

Procesamiento de imágenes

Convolución

https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/filter_2d/filter_2d.html

https://docs.opencv.org/3.4/d4/d86/group_imgproc_filter.html

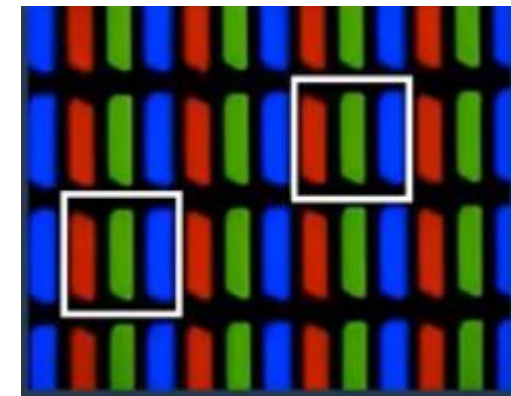
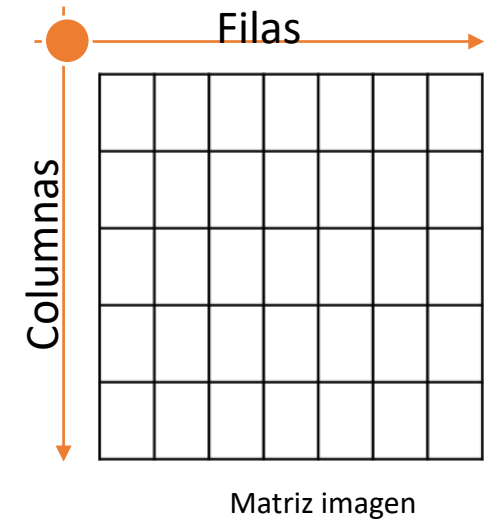
<https://developer.apple.com/library/archive/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html>

<http://dx.doi.org/10.4995/msel.2016.4524>

<https://www.youtube.com/watch?v=BQyMZ0caFbg>

Repaso de conceptos

- ¿Qué es imagen?
 - Es una matriz (filas y columnas) de pixeles.
- ¿Qué es un pixel?
 - Es la unidad más pequeña de la imagen y están descritos en intensidad (de 0 a 255) de rojo, verde y azul (RGB).
 - Ejemplo: Azul en `rgb(0, 0, 255)` y blanco en `rgb(255, 255, 255)`



Conjunto de pixeles

Kernel

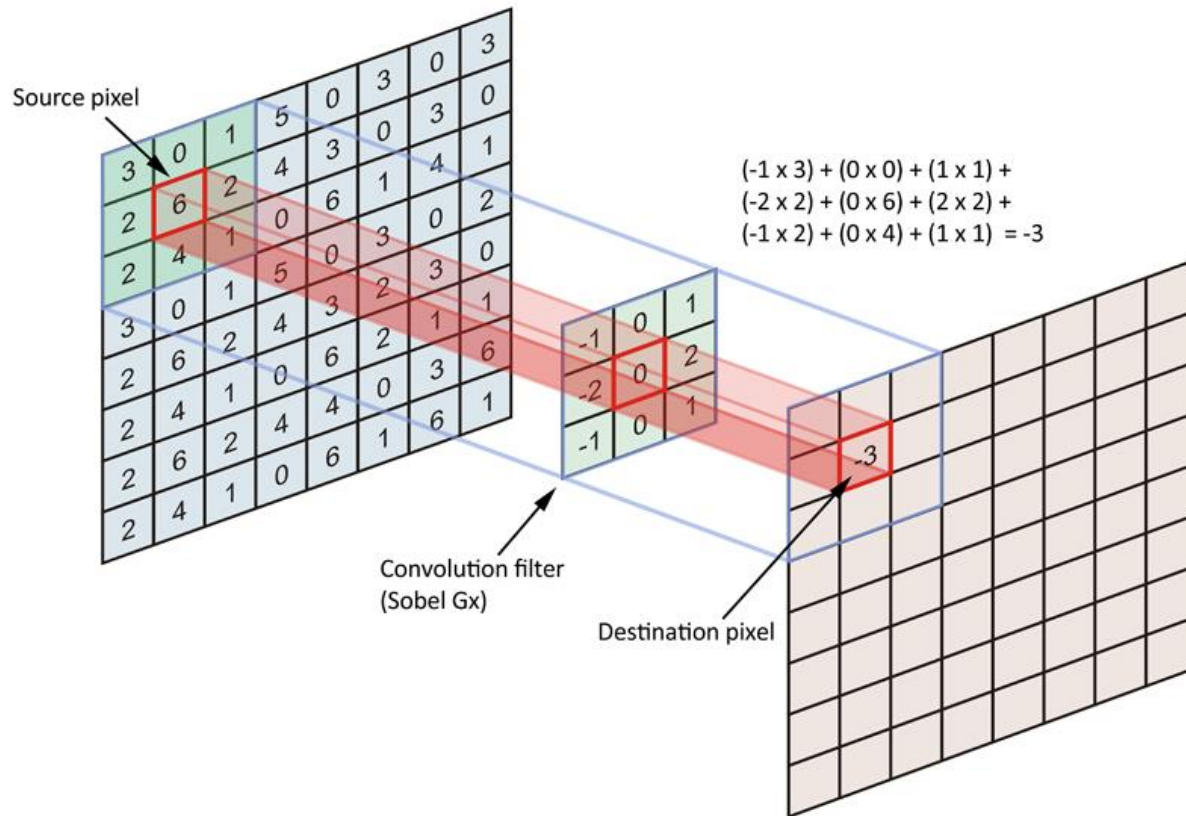
Un *kernel* es esencialmente una matriz de coeficientes numéricos de tamaño fijo que cuenta con un punto de anclaje en esa matriz, este se encuentra generalmente en el centro.

