



# Filtragem Espacial

---

## Visão Computacional

Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis M. L. Martins

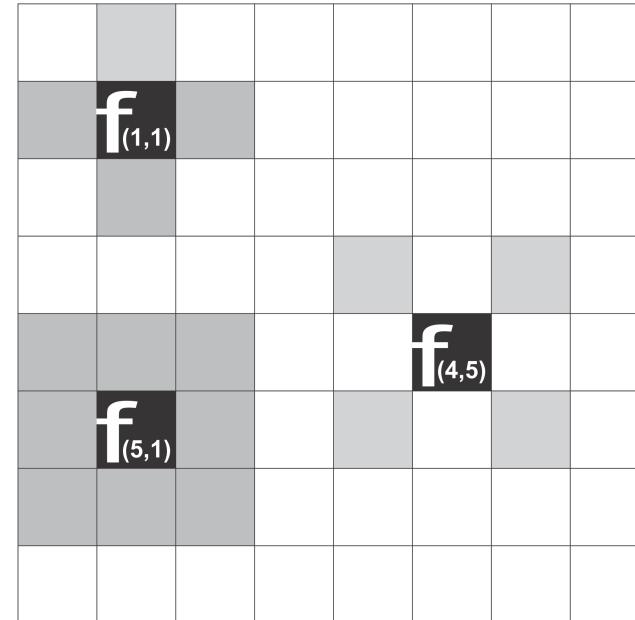
# Objetivos de Aprendizagem

---

- **Compreender** os fundamentos matemáticos que sustentam a filtragem espacial em imagens digitais.
- **Implementar** operações de convolução, suavização e realce utilizando kernels clássicos (Gaussiano, Laplaciano, Sobel, etc.).
- **Avaliar** o resultado da aplicação dos filtros através de percepção visual.
- **Aplicar** técnicas de filtragem para resolver problemas reais: remoção de ruído salt-and-pepper, realce de bordas, etc.

# Introduzindo os Conceitos de Conectividade e Vizinhança de um Pixel

- Cada pixel possui **vizinhos** que definem sua relação espacial com o restante da imagem.
- **Conectividade**: Dois pixels são **conectados** se existe um caminho entre eles formado por pixels vizinhos.
- **Vizinhança de um pixel**: Conjunto  $\mathcal{N}(x, y)$  que contém todos os pixels que podem ser alcançados a partir de  $(x, y)$  via conectividade.
  - Conectividade 4 pode separar objetos que tocam apenas em cantos; conectividade 8 os une.



Exemplo de vizinhança de um pixel. Fonte: [COVAP-UTFPR](#).

- **Coordenadas** da Vizinhança de um pixel na posição  $(x, y)$ :
  - $\mathcal{N}_4(x, y) = \{(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)\}$
  - $\mathcal{N}_D(x, y) = \{(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)\}$
  - $\mathcal{N}_8(x, y) = \mathcal{N}_4(x, y) \cup \mathcal{N}_D(x, y)$

# Adjacência

- **Adjacência:** Dois pixels são **adjacentes** se compartilham pelo menos um ponto comum (contorno ou vértice).
  - **4-adjacência** – vizinhos que compartilham uma borda:  $(x \pm 1, y)$  e  $(x, y \pm 1)$ .
  - **8-adjacência** – inclui também os quatro diagonais:  $(x \pm 1, y \pm 1)$ .
- **Conectividade:** Um conjunto de pixels forma uma **região** se cada pixel da região pode ser alcançado a partir de qualquer outro via um caminho de adjacentes dentro da mesma região.

a	0	1	1		b	0	1	-1		c	0	1	-1	
	0	1	0			0	1	0			0	1	0	
	0	0	1			0	0	1			0	0	1	
d	1	1	1	$R_i$	e	0	0	0	0	f	0	0	0	
	1	0	1			0	1	1	0		0	1	0	
	0	1	0			0	1	1	0		0	1	0	
	0	0	1	$R_j$		0	1	1	1		0	1	0	
	1	1	1			0	1	1	1		0	1	0	
	1	1	1			0	0	0	0		0	0	0	

