Fundamentals of Image Processing (part II)

Alexandre Xavier Falcão

Institute of Computing - UNICAMP

afalcao@ic.unicamp.br

Adjacency-based image transformations

- We presented some point-based and adjacency-based transformations in the previous lecture.

Adjacency-based image transformations

- We presented some point-based and adjacency-based transformations in the previous lecture.

- This lecture covers another important adjacency-based transformation: convolution.

Adjacency-based image transformations

- We presented some point-based and adjacency-based transformations in the previous lecture.
- This lecture covers another important adjacency-based transformation: convolution.
- The convolution between an image and a kernel defines a linear filtering.

Adjacency-based image transformations

- We presented some point-based and adjacency-based transformations in the previous lecture.
- This lecture covers another important adjacency-based transformation: convolution.
- The convolution between an image and a kernel defines a linear filtering.
- One may use multiple linear filters for image feature extraction in deep learning.

Agenda

- Simple kernel and multi-band kernel.
- Convolution with a multi-band kernel.
- Kernel bank.
- Convolution with a kernel bank.

Kernel

- For a given adjacency $\mathcal{A}$, a kernel $(\mathcal{A}, W)$ is a moving image, where $W(q_k - p) = w_k \in \Re$ is a weight assigned to the adjacent $q_k \in \mathcal{A}(p)$ of any pixel p, $k = 1, 2, \ldots, K$.

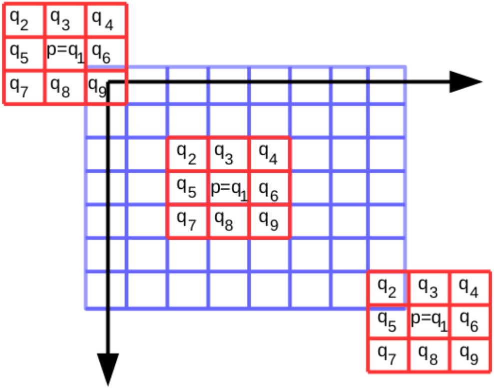
Kernel

- For a given adjacency $\mathcal{A}$, a kernel $(\mathcal{A}, W)$ is a moving image, where $W(q_k - p) = w_k \in \Re$ is a weight assigned to the adjacent $q_k \in \mathcal{A}(p)$ of any pixel p, $k = 1, 2, \ldots, K$.
- For images with m bands (e.g., color images), the kernel is a multi-band image $(\mathcal{A}, \mathrm{W})$, where $\mathrm{W}(q_k - p) = \mathrm{w}_k \in \Re^m$ is a weight vector assigned to the adjacent $q_k \in A(p)$.

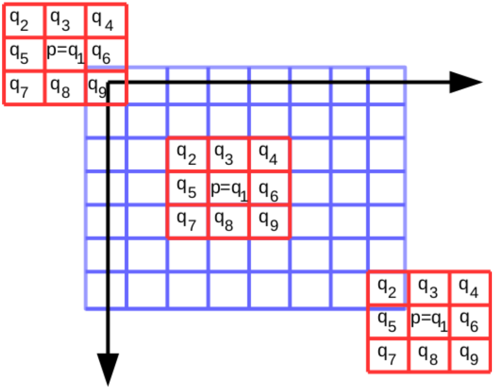
Kernel

- For a given adjacency $\mathcal{A}$, a kernel $(\mathcal{A}, W)$ is a moving image, where $W(q_k - p) = w_k \in \Re$ is a weight assigned to the adjacent $q_k \in \mathcal{A}(p)$ of any pixel p, $k = 1, 2, \ldots, K$.
- For images with m bands (e.g., color images), the kernel is a multi-band image $(\mathcal{A}, \mathrm{W})$, where $\mathrm{W}(q_k - p) = \mathrm{w}_k \in \Re^m$ is a weight vector assigned to the adjacent $q_k \in A(p)$.
- For filtering an image $\hat{I} = (D_I, \mathrm{I})$ with m bands, a kernel slides from $(-\infty, -\infty)$ to $(+\infty, +\infty)$ (top-to-bottom,left-to-right), but the filtered image is computed only for $p \in D_J \subseteq D_I$.

A $3 \times 3 \times m$ filter sliding from top to bottom and left to right over the image domain D_I.



A $3 \times 3 \times m$ filter sliding from top to bottom and left to right over the image domain D_I.



Different choices of kernel weights imply distinct filters.

Convolution

The convolution between an image $\hat{I} = (D_I, \mathrm{I})$ and a filter $(\mathcal{A}, \mathrm{W})$ creates a gray-scale image $\hat{J} = (D_J, J)$,

$$
\begin{aligned}
J(p) &= \sum_{k=1}^{K} \langle \mathrm{I}(q_k), \mathrm{w}_k \rangle, \\
\langle \mathrm{I}(q_k), \mathrm{w}_k \rangle &= \sum_{i=1}^{m} I_i(q_k) w_{ki},
\end{aligned}
$$

for all $p \in D_I$, with D_J forced to be $\subseteq D_I$.

Convolution

Image function I

3	2	2	5
2	2	5	1
2	3	5	0
0	4	4	1

Zero-padding in red

Image function J

6	1	7	-9
9	8	-2	-17
12	13	-10	-19
11	11	-9	-13

same image domain

Kernel 3 x 3

-1	0	1
-2	0	2
-1	0	1

The convolution algorithm

- Input: $\hat{I} = (D_I, \mathrm{I})$ and $\{dx_k, dy_k, \mathrm{w}_k\}$, $k = 1, 2, \ldots, K$.
- Output: $\hat{J} = (D_J, J)$.

