



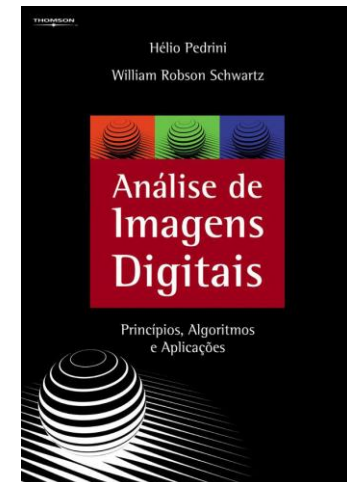
VISÃO COMPUTACIONAL

MORFOLOGIA MATEMÁTICA

Prof. Msc. Giovanni Lucca França da Silva
E-mail: giovanni-lucca@live.com

SOBRE A DISCIPLINA

- Bibliografia principal:
 - GONZALEZ, Rafael C.; WOODS, Richard C. **Processamento digital de imagens.** Pearson, 2011.
- Bibliografia complementar:
 - PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações.** Thomson Learning, 2008.



NA AULA PASSADA...

- Realces.

ROTEIRO

- Introdução.
- Morfologia Matemática.

INTRODUÇÃO

- Morfologia.
 - Ramo da biologia que lida com a forma e a estrutura dos animais e plantas.
- No contexto da morfologia matemática:
 - Ferramenta para extrair componentes das imagens, úteis na representação e na descrição da forma de uma região.
 - Formulada na década de 60 por Matheron e Serra na França.

INTRODUÇÃO

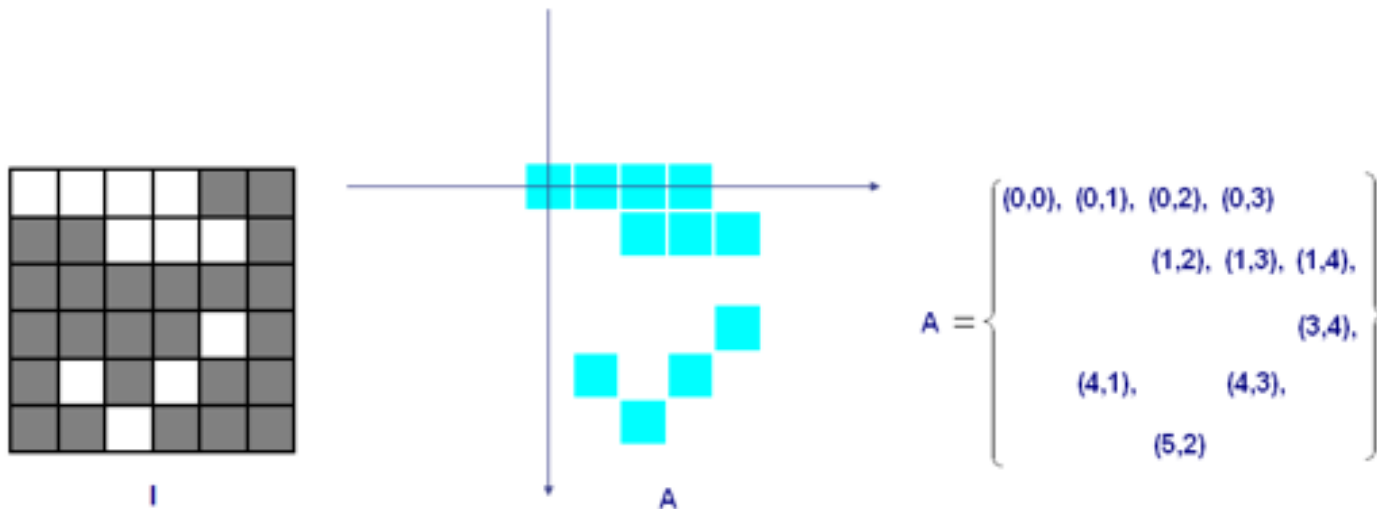
- Aplicações:
 - Realce.
 - Segmentação.
 - Restauração.
 - Detecção de bordas.
 - Extração de características.
 - Compressão.

INTRODUÇÃO

- A morfologia matemática utiliza a teoria dos conjuntos para representar a forma dos objetos de uma imagem.
- Teoria dos conjuntos:
 - Seja A um conjunto contido em \mathbb{Z}^2 .
 - Quando $a = (a_1, a_2)$ é um elemento de A , escreve-se: $a \in A$.
 - Quando a não é elemento de A , escreve-se: $a \notin A$.
 - Quando A não possui elementos, diz-se que ele é um conjunto vazio.

INTRODUÇÃO

- Dessa forma, uma imagem binária pode ser considerada uma coleção de coordenadas discretas que correspondem aos pontos pertencentes aos objetos na imagem.



INTRODUÇÃO

- Principais operações na teoria dos conjuntos.
 - Sejam A e B duas imagens binárias no espaço Z^2 .
 - A: o conjunto de coordenadas dos pixels.
 - A': complemento de A.
 - $A \cup B$: união (operação OR).
 - $A \cap B$: interseção (operação AND).
 - $A - B$: conjunto de todos os elementos que pertencem a A, mas não pertencem a B.
 - Translação: deslocamento de A mediante a um elemento p.
 - Reflexão: rotação de A em 180 graus baseado na origem.

INTRODUÇÃO

- Principais operações na teoria dos conjuntos.
 - Complemento:
 - Imagem inversa.
 - $A^c = \{p \mid p \notin A\}$.

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

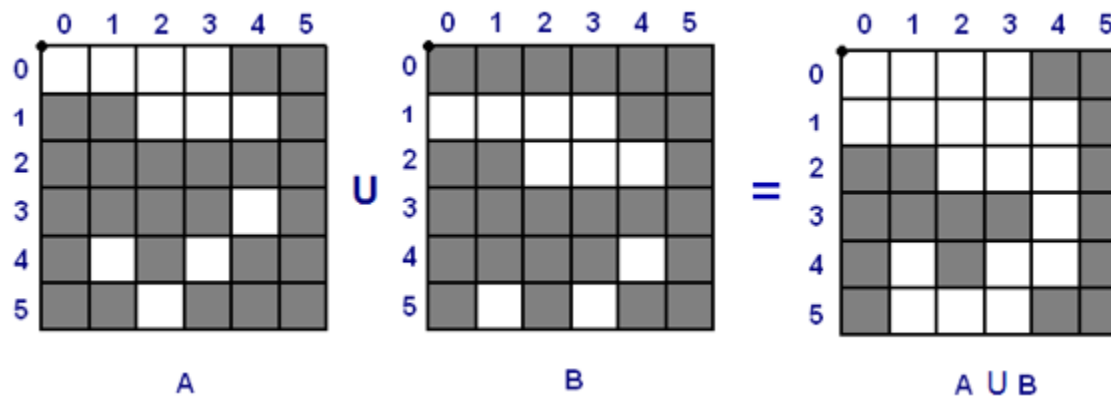
A

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

A^c

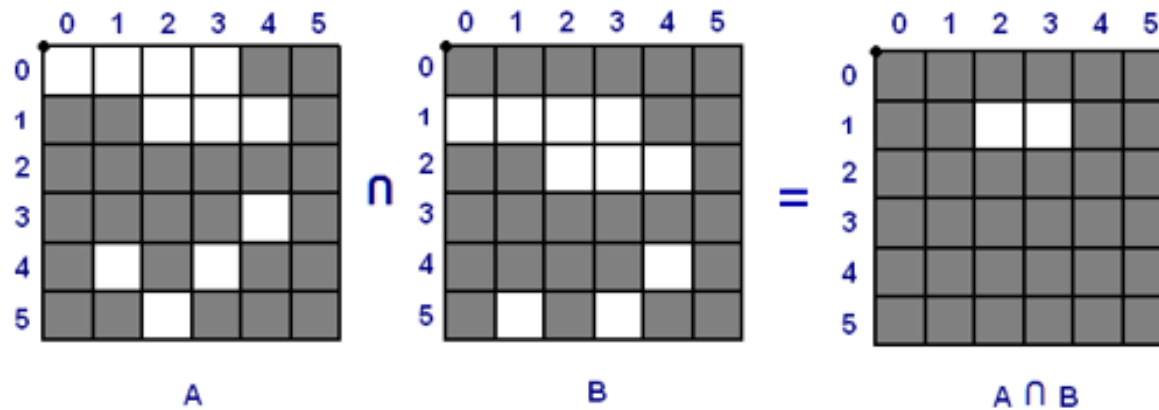
INTRODUÇÃO

- Principais operações na teoria dos conjuntos.
 - União:
 - Operação OR entre duas imagens.
 - $A \cup B = \{c \mid c \in A \text{ ou } c \in B\}$.



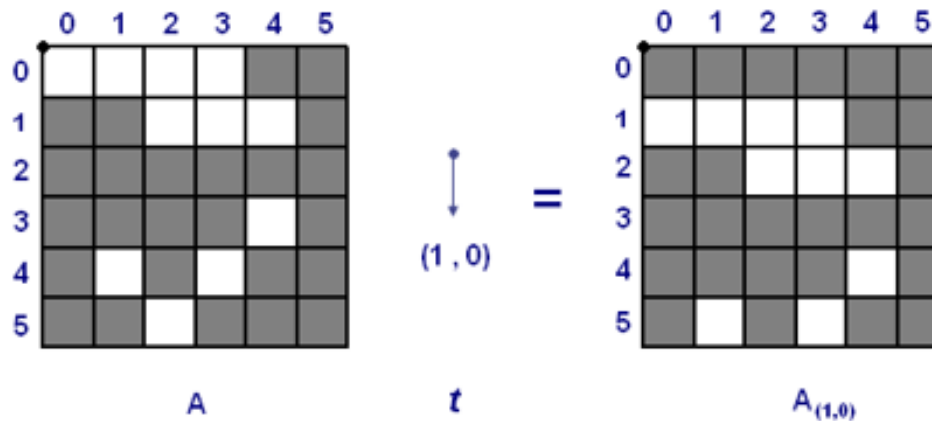
INTRODUÇÃO

- Principais operações na teoria dos conjuntos.
 - Interseção:
 - Operação AND entre duas imagens.
 - $A \cap B = \{c \mid c \in A \text{ e } c \in B\}$.