**Figura 2.25** (a) Um arranjo de pixels. (b) Pixels que são adjacentes-8 (a adjacência é mostrada em linhas tracejadas; observe a ambiguidade). (c) adjacência-m. (d) Duas regiões (de 1s) que são adjacentes se a adjacência-8 for utilizada. (e) O ponto circulado é parte da fronteira dos pixels de valor 1 somente se a adjacência-8 entre a região e o fundo for utilizada. (f) A fronteira interna da região de valor 1 não forma um caminho fechado, mas sua fronteira externa, sim.

Exemplo de adjacência de pixels. Fonte: Jackson Gomes.

# Filtragem Espacial

- **Ideia Geral:** Cada pixel  $f(x, y)$  é substituído por uma **combinação ponderada dos vizinhos**.
- **Objetivo principal:** modificar características da imagem (suavizar, realçar bordas, remover ruído).
- **Componente principal:** **Filtro** aplicado sobre a imagem. Expresso como uma **matriz** ou, de forma análoga, como uma imagem.
  - **Nomenclatura:** Filtros, Máscaras, Kernel:  $h(m, n)$ .

(a) imagem aguçada



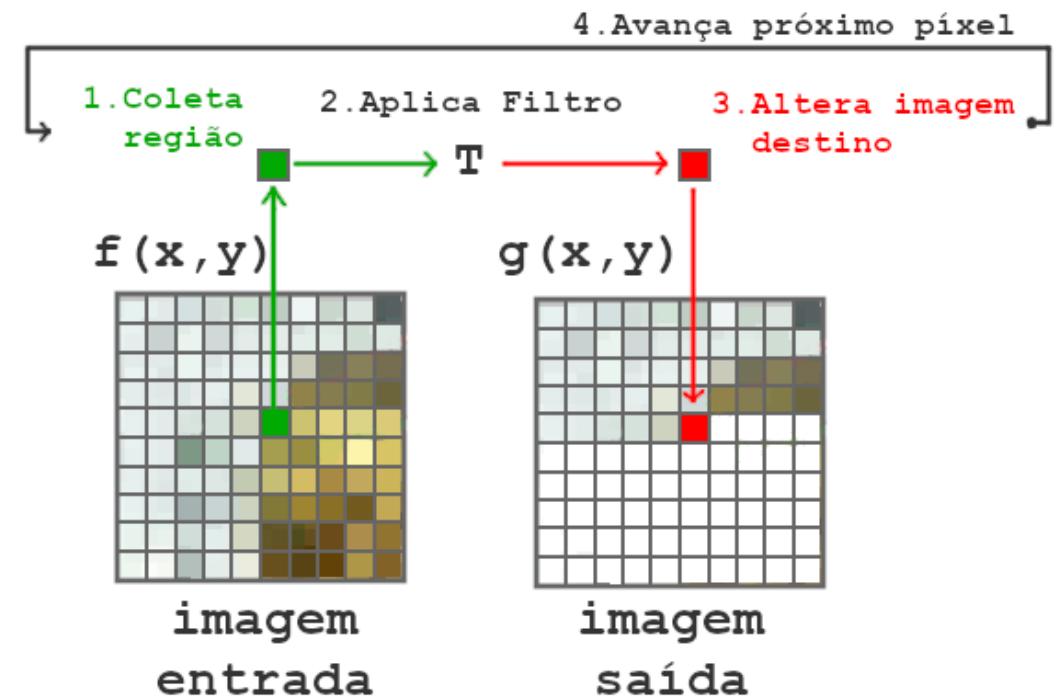
(b) imagem suavizada



Exemplo de aplicação de filtro de aguçamento e de suavização. Fonte: [COVAP-UTFPR](#)  
[PUC-Campinas](#) | Prof. Dr. Denis Martins | [denis.mayr@puc-campinas.edu.br](mailto:denis.mayr@puc-campinas.edu.br)