1. For each $p = (x_p, y_p) \in D_J$, do
2. $\quad J(p) \leftarrow 0$.
3. $\quad$ For $k \leftarrow 1, 2, \ldots, K$, do
4. $\quad\quad q = (x_q, y_q) \leftarrow (x_p + dx_k, y_p + dy_k)$
5. $\quad\quad$ If $q = (x_q, y_q) \in D_I$, then
6. $\quad\quad\quad J(p) \leftarrow J(p) + \langle \mathrm{I}(q) \cdot \mathrm{w}_k \rangle$.

Image filtering for feature extraction

- The Sobel filters, for example, can enhance vertical and horizontal edges of $\hat{I}$. The corresponding moving images, in which the origin p is the central pixel, are

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

The Sobel-vertical-edge kernel can enhance the characters of a car plate and, as we will see later, the integral image can be exploited to assign higher scores to the best candidate locations.

Convolution with a kernel bank

The convolution between an image $\hat{I} = (D_I, \mathrm{I})$ and a set (bank) of b kernels $\{(\mathcal{A}, \mathrm{W}_j)\}_{j=1}^{b}$ produces an image $\hat{J} = (D_J, \mathrm{J})$ with b bands J_j, $j \in [1, b]$,

$$J_j(p) = \sum_{k=1}^{K} \langle \mathrm{I}(q_k), \mathrm{w}_{k,j} \rangle.$$

Convolution with a kernel bank

The convolution between an image $\hat{I} = (D_I, \mathrm{I})$ and a set (bank) of b kernels $\{(\mathcal{A}, \mathrm{W}_j)\}_{j=1}^{b}$ produces an image $\hat{J} = (D_J, \mathrm{J})$ with b bands J_j, $j \in [1, b]$,

$$J_j(p) \;\; = \;\; \sum_{k=1}^{K} \langle \mathrm{I}(q_k), \mathrm{w}_{k,j} \rangle.$$

The adjacency $\mathcal{A}$ is usually fixed for all kernels because the convolution can be very efficiently computed by matrix multiplication using parallel programming.

Convolution with a kernel bank

The convolution between an image $\hat{I} = (D_I, \mathrm{I})$ and a set (bank) of b kernels $\{(\mathcal{A}, \mathrm{W}_j)\}_{j=1}^{b}$ produces an image $\hat{J} = (D_J, \mathrm{J})$ with b bands J_j, $j \in [1, b]$,

$$J_j(p) \quad = \quad \sum_{k=1}^{K} \langle \mathrm{I}(q_k), \mathrm{w}_{k,j} \rangle.$$

The adjacency $\mathcal{A}$ is usually fixed for all kernels because the convolution can be very efficiently computed by matrix multiplication using parallel programming.

One can also reduce the number of bands from m to $b < m$ by convolving an image with b filters of size $1 \times 1 \times m$.

A kernel bank $\{(\mathcal{A}, \mathrm{W}_j)\}_{j=1}^{b}$ may be organized such that the weights $\mathrm{w}_{k,j}$ of each kernel $(\mathcal{A}, \mathrm{W}_j)$ are placed along column j, forming a kernel matrix $\mathcal{K}$.

$$\mathcal{K} = \begin{bmatrix} \mathrm{w}_{1,1} & \mathrm{w}_{1,2} & \ldots & \mathrm{w}_{1,b} \\ \mathrm{w}_{2,1} & \mathrm{w}_{2,2} & \ldots & \mathrm{w}_{2,b} \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{w}_{K,1} & \mathrm{w}_{K,2} & \ldots & \mathrm{w}_{K,b} \end{bmatrix}_{mK \times b}$$

where each vector $\mathrm{w}_{k,j} \in \Re^m$ is a column matrix.

Image filtering for feature extraction

For an image $(D_I, \mathbf{I})$ and adjacency $\mathcal{A}$, the vectors $\mathbf{I}(q_{i,k}) \in \Re^m$ of $q_{i,k} \in \mathcal{A}(p_i)$, $k \in [1, K]$, adjacent to each pixel $p_i \in D_I$, $i \in \{1, 2, \ldots, |D_I|\}$, are organized along the rows of a data matrix $\mathcal{X}_I$.

$$\mathcal{X}_I = \begin{bmatrix} \mathbf{I}(q_{1,1}) & \mathbf{I}(q_{1,2}) & \ldots & \mathbf{I}(q_{1,K}) \\ \mathbf{I}(q_{2,1}) & \mathbf{I}(q_{2,2}) & \ldots & \mathbf{I}(q_{2,K}) \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{I}(q_{|D_I|,1}) & \mathbf{I}(q_{|D_I|,2}) & \ldots & \mathbf{I}(q_{|D_I|,K}) \end{bmatrix}_{|D_I| \times mK}$$

where each vector $\mathbf{I}(q_{i,k}) \in \Re^m$ is a row matrix.

The multiplication $\mathcal{X}_I \mathcal{K}$ outputs a matrix $\mathcal{X}_J$,

$$\mathcal{X}_J = \begin{bmatrix} \mathrm{J}(p_1) \\ \mathrm{J}(p_2) \\ \vdots \\ \mathrm{J}(p_{|D_I|}) \end{bmatrix}_{|D_I| \times b}$$

where each vector $\mathrm{J}(p_i) \in \Re^b$, $i = 1, 2, \ldots, |D_I|$, is a row matrix.

That is, $\mathcal{X}_J$ is the matrix organization of the filtered image (D_J, J), $D_J = D_I$.

Examples

Let's see Convolution.ipynb in notebooks.tar.gz