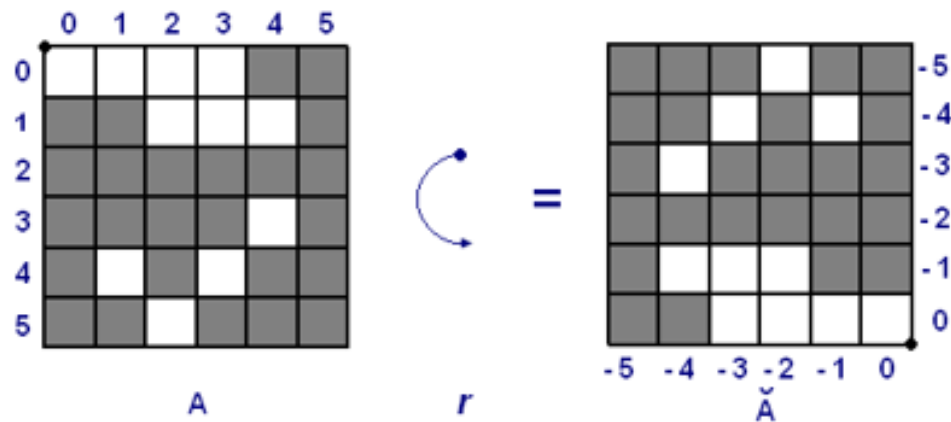
INTRODUÇÃO

- Principais operações na teoria dos conjuntos.
 - Translação:
 - Deslocamento de A mediante a um elemento p.
 - $A^t = \{ c \mid c = a + t, \text{ para todos os } a \in A \}.$



INTRODUÇÃO

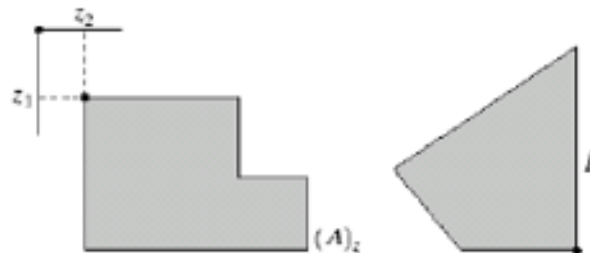
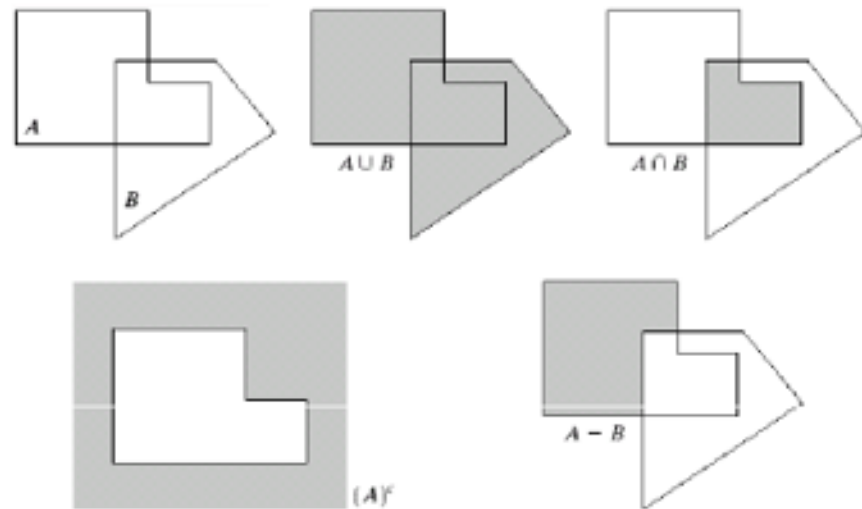
- Principais operações na teoria dos conjuntos.
 - Reflexão:
 - Rotação de A em 180 graus baseado na origem.
 - $A = \{c \mid c = -a, \text{ para todos os } a \in A\}.$



INTRODUÇÃO

- Principais operações na teoria dos conjuntos.

- Complemento.
- União.
- Interseção.
- Diferença.
- Translação.
- Reflexão.



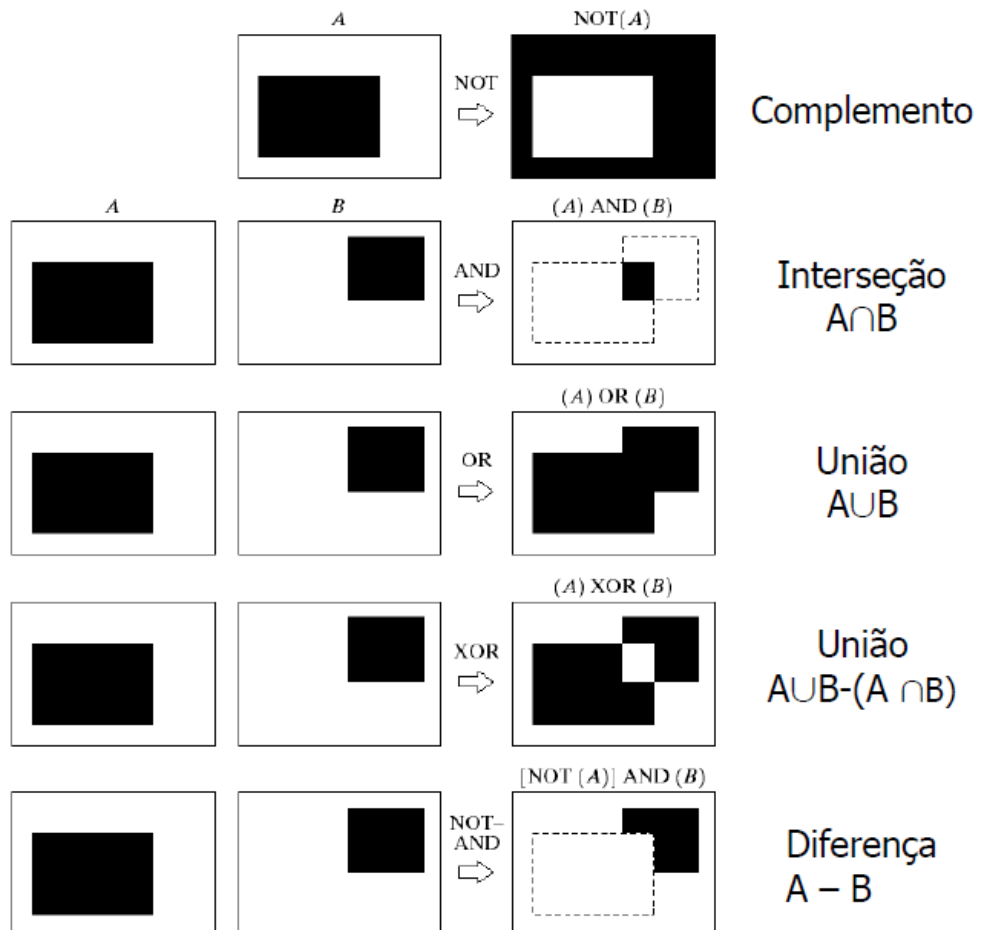
INTRODUÇÃO

- Operações lógicas.
 - AND, OR, NOT.

p	q	p AND q ($p \cdot q$, $p \wedge q$)	p OR q ($p + q$, $p \vee q$)	NOT p ($\neg p$)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

INTRODUÇÃO

- Operações lógicas.
 - Imagens binárias.

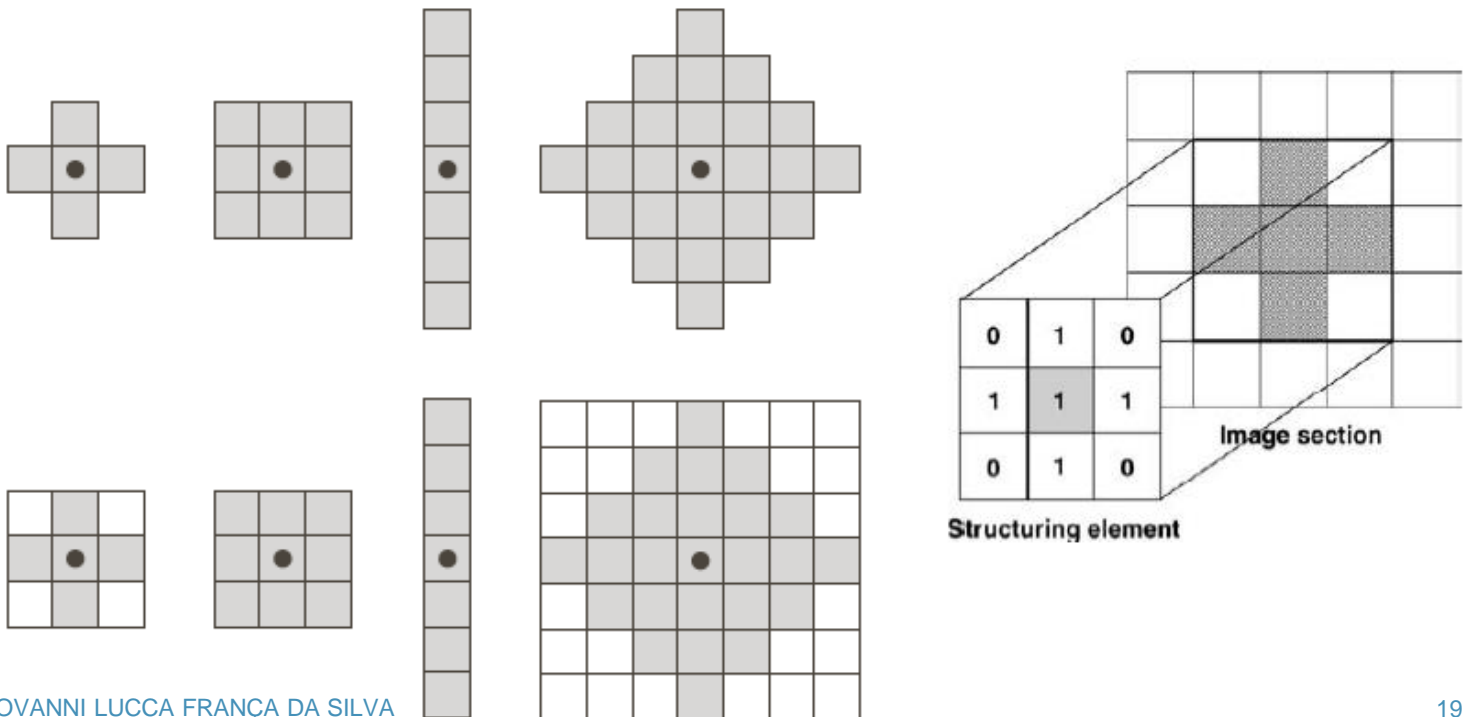


MORFOLOGIA MATEMÁTICA

- Um operador morfológico é um mapeamento entre o conjunto A que define a imagem e um conjunto B , chamado elemento estruturante, também definido em Z^2 .
- Elemento estruturante é uma matriz retangular de pixels que contém os valores 0 ou 1.
 - Expresso com respeito a uma origem local.
 - A forma e o tamanho devem ser adaptados para as propriedades geométricas dos objetos.