# Filtro Espacial

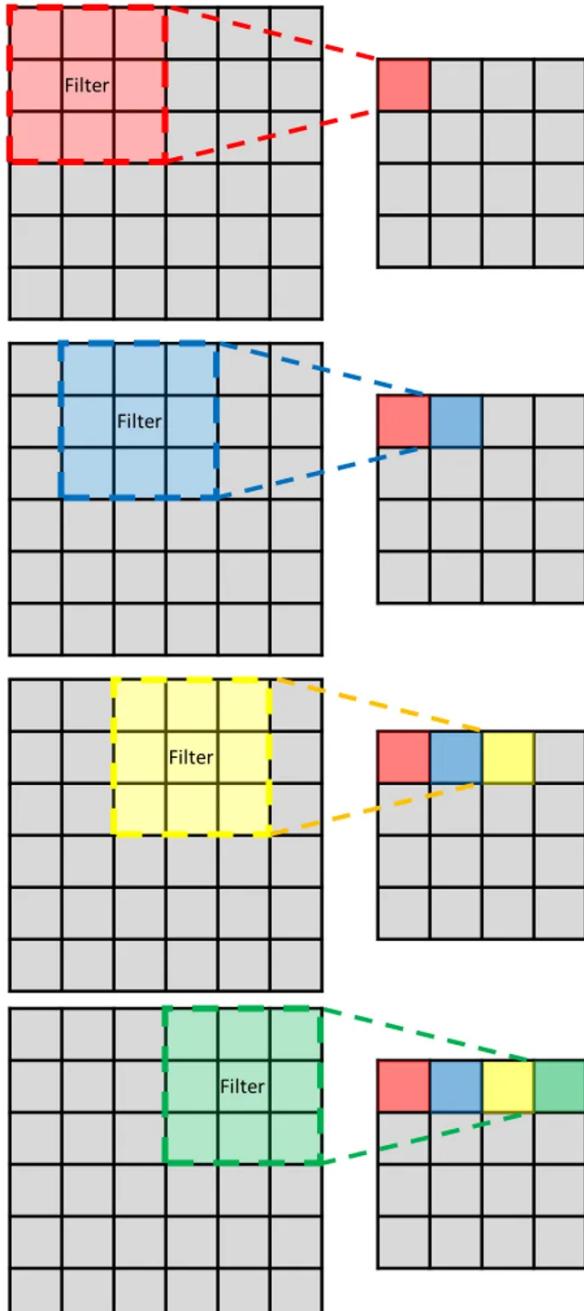
- **Componentes:**
  - **Vizinhança:** normalmente um pequeno retângulo.
  - **Operação:** realizada sobre os pixels da imagem incluídos na vizinhança.
- **Filtragem:** cria um **novo pixel** com coordenadas iguais às coordenadas do **centro** da vizinhança, e cujo valor é o **resultado** da operação de filtragem.
- **Imagen Filtrada** (ou processada): Gerada à medida que o **centro do filtro** percorre cada pixel na imagem de entrada.



Esquema geral da filtragem espacial. Fonte: [visaocomputacional.com.br](http://visaocomputacional.com.br).

