

Procesamiento de Imágenes

Fecha: 20-Marzo-2020
Inicio: 1:00 pm

Reposo Lección Rescada:

$$B_{ij} = \left[\frac{S_{\max} - S_{\min}}{r_{\max} - r_{\min}} \right] \cdot (A_{ij} - r_{\min}) + S_{\min}$$

Vecindarios y Convolución: En lecciones anteriores, vimos la operación promedio para completar un pixel cuyo valor es 0.



①-⑧ Píxeles Vecinos
→ Usamos los valores de ①-⑧ para completar el pixel faltante.

Pixel Faltante.

En general, el concepto de "Procesamiento por Vecindario" significa modificar el valor de un pixel, utilizando la información de los pixeles que lo rodean.

Las operaciones que se realizan en un procesamiento por vecindario, normalmente siguen los siguientes pasos: Sea $A \in \mathbb{R}^{m \times n}$ una imagen a escala de grises:

① Definir un pixel de la imagen: A_{ij}

①

$A_{i-1,j-1}$	$A_{i-1,j}$	$A_{i-1,j+1}$
$A_{i,j-1}$	$A_{i,j}$	$A_{i,j+1}$
$A_{i+1,j-1}$	$A_{i+1,j}$	$A_{i+1,j+1}$

② Desarrollar una operación que involucre los pixeles del vecindario: $A_{i-1,j-1}, A_{i-1,j}, \dots, A_{i+1,j+1}$.

③ Sea $B \in \mathbb{R}^{m \times n}$ la imagen de salida. Aplicar el resultado de la operación en la misma coordenada (i,j) de la imagen B . B_{ij}

$A_{i,j}$	$A_{i,j+1}$		
$A_{i+1,j}$	$A_{i+1,j+1}$		
$A_{i-1,j}$	$A_{i-1,j+1}$		
$A_{i,j}$	$A_{i,j+1}$		
$A_{i+1,j}$	$A_{i+1,j+1}$		

④ Repetir el proceso para todos los pixeles

Nota: En esta parte del curso, estudiaremos un conjunto de métodos relacionados con el procesamiento por vecindario que involucra la operación de convolución.

Convención en 1D para vectores de dimensión infinito.

Sea x, y 2 vectores de dimensión infinito.

$$x = [\dots, x_{-3}, x_{-2}, x_{-1}, x_0, x_1, x_2, x_3, \dots]$$

$$y = [\dots, y_{-3}, y_{-2}, y_{-1}, y_0, y_1, y_2, y_3, \dots]$$

La convolución entre x y y es el vector z de dimensión infinito definida por:

$$z_j = (x * y)_j = \sum_{i=-\infty}^{\infty} x(i) \cdot y(j-i)$$

Convención en 1D para vectores de dimensión finita

Sea $x = [x_1, \dots, x_m]^T \in \mathbb{R}^m$ y $y = [y_1, \dots, y_n]^T \in \mathbb{R}^n$. Entonces la convolución de x, y es el vector $z \in \mathbb{R}^{m+n-1}$, definido por:

$$z_i = (x * y)_i = \sum_j x_j \cdot y_{i-j+1}$$

donde $j \in \{\underbrace{\max(1, i+1-n)}_{\text{Inicio}}, \dots, \underbrace{\min(i, m)}_{\text{Fin}}\}$.

Convención en 2D para matrices de dimensión finita.

Sea $A \in \mathbb{R}^{m_1 \times n_1}$, $B \in \mathbb{R}^{m_2 \times n_2}$. La convolución de A y B es una matriz $C \in \mathbb{R}^{(m_1+m_2-1) \times (n_1+n_2-1)}$, definida por:

$$C_{j,k} = (A * B)_{j,k} = \sum_p \sum_q A_{p,q} \cdot B_{j-p+1, k-q+1}.$$

donde

$$p \in \{ \max(1, j-m_2+1), \dots, \min(j, m_1) \}$$

$$q \in \{ \max(1, k-n_2+1), \dots, \min(k, n_1) \}$$

Tarea: Implementar la fórmula de convolución de matrices, y calcule $A * B$, donde

$$A = \begin{pmatrix} -2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix}; \quad B = \begin{pmatrix} -4 & 3 & 4 \\ 0 & 0 & 0 \end{pmatrix}.$$

Nota: En GNU Octave, el comando '`conv2(A,B)`' realiza la convolución entre las matrices A y B .

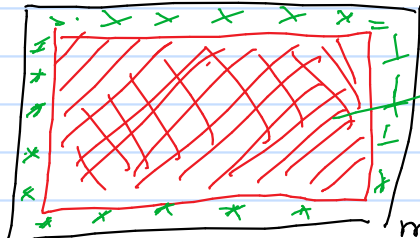
Aplicación de la convolución en imágenes

Sea $A \in \mathbb{R}^{m \times n}$ una imagen en escala de grises. Sea $B \in \mathbb{R}^{3 \times 3}$.

Dependiendo del valor de B , se puede aplicar una transformación a la imagen A , a través de la convolución.

• $B = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & 1 & 0 \end{bmatrix}$, entonces B es un filtro-paso alto.

Note que $A * B \in \mathbb{R}^{(m+2) \times (n+2)}$. Por lo tanto, en imágenes, eliminamos la primera y última fila y la primera y última columna.

$(A * B) =$  $m+2 \times n+2$. Consideramos el centro.

• $B = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$, entonces B es un filtro paso-bajo.

• $B = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$, entonces B detecta bordes horizontales.

• $B = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$, entonces B detecta bordes verticales.

• $B = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$, entonces B detecta bordes en general.