MORFOLOGIA MATEMÁTICA

- Exemplos de elementos estruturantes.



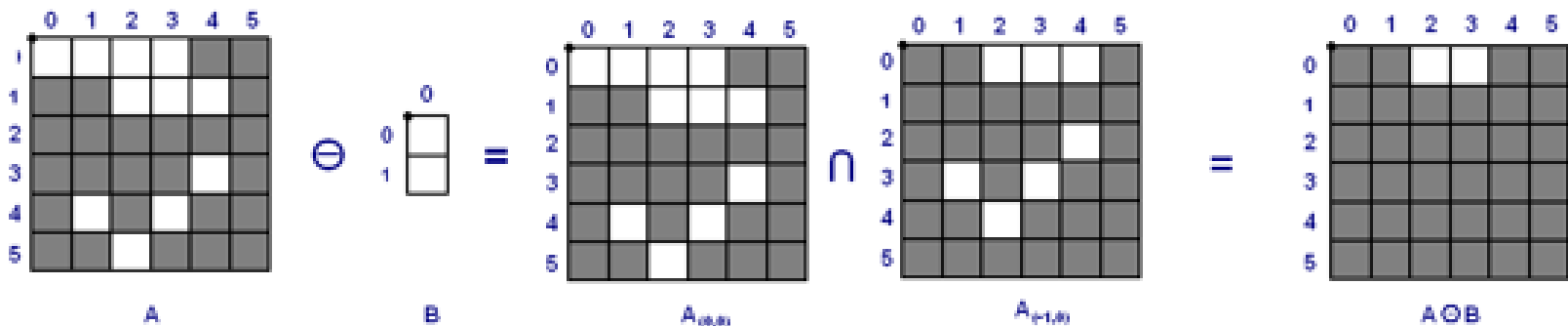
MORFOLOGIA MATEMÁTICA

- Operadores morfológicos:
 - Erosão.
 - Dilatação.
 - Abertura.
 - Fechamento.
 - Esqueletização.
- Todas as operações morfológicas podem ser definidas em termos da erosão e dilatação.

MORFOLOGIA MATEMÁTICA

■ Erosão.

- Sejam A e B dois conjuntos, em que $A \subseteq \mathbb{Z}^2$ e $B \subseteq \mathbb{Z}^2$.
- A erosão do conjunto A pelo conjunto B é expresso por $A \ominus B = \{x \mid x + b \in A, \text{ para todo } b \in B\}$.

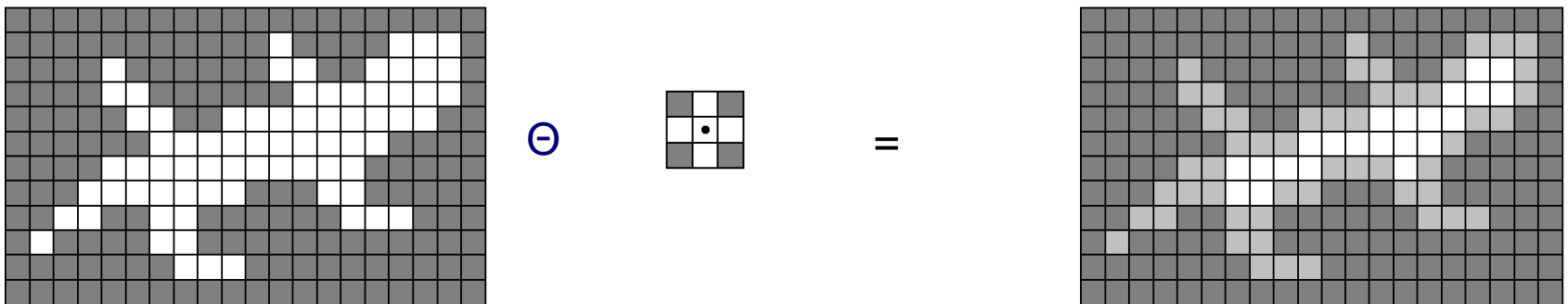
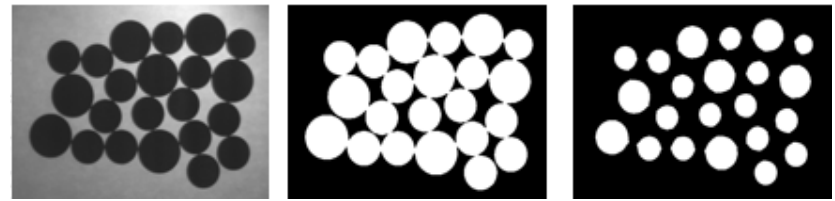


MORFOLOGIA MATEMÁTICA

■ Erosão.

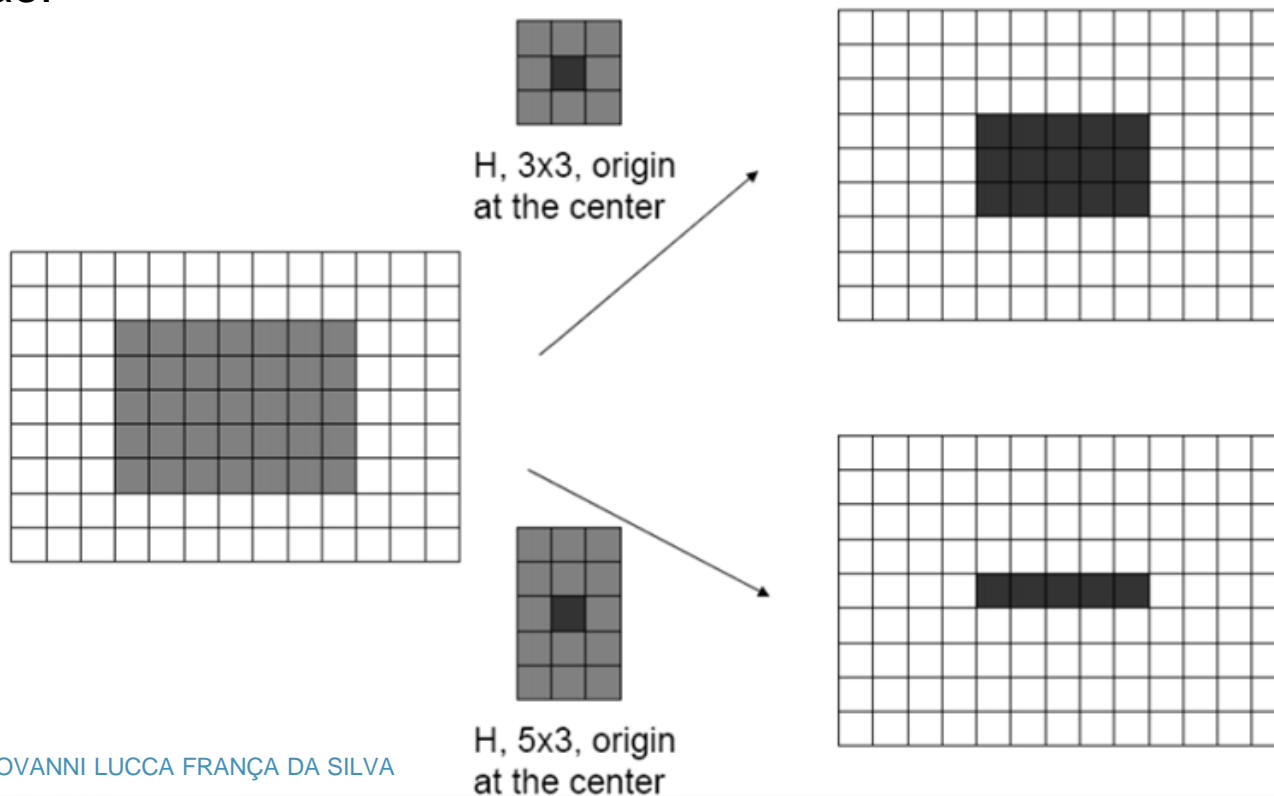
■ Aplicações:

- Separa regiões conectadas.
- Remove pequenos detalhes.
- Remove todas as estruturas menores que o elemento estruturante.



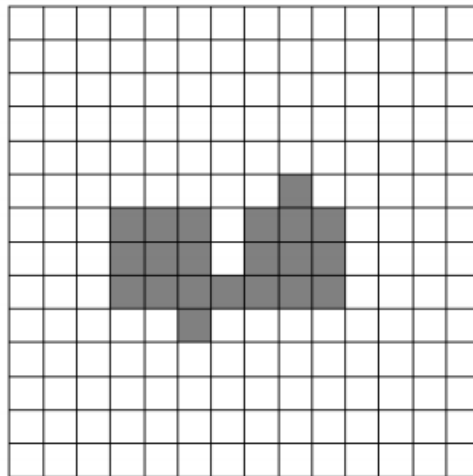
MORFOLOGIA MATEMÁTICA

- Erosão.

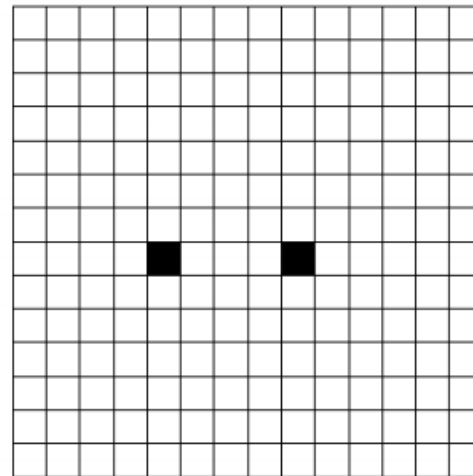


MORFOLOGIA MATEMÁTICA

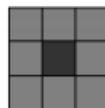
- Erosão.