Input Image

Resulting Image



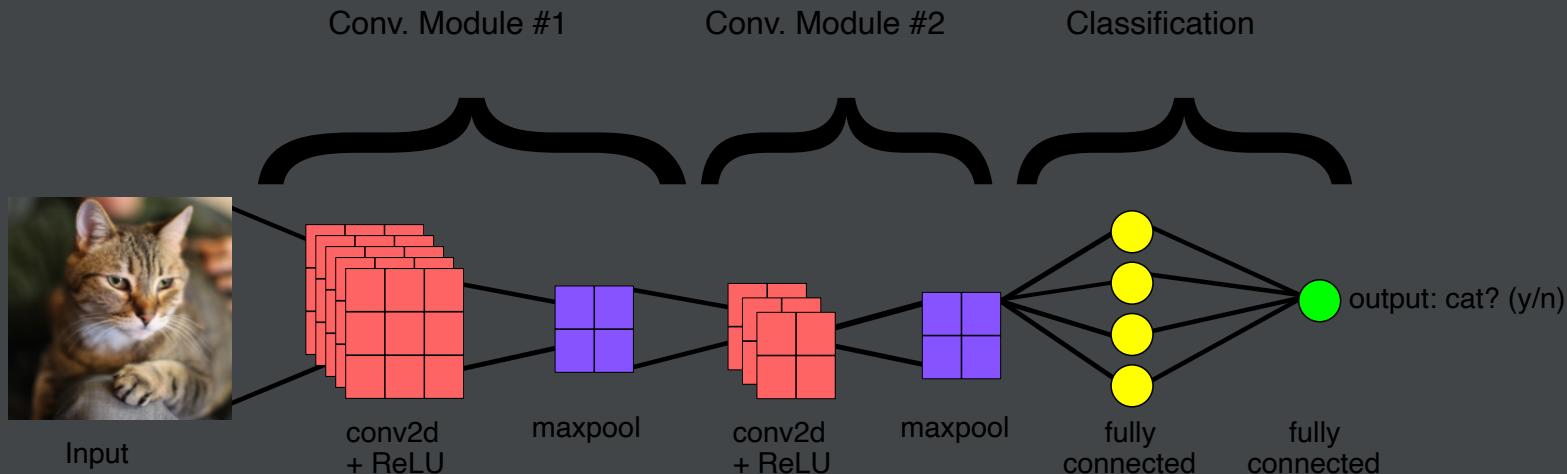
# Convolução

- Operação **fundamental** nesta disciplina.
- **Kernel:** matriz de pesos ( $w_{mn}$ ).
- $g(x, y) = w(x, y) * f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t)f(x - s, y - t)$
- Duas representações geral de um kernel 3 x 3 na **lousa**
  - $w = (w_1, w_2, \dots, w_9)$
  - $w(x, y) = (w(-1, -1), w(-1, 0), \dots, w(1, 1))$

Input	Kernel	Output
$\begin{array}{ c c c } \hline 0 & 1 & 2 \\ \hline 3 & 4 & 5 \\ \hline 6 & 7 & 8 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 0 & 1 \\ \hline 2 & 3 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 19 & 25 \\ \hline 37 & 43 \\ \hline \end{array}$
*		=

Operação de correlação onde as partes sombreadas são o primeiro elemento de saída, bem como os elementos tensores de entrada e kernel usados para o cálculo de saída. Fonte: [Deep Dive into Deep Learning](#).

Convolução é a base para as CNNs,  
**redes neurais convolucionais**,  
e para deep learning.



Exemplo de uma CNN. Fonte: Google Developers.

# Exemplo de Filtragem Espacial

---

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$K$



$F(x, y)$

$K * F(x,y) = \text{filtered, output image}$

Exemplo de aplicação de filtro espacial. Fonte: [Cezanne Camacho](#).

# Exercício Prático: Convolução

- Dados a imagem de entrada  $f$  e o kernel  $w$  abaixo (esquerda), calcule a matriz da imagem resultante da convolução entre  $f$  e  $w$ .
- $g(x, y) = w(x, y) * f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t)f(x - s, y - t)$

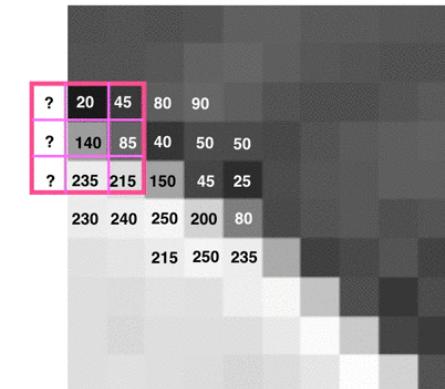
Image Matrix

1	0	0	1	1
1	0	1	1	1
0	1	1	0	1
0	1	0	1	0
1	1	1	0	1

Filter Matrix

1	0	0
1	0	0
0	1	1

0	-1	0
-1	4	-1
0	-1	0



What to do at the edges?