- Puede considerarse como una cuadrícula bidimensional de números que pasa sobre cada píxel de una imagen en secuencia, realizando cálculos a lo largo del camino.
- Cada operación en procesamiento de imágenes tiene un *kernel*.
 - Ejemplo: El *kernel* para la operación desenfoque (*blur*) es:

$$K = \frac{1}{\text{ksize.width} * \text{ksize.height}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ \dots & & & & & \\ 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix} \longrightarrow K = \frac{1}{3 \cdot 3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Convolución

Convolución es una operación entre cada parte de una imagen y un operador (kernel).



- Los números en el kernel representan la cantidad por la cual multiplicar el número de la fuente.
- El número de la fuente representa la intensidad del píxel sobre el que flota el elemento del kernel.
- Durante la convolución, el centro del núcleo pasa sobre cada píxel en la imagen.
- El proceso multiplica cada número en el kernel por el valor de intensidad de píxel directamente debajo de él (de la fuente).
- Esto debería generar tantos productos como números en el kernel (por píxel).
- El paso final del proceso suma todos los productos juntos, los divide por la cantidad de números en el kernel, y este valor se convierte en la nueva intensidad del píxel que estaba directamente debajo del centro del kernel.

Bordes

Para aplicar la convolución en los bordes de la imagen, existen varias alternativas, entre las que se encuentran:

- Completar con ceros los valores de alrededor.
- Completar los valores de la parte simétrica opuesta.
- Repetir los valores en el borde.

| | | | | |
|---|---|---|---|---|
| a | a | b | c | c |
| a | a | b | c | c |
| d | d | e | f | f |
| g | g | h | i | i |
| g | g | h | i | i |

Filtros o máscaras

Las operaciones de convolución también son conocidas como filtros o máscaras.

- Algunas de las más comunes son:

- Tipos de filtro:

| Enfoque | Desenfoque | Realce de bordes | Repujado |
|---|---|--|--|
| $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ | $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ | $\begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ | $\begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$ |

| Detección de bordes | Filtro de tipo Sobel | Filtro de tipo Sharpen |
|--|--|---|
| $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ | $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ | $\begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$ |

| Filtro Norte | Filtro Este | Filtro de tipo Gauss |
|---|---|--|
| $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$ | $\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$ | $\begin{bmatrix} 1 & 2 & 3 & 1 & 1 \\ 2 & 7 & 11 & 7 & 2 \\ 3 & 11 & 17 & 11 & 3 \\ 2 & 7 & 11 & 7 & 1 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$ |

Filtros de paso bajo:

- Suavizado de imagen
- Eliminación de ruido
- Ejemplo: Desenfoque

Filtros de paso alto:

- Resalto de zonas de mayor variabilidad
- Sustracción de la media
- Ejemplo: Enfoque

Filtros direccionales:

- Resalto de píxeles que determinan direcciones

Filtros para la detección de bordes:

- Aplicación en ingeniería, estudio del terreno
- Ejemplo: Detectar bordes

OpenCV - Python

- Kernel
 - `np.array()`
- Convolución
 - `cv2.filter2D()`

Práctica

```
1  #https://opencv-python-tutroals.readthedocs.io/en/latest/p
2
3  import cv2
4  import numpy as np
5  from matplotlib import pyplot as plt
6
7  # Load image.
8  img = cv2.imread('bubbles.jpg')
9
10 #kernel = np.ones((5,5),np.float32)/25
11 kernel = np.array([[0,1,0],[1,-4,1],[0,1,0]])
12 dst = cv2.filter2D(img,-1,kernel)
13
14 plt.subplot(121),plt.imshow(img),plt.title('Original')
15 plt.xticks([], plt.yticks([]))
16 plt.subplot(122),plt.imshow(dst),plt.title('Convolution')
17 plt.xticks([], plt.yticks([]))
18 plt.show()
```