F



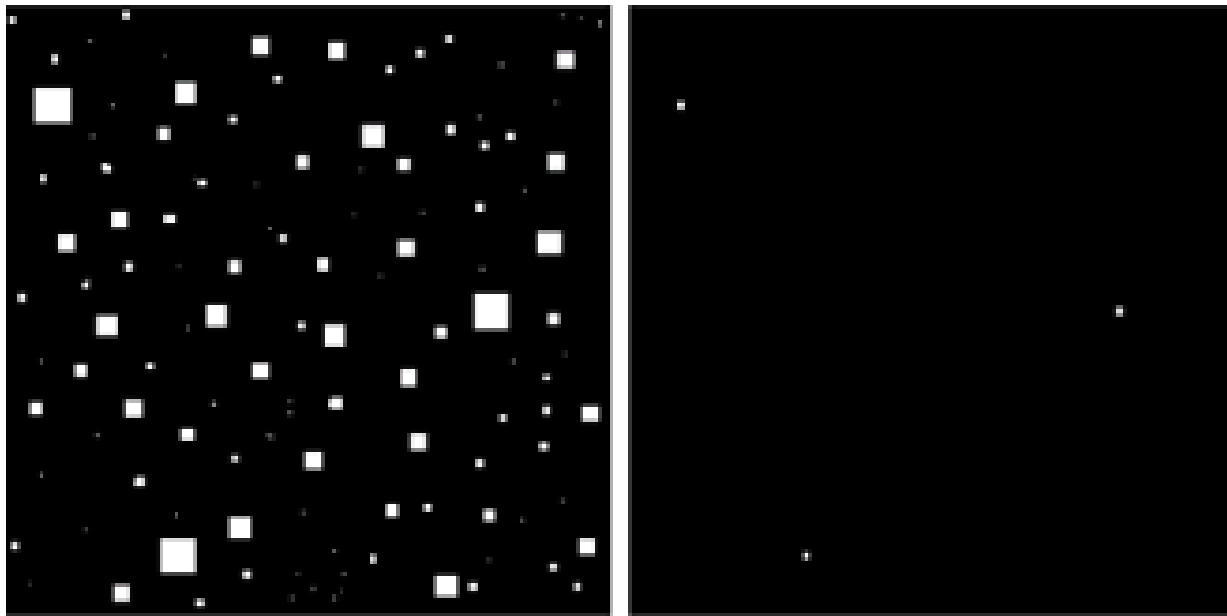
G



H, 3x3, origin at the center

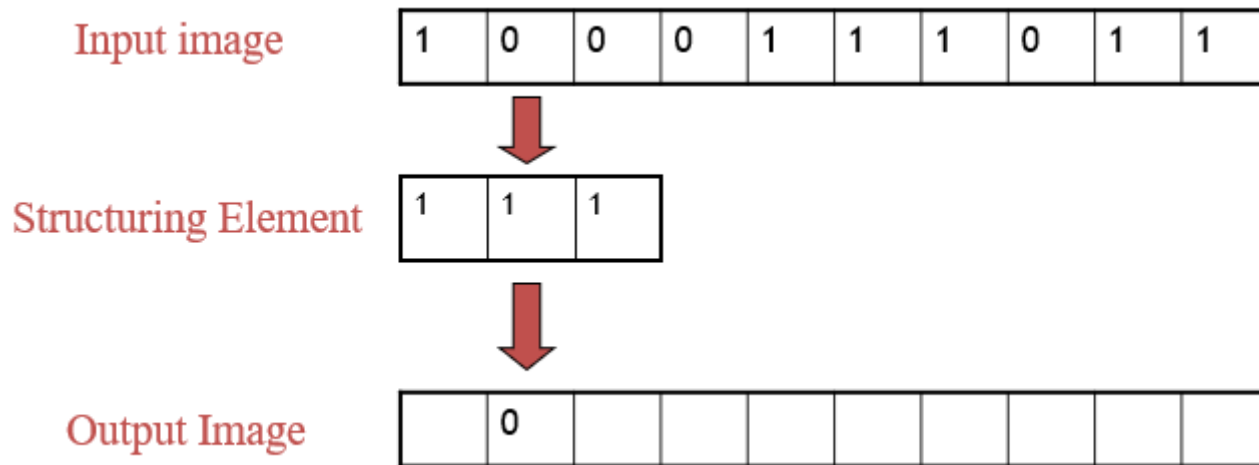
MORFOLOGIA MATEMÁTICA

- Erosão.



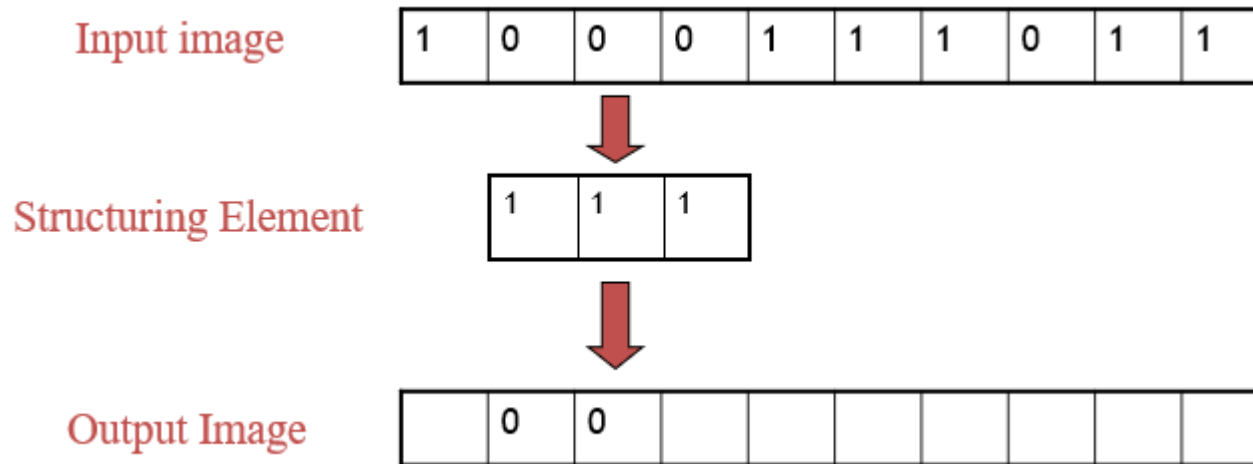
MORFOLOGIA MATEMÁTICA

- Erosão.



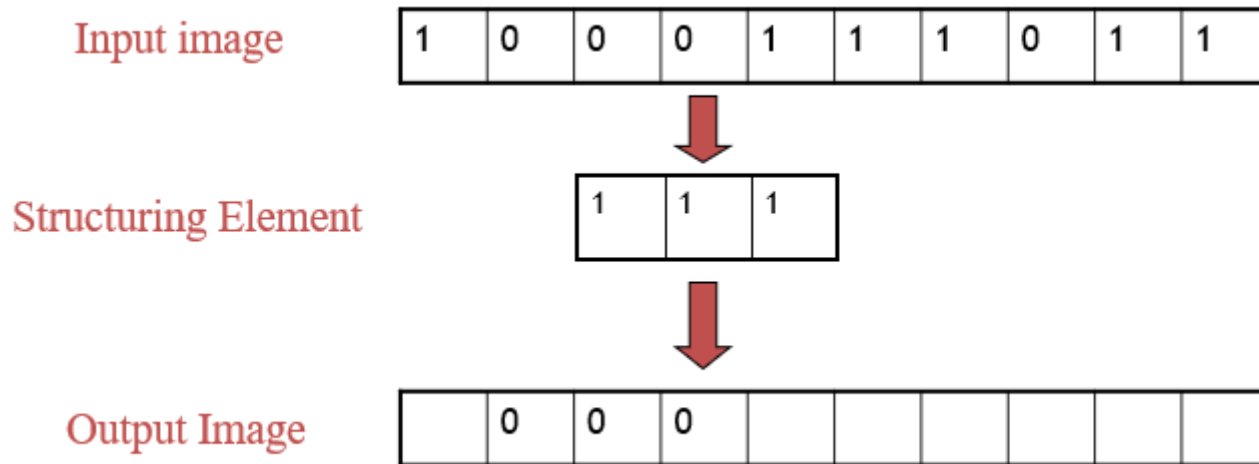
MORFOLOGIA MATEMÁTICA

- Erosão.



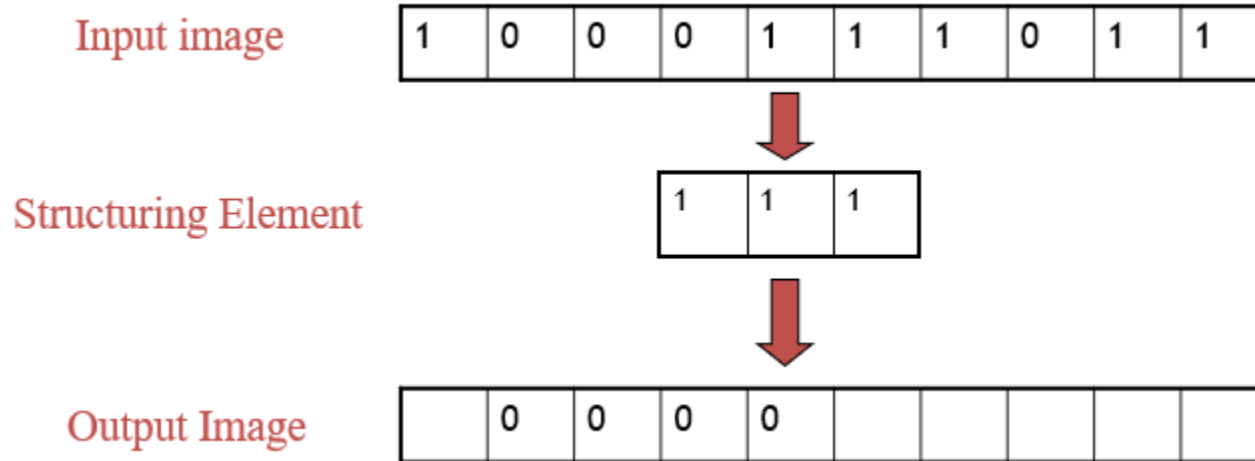
MORFOLOGIA MATEMÁTICA

- Erosão.



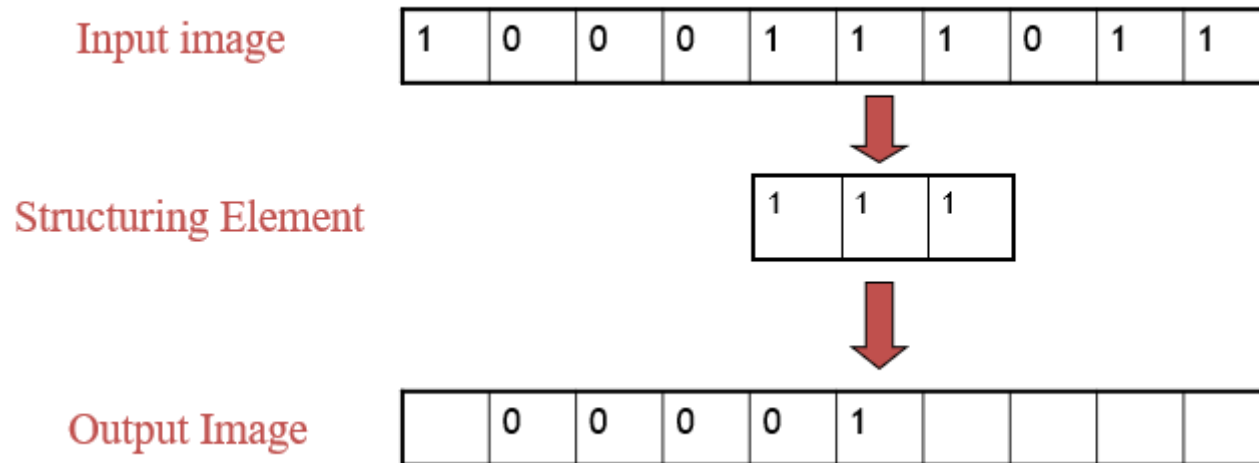
MORFOLOGIA MATEMÁTICA

- Erosão.