Exemplo de convolução. Fonte: [Austing Walters](#).

Convolução nas bordas de uma imagem. Fonte: [Cezanne Camacho](#).

# Exercício Prático: Convolução (Resolução)

Image Matrix

<b>1=1x1</b>	<b>0=0x0</b>	<b>0=0x0</b>	1	1
<b>1=1x1</b>	<b>0=0x0</b>	<b>0=1x0</b>	1	1
<b>0=0x0</b>	<b>1=1x1</b>	<b>1=1x1</b>	0	1
0	1	0	1	0
1	1	1	0	1

Convolved Matrix

4		

Exemplo de convolução. Fonte: [Austing Walters](#).

# Implementação Prática (OpenCV)

```
import cv2
import numpy as np

# Carregar a imagem em escala de cinza
imagem = cv2.imread("exemplo.jpg", cv2.IMREAD_GRAYSCALE)

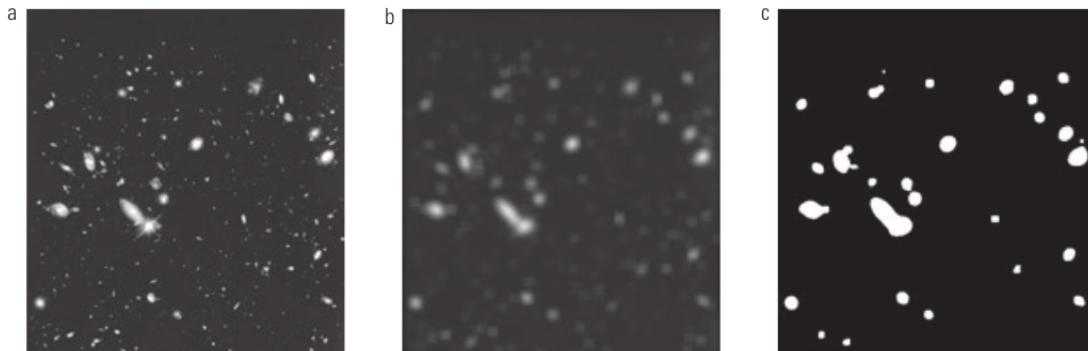
# Definir um kernel de suavização (média)
kernel = np.ones((3,3), np.float32) / 9 # Normalizamos dividindo por 9

# Aplicar a convolução com o kernel
imagem_suavizada = cv2.filter2D(imagem, -1, kernel)

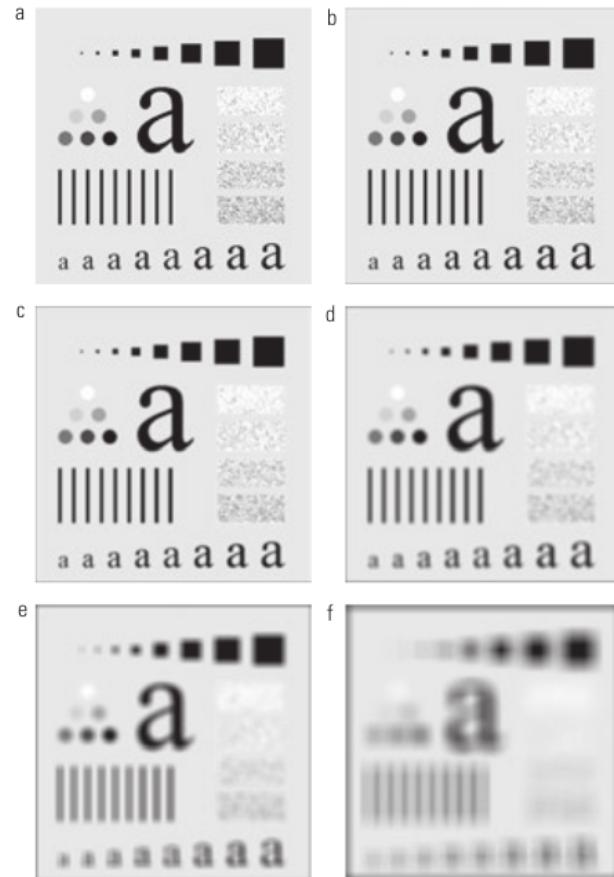
# Mostrar as imagens original e suavizada
cv2.imshow("Original", imagem)
cv2.imshow("Suavizada", imagem_suavizada)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

# Exemplos de Filtros Espaciais

- **Filtro de Média:** Utilizado para suavização da imagem.
- **Diminui a nitidez** pela redução de fortes transições de intensidade entre os pixels.
- **Desfocar os objetos menores** e deixar os maiores em maior evidência.



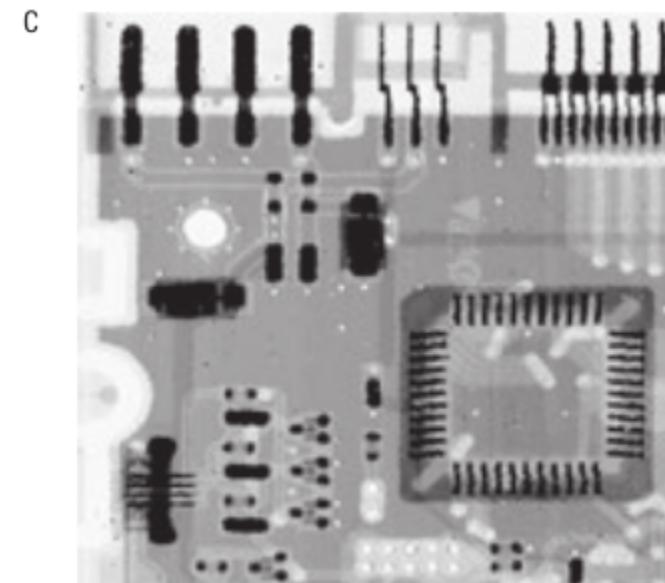
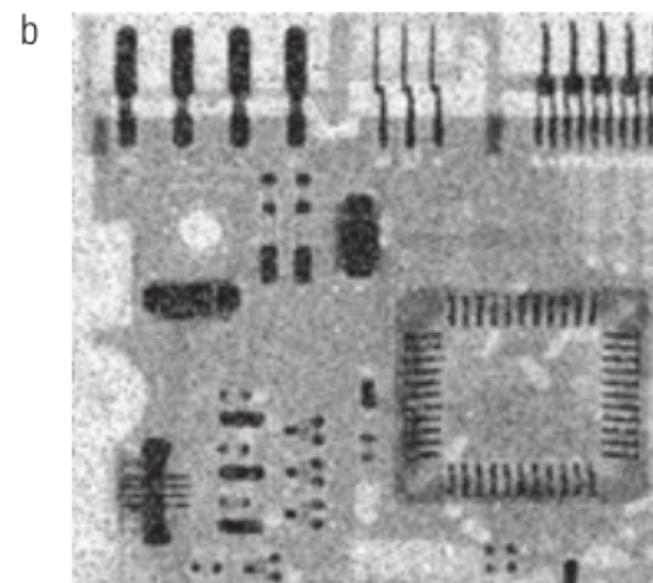
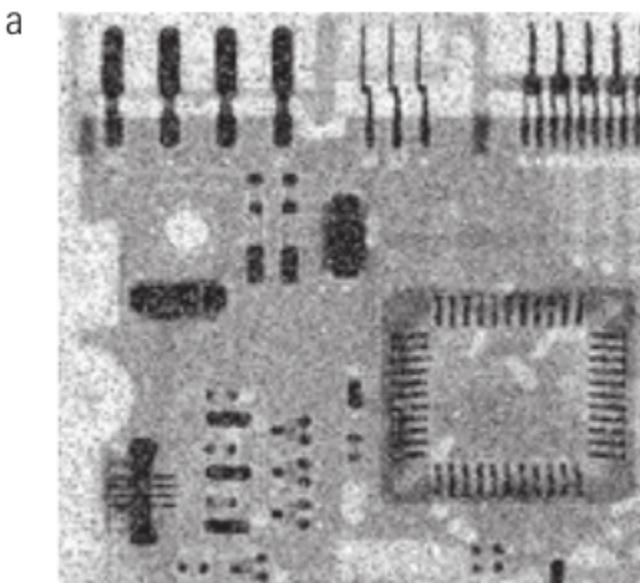
