix}-k/8 & -k/8 & -k/8\\ -k/8 & k+1 & -k/8\\ -k/8 & -k/8 & -k/8\end{bmatrix}$$ Edge enhancement

Spatial frequency

- Smooth transitions in image regions correspond low spatial frequency range.

- Sharp transitions (edges, object borders, etc) in image regions correspond low spatial frequency range.

Convolution - Smoothing

- The simplest kernel
- Reduces the noise
- Smoothens the edges

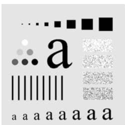
$$1/9\times\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

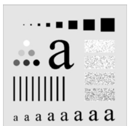
$$1/25\times\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}$$

 → 

Convolution - Smoothing

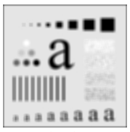
- Kernel size effect

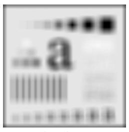

Orjinal imge


3x3


5x5


9x9


15x15


35x35

Convolution - Smoothing

- Weighted averaging:

$$g(x,y)=\frac{\sum_{i=-m}^{m}\sum_{j=-n}^{n} w(i,j)\,f(x-i,y-j)}{\sum_{i=-m}^{m}\sum_{j=-n}^{n} w(i,j)} \qquad 1/15\times\begin{bmatrix}1&2&1\\2&3&2\\1&2&1\end{bmatrix}$$

- To eliminate smoothing at the corners while smoothing homogenous region

$$g(x,y)=\begin{cases}\dfrac{1}{ws\times ws}\sum_i\sum_j f(x-i,y-j) \ , & \left|f(x,y)-\dfrac{1}{ws\times ws}\sum_i\sum_j f(x-i,y-j)\right|<T\\ f(x,y) \ , & \text{diğer}\end{cases}$$

↓

Implement in MATLAB