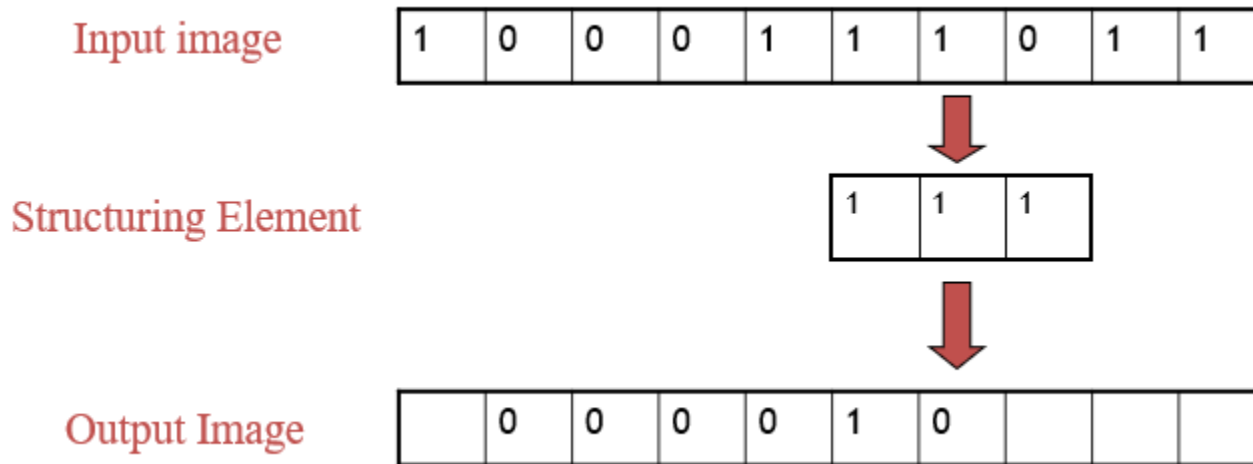
MORFOLOGIA MATEMÁTICA

- Erosão.



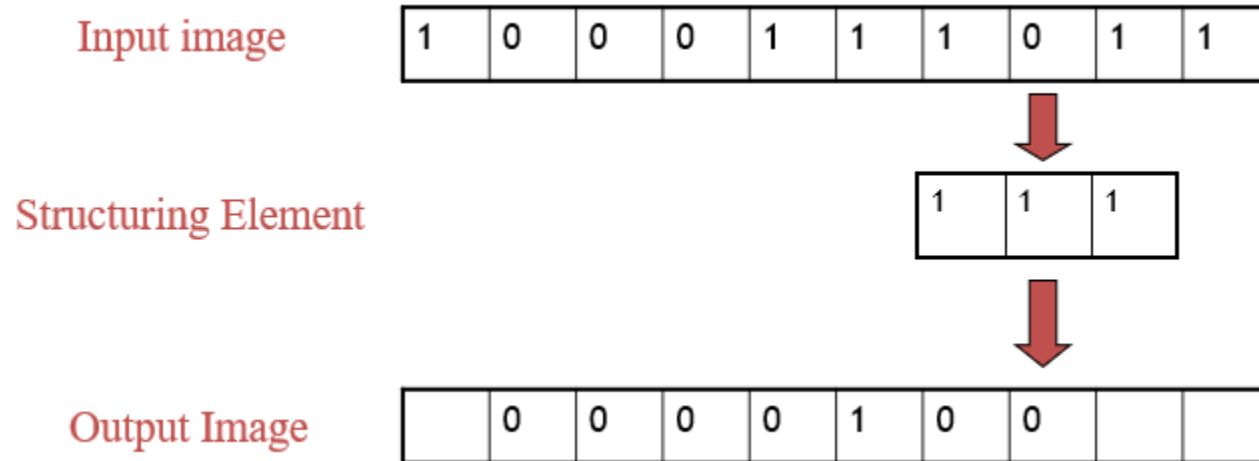
MORFOLOGIA MATEMÁTICA

- Erosão.



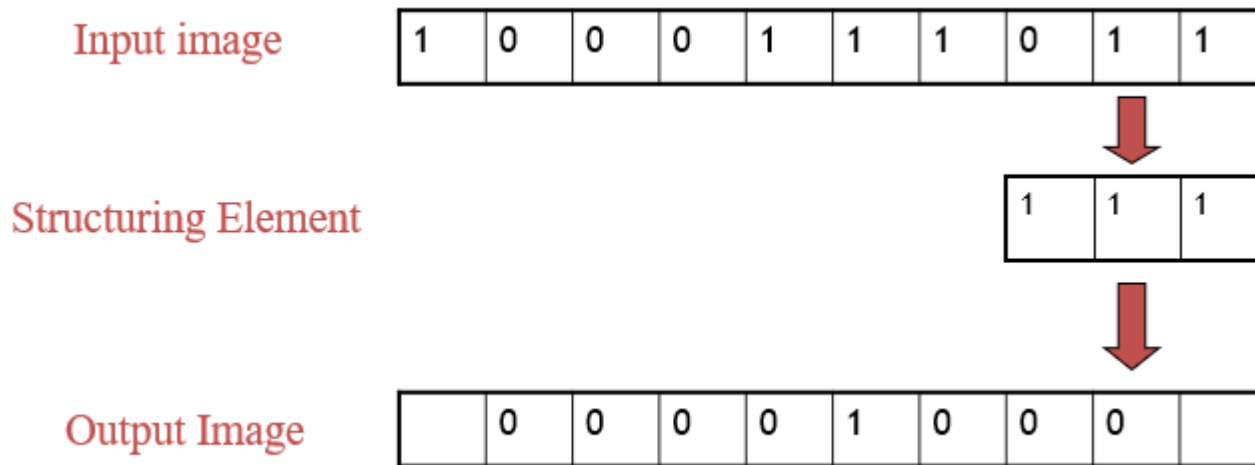
MORFOLOGIA MATEMÁTICA

- Erosão.



MORFOLOGIA MATEMÁTICA

- Erosão.



MORFOLOGIA MATEMÁTICA

- Erosão.
 - Código no OpenCV.

C++: `void erode(InputArray src, OutputArray dst, InputArray kernel, Point anchor=Point(-1,-1), int iterations=1, int borderType=BORDER_CONSTANT, const Scalar& borderValue=morphologyDefaultBorderValue())`

Python: `cv2.erode(src, kernel[, dst[, anchor[, iterations[, borderType[, borderValue]]]]) → dst`

Parameters:

- **src** - input image; the number of channels can be arbitrary, but the depth should be one of CV_8U, CV_16U, CV_16S, CV_32F or CV_64F.
- **dst** - output image of the same size and type as **src**.
- **kernel** - structuring element used for erosion; if `element=Mat()`, a 3 x 3 rectangular structuring element is used. Kernel can be created using `getStructuringElement()`.
- **anchor** - position of the anchor within the element; default value (-1, -1) means that the anchor is at the element center.
- **iterations** - number of times erosion is applied.
- **borderType** - pixel extrapolation method (see `borderInterpolate` for details).
- **borderValue** - border value in case of a constant border

MORFOLOGIA MATEMÁTICA

- Erosão.
 - Exemplo de código no OpenCV.

```
import cv2
import numpy as np

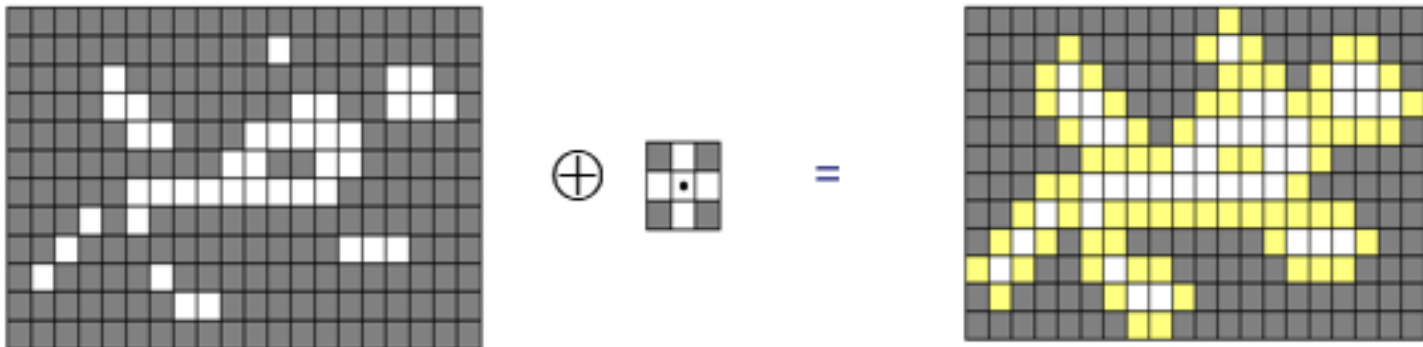
img = cv2.imread('j.png',0)
kernel = np.ones((5,5),np.uint8)
erosion = cv2.erode(img,kernel,iterations = 1)
```



MORFOLOGIA MATEMÁTICA

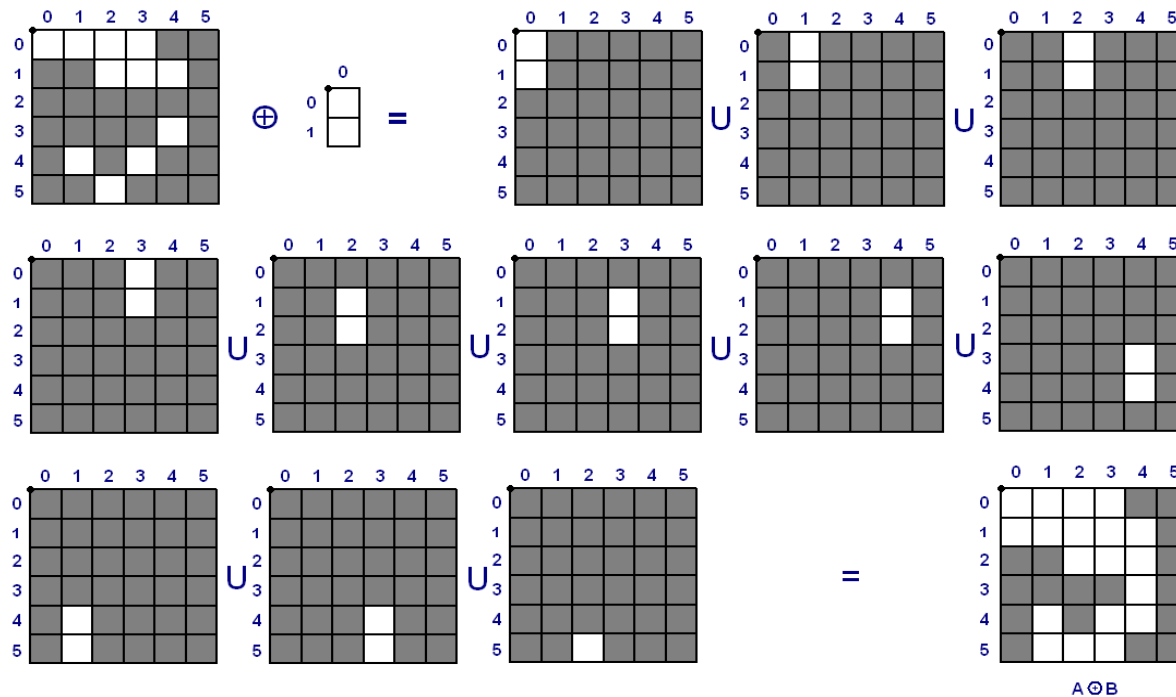
- Dilatação.