Exemplo de kernel de média 15 x 15 sobre uma imagem capturada pelo telescópio Hubble. Note que o desfoque ajuda a dar mais ênfase aos objetos maiores. A terceira imagem é o resultado de [thresholding](#). Fonte: [COVAP-UTFPR](#).



Exemplo de aplicação de filtro de média.  
Filtros quadrados de tamanhos 3, 5, 9, 15 e 35. Fonte: [COVAP-UTFPR](#)

# Exemplos de Filtros Espaciais (cont.)

- **Filtro de Mediana**: Utilizado para redução de ruído da imagem.
- Elimina **picos isolados** (outliers).
- Substitui o valor de um pixel pela **mediana** dos valores de intensidade na vizinhança desse pixel.
- Note que o valor original do pixel é incluído no cálculo da mediana.



Exemplo de remoção de **ruído sal e pimenta** em imagem de raio X de uma placa de circuito por filtro de mediana. Fonte: [COVAP-UTFPR](#).

# Filtro Gaussiano

- Filtro de média ponderada, mas com seus pesos seguindo uma aproximação da **distribuição gaussiana**.
- $G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$
- $\sigma$  controla o **largura** da distribuição; quanto maior, mais suave a imagem resultante.
- Suaviza a imagem (leve borramento), mas preserva melhor as bordas.
- Indicado para reduzir **ruído branco** e no pré-processamento em detecção de bordas.

$\frac{1}{159}$

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

5x5