- Sejam A e B dois conjuntos, em que $A \subseteq \mathbb{Z}^2$ e $B \subseteq \mathbb{Z}^2$.
- A dilatação do conjunto A pelo conjunto B é expresso por $A \oplus B = \{x \mid x = a + b, a \in A, b \in B\}$.



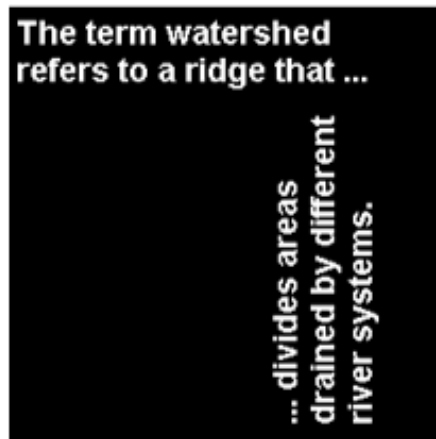
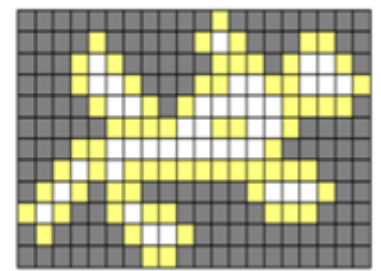
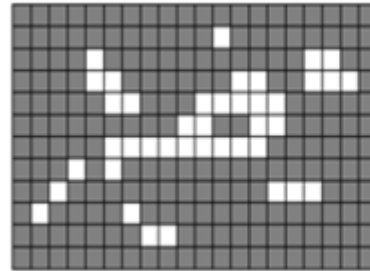
MORFOLOGIA MATEMÁTICA

■ Dilatação.



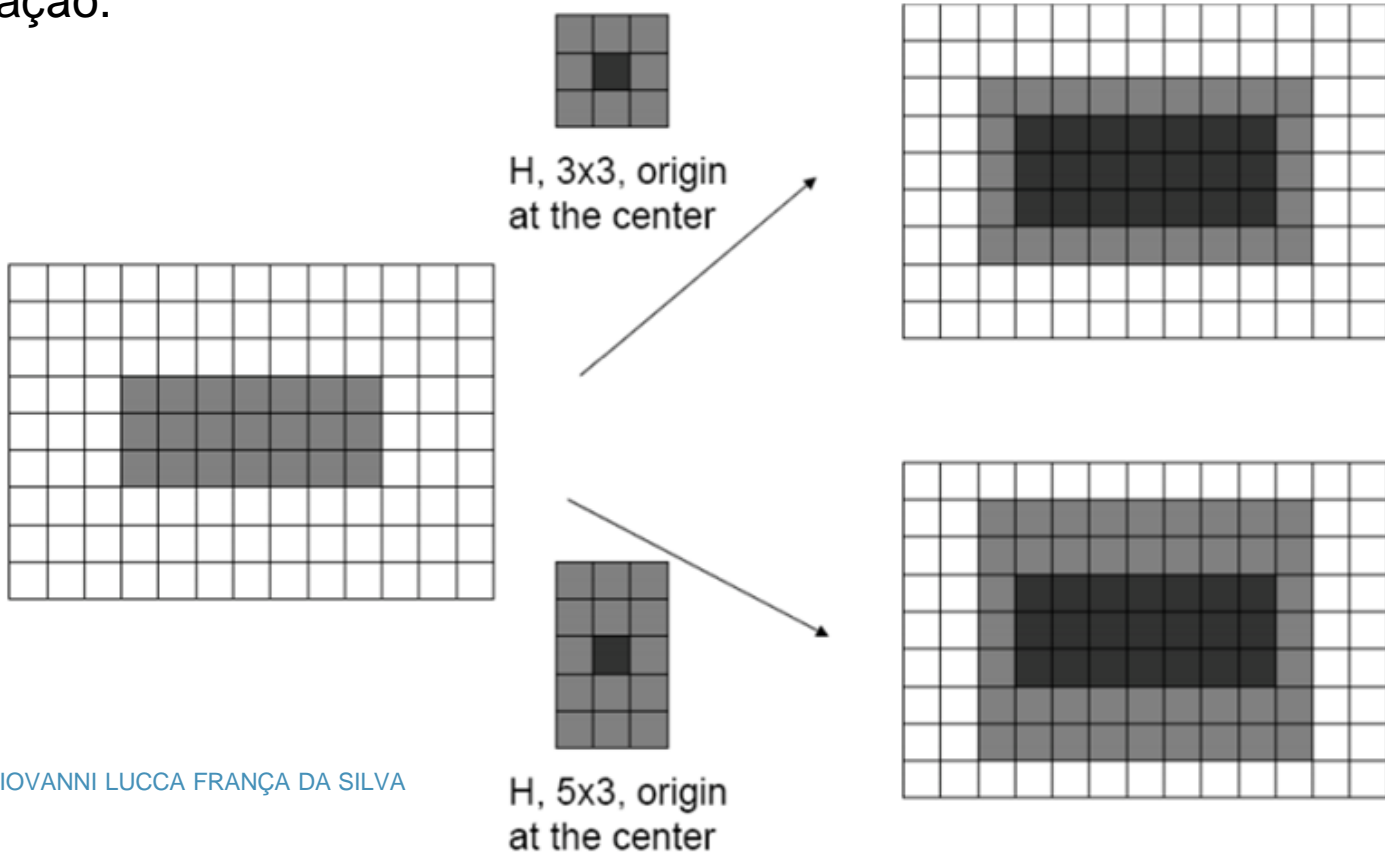
MORFOLOGIA MATEMÁTICA

- Dilatação.
 - Aplicações:
 - Une regiões separadas.
 - Preenche ou amplifica contornos.



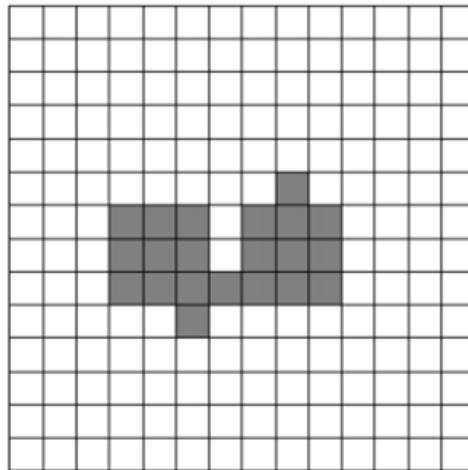
MORFOLOGIA MATEMÁTICA

- Dilatação.

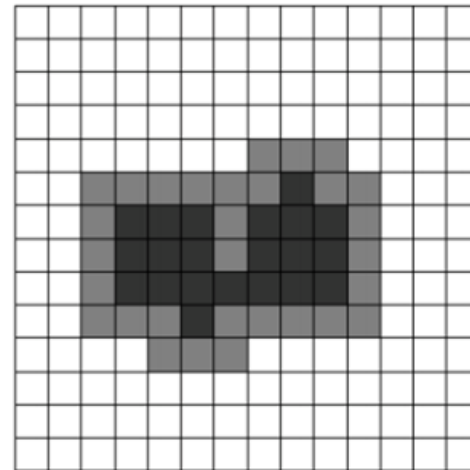


MORFOLOGIA MATEMÁTICA

- Dilatação.



F



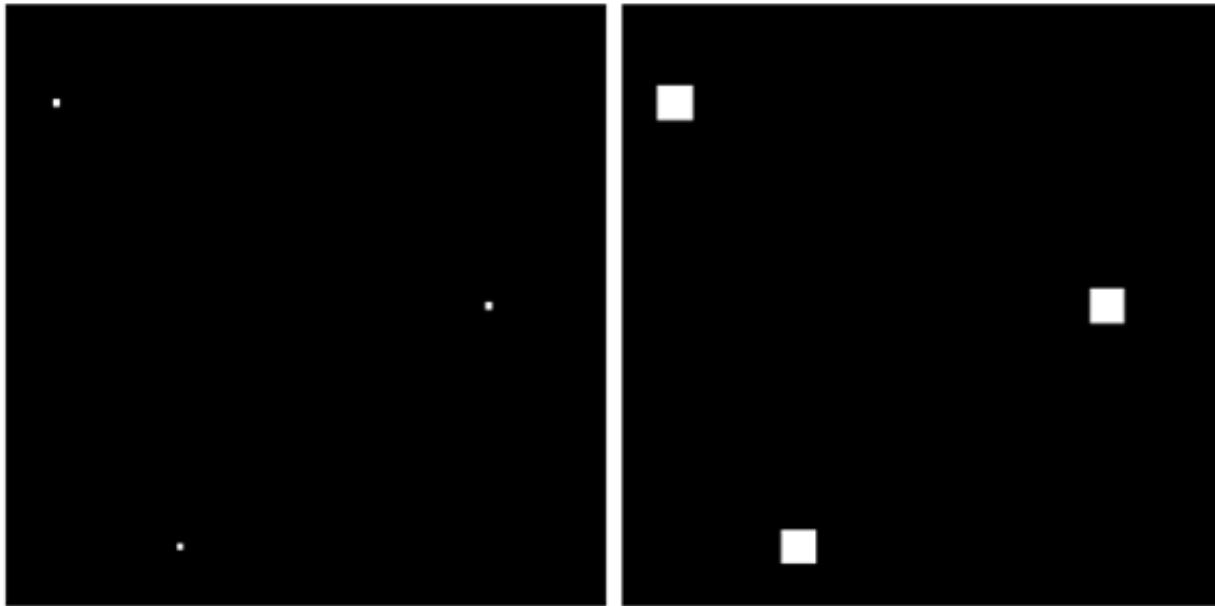
G



H, 3x3, origin at the center

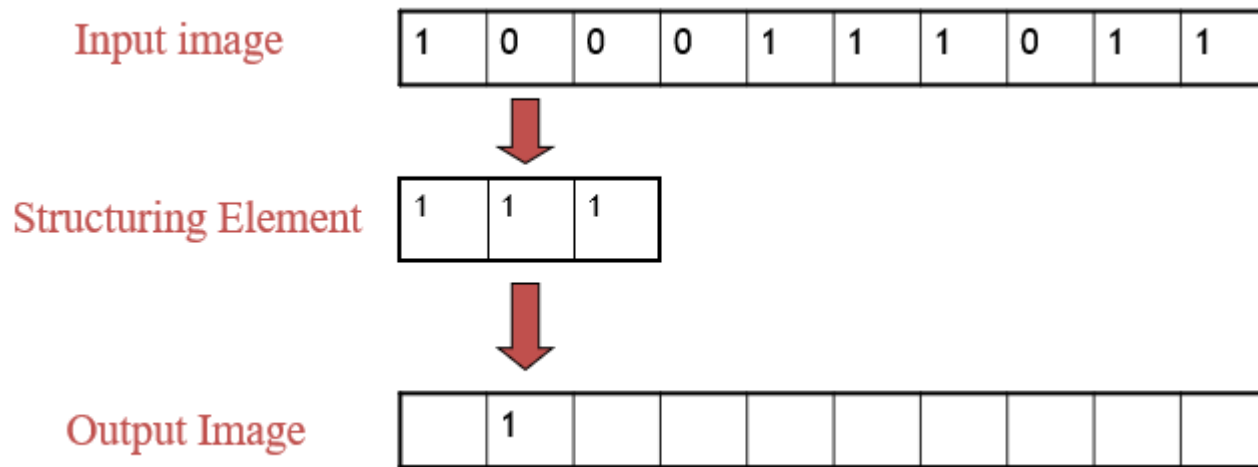
MORFOLOGIA MATEMÁTICA

- Dilatação.



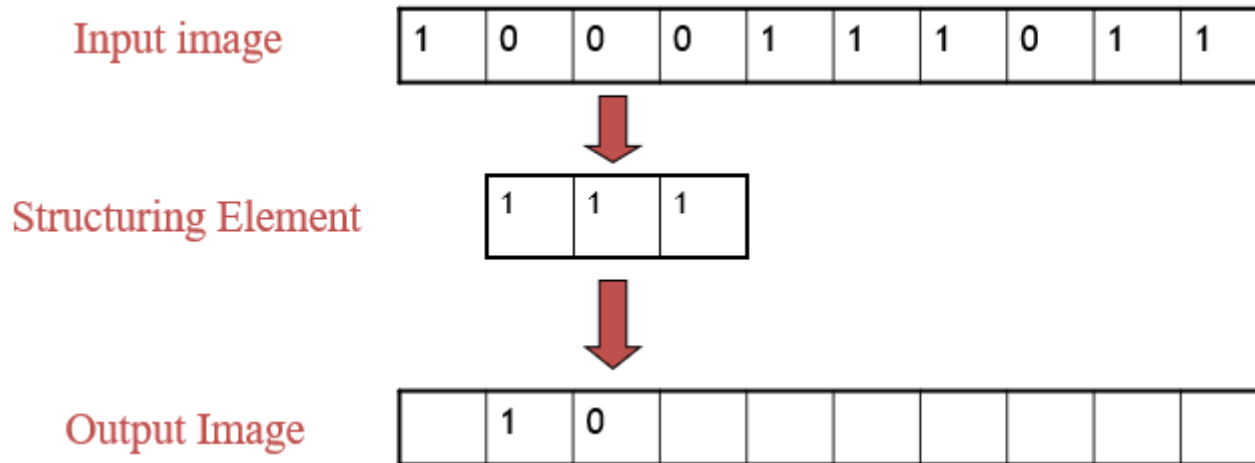
MORFOLOGIA MATEMÁTICA

- Dilatação.



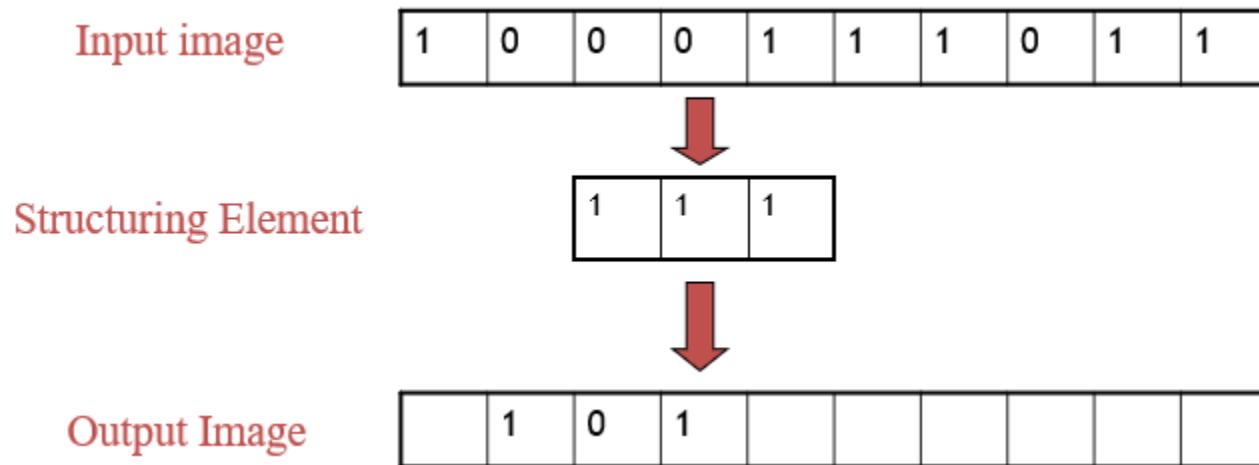
MORFOLOGIA MATEMÁTICA

- Dilatação.



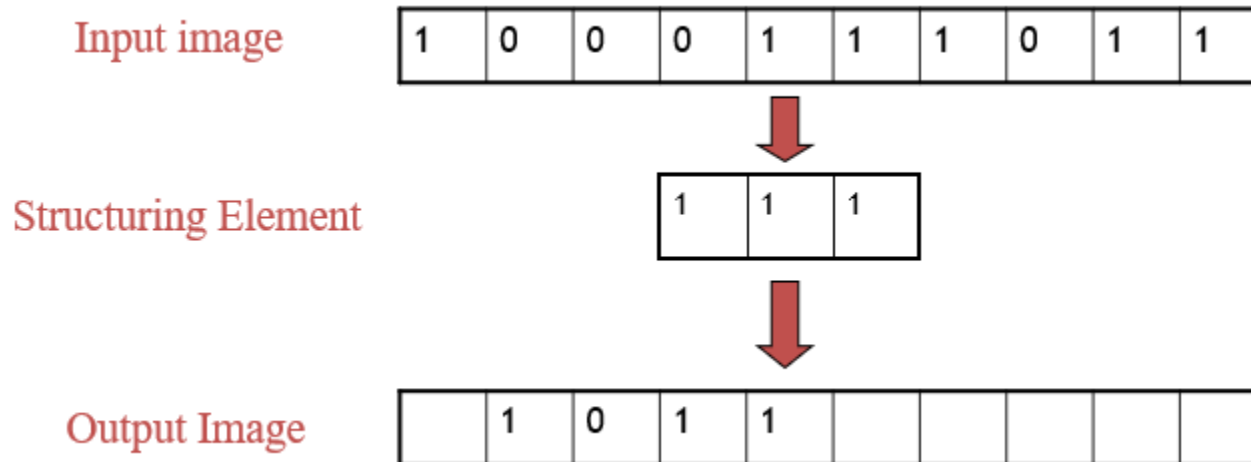
MORFOLOGIA MATEMÁTICA

- Dilatação.



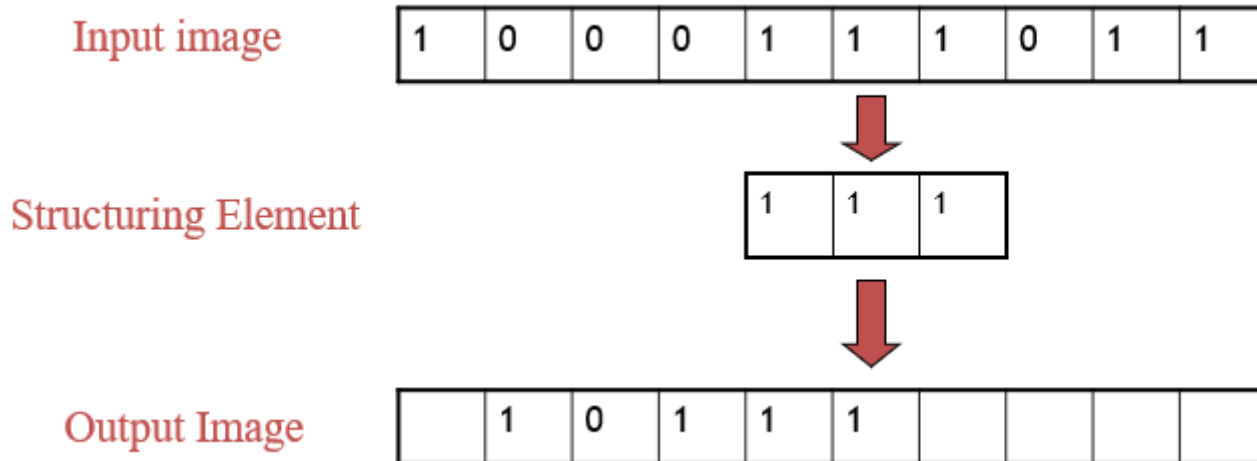
MORFOLOGIA MATEMÁTICA

- Dilatação.