## Filtro Gaussiano



Aplicação de filtro gaussiano de diferentes tamanhos. Fonte: [Slide Serve](#).

# Outros de Filtros Espaciais

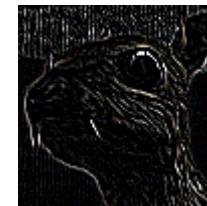
Imagen Original (identidade)

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



Detecção de Borda

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



Exemplo de filtros espaciais. Fonte: [COVAP-UTFPR](#).

Filtro Laplaciano

$$w = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Detecta variação de intensidade (bordas).

Filtro de Sobel

$$w_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \quad w_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Estima gradientes em direções horizontal e vertical.

Demonstração em <https://setosa.io/ev/image-kernels/>

# Classificação de Filtros

---

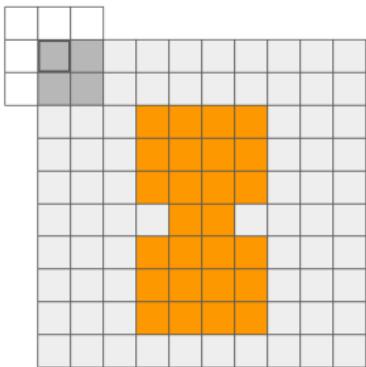
1. **Lineares** – resposta proporcional à entrada (ex.: convolução).
2. **Não-lineares** – dependência de funções não lineares (ex.: mediana).

Tipo	Operação principal	Exemplo
Linear	$y(i, j) = \sum_{m,n} h(m, n) x(i - m, j - n)$	Média, Gaussiano
Não-linear	Função de decisão sobre vizinhança	Max, Mediana

# Extra: Dilatação e Erosão

---

Dilatação



Erosão

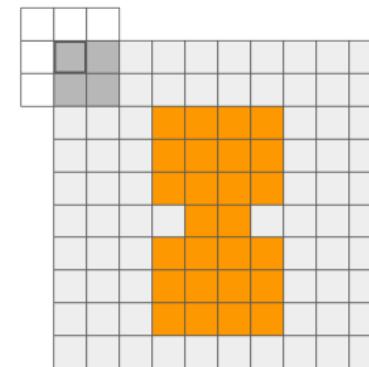
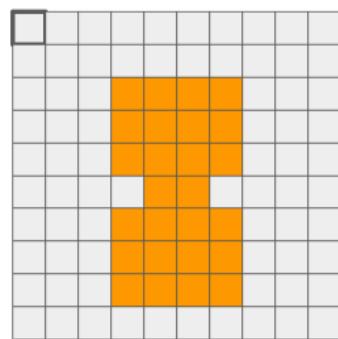
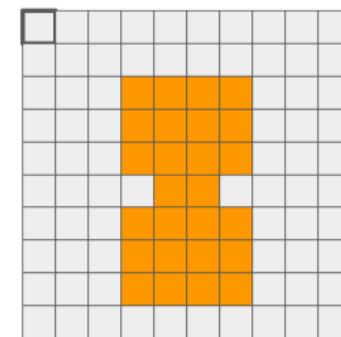
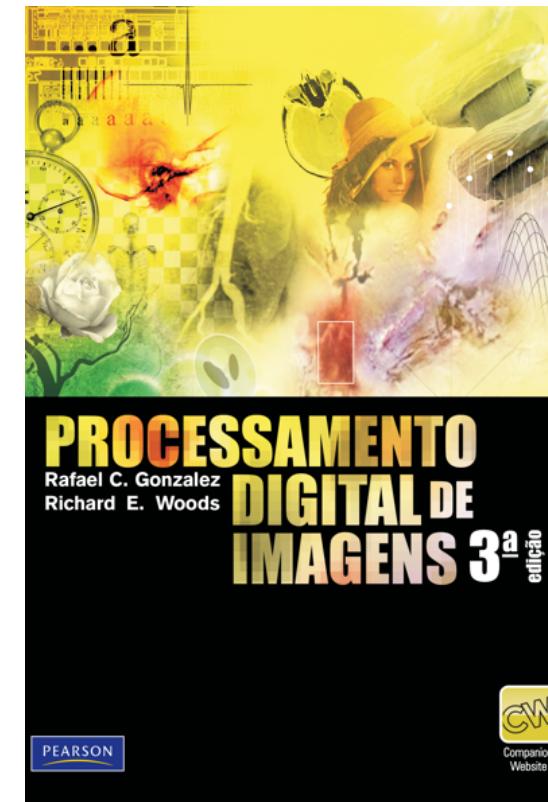


Ilustração de operação de dilatação e erosão via filtros espaciais. Fonte: [Penny Xu](#).

# Resumo e Próximos Passos

---

- **Convolução**: ferramenta matemática central.
- **Filtros lineares** (média, Gaussiano) oferecem suavização controlada; a separabilidade do kernel pode reduzir drasticamente o custo computacional.
- **Filtros não-lineares** (mediana, adaptativos) são essenciais para robustez contra ruído impulsivo e preservação de bordas em cenários adversos.
- A escolha entre **suavização** e **realce** depende da aplicação: imagens médicas, visão robótica ou processamento de vídeo em tempo real.
- **Próximos passos**: Implementar filtros e analisar resultados (prática).



**Atividade recomendada:** Leitura das seções 3.4 a 3.7.

# Perguntas e Discussão

---

- Cite **três aplicações** industriais ou científicas onde a filtragem espacial desempenha um papel crítico e explique por que o filtro escolhido foi adequado ao problema.
- Como você aplicaria filtros espaciais para **melhorar a qualidade** de imagens médicas (ex.: tomografia, ressonância magnética) **sem introduzir artefatos** que possam comprometer o diagnóstico?
- Quais são as **principais limitações** dos filtros lineares quando se trata de preservar detalhes finos em imagens de alta resolução?
- Como os filtros não-lineares, como a mediana, podem **alterar a distribuição** de intensidade da imagem e quais implicações isso tem para tarefas subsequentes (segmentação, classificação)?