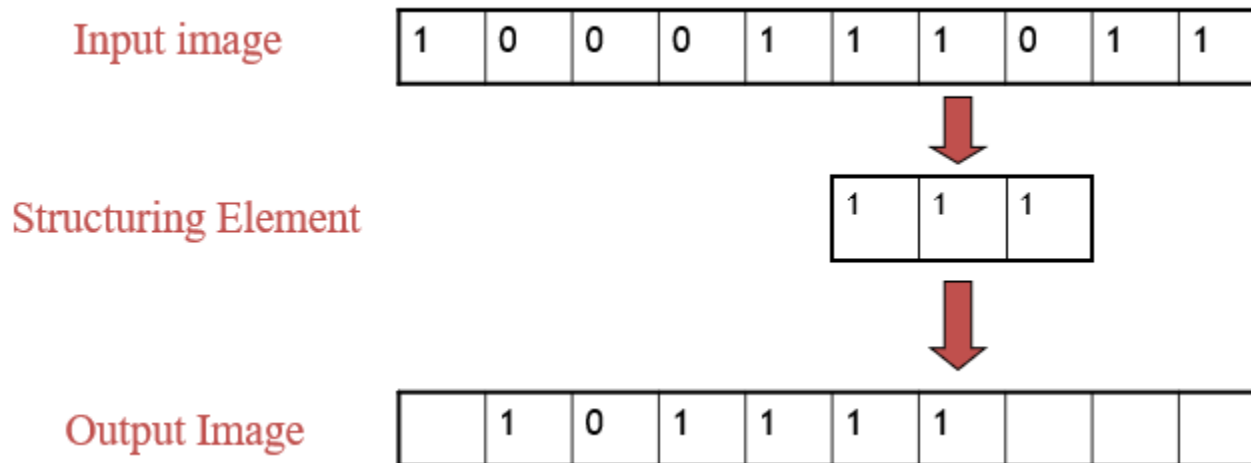
MORFOLOGIA MATEMÁTICA

- Dilatação.



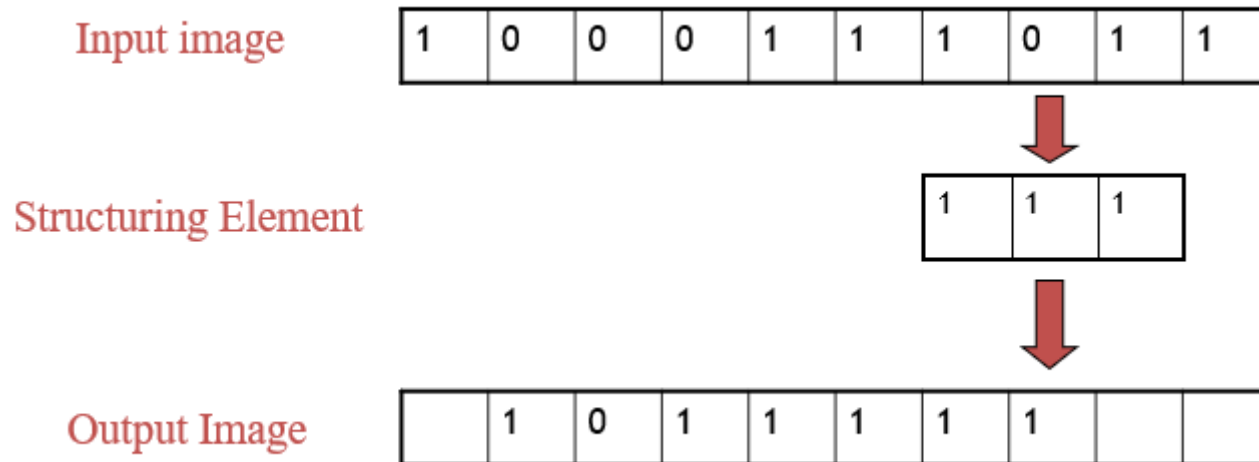
MORFOLOGIA MATEMÁTICA

- Dilatação.



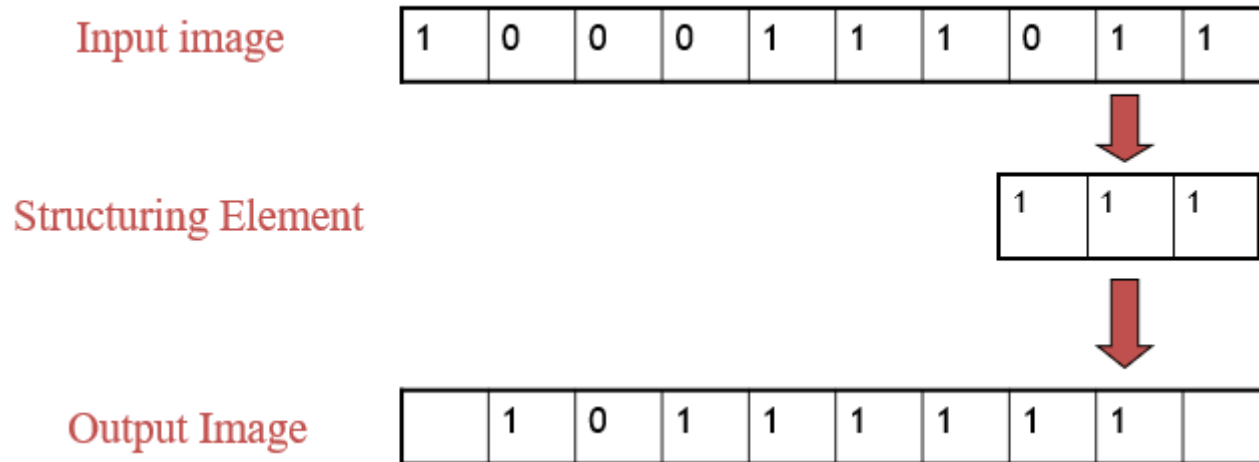
MORFOLOGIA MATEMÁTICA

- Dilatação.



MORFOLOGIA MATEMÁTICA

- Dilatação.



MORFOLOGIA MATEMÁTICA

■ Dilatação.

■ Código no OpenCV.

C++: `void dilate(InputArray src, OutputArray dst, InputArray kernel, Point anchor=Point(-1,-1), int iterations=1, int borderType=BORDER_CONSTANT, const Scalar& borderValue=morphologyDefaultBorderValue())`

Python: `cv2.dilate(src, kernel[, dst[, anchor[, iterations[, borderType[, borderValue]]]]) → dst`

Parameters:

- **src** - input image; the number of channels can be arbitrary, but the depth should be one of CV_8U, CV_16U, CV_16S, CV_32F or CV_64F.
- **dst** - output image of the same size and type as src.
- **kernel** - structuring element used for dilation; if `element=Mat()`, a 3 x 3 rectangular structuring element is used. Kernel can be created using `getStructuringElement()`
- **anchor** - position of the anchor within the element; default value (-1, -1) means that the anchor is at the element center.
- **iterations** - number of times dilation is applied.
- **borderType** - pixel extrapolation method (see `borderInterpolate` for details).
- **borderValue** - border value in case of a constant border

MORFOLOGIA MATEMÁTICA

- Dilatação.
 - Exemplo de código no OpenCV.

```
import cv2
import numpy as np

img = cv2.imread('j.png',0)
kernel = np.ones((5,5),np.uint8)
dilation = cv2.dilate(img,kernel,iterations = 1)
```



MORFOLOGIA MATEMÁTICA

- Abertura.
 - Define-se abertura de um conjunto A por B como a erosão de A por B seguido de uma dilatação por B.

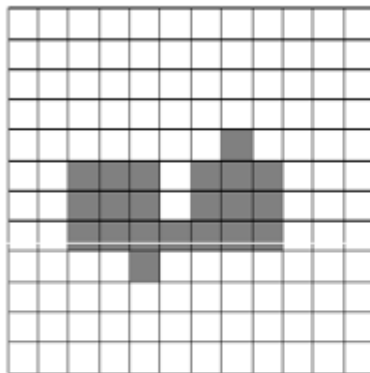
$$A \circ B = (A \ominus B) \oplus B$$

- A erosão realiza um encolhimento das regiões.
- A dilatação expande os componentes da imagem.
- Através da combinação destes dois operadores é possível definir procedimentos de filtragem.
- A abertura geralmente suaviza o contorno de um objeto e elimina estruturas finas.

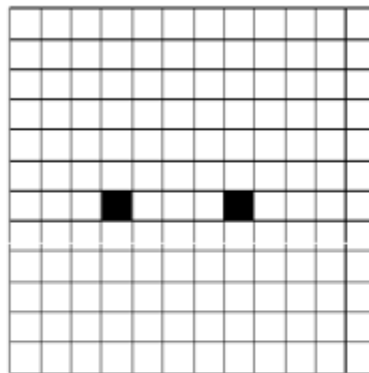
MORFOLOGIA MATEMÁTICA

- Abertura.

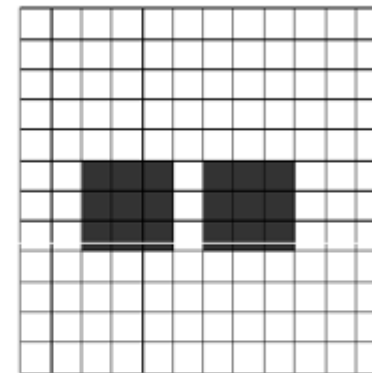
$$A \circ B = (A \ominus B) \oplus B$$



F



$F \ominus H$



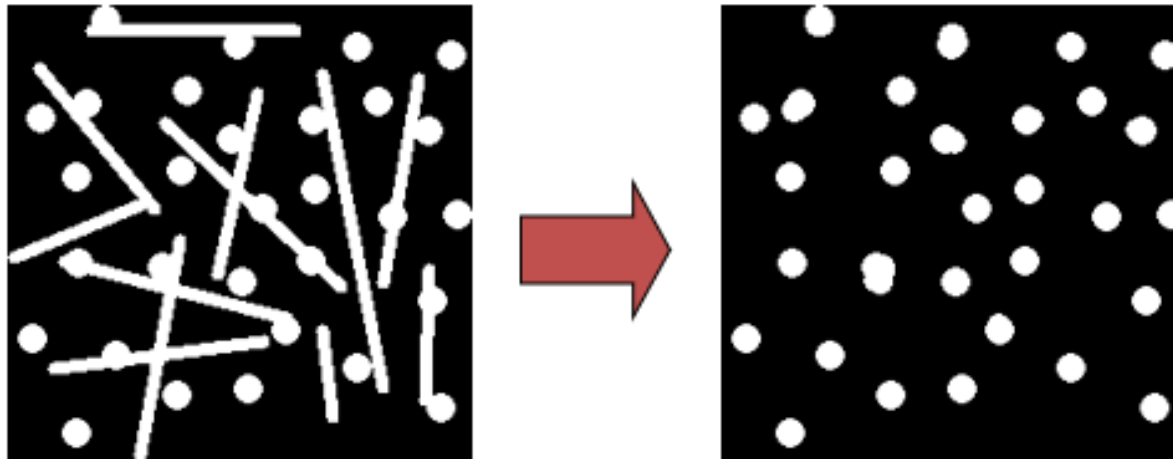
$(F \ominus H) \oplus H$



H, 3x3, origin at the center

MORFOLOGIA MATEMÁTICA

- Abertura.
- B = estrutura de disco 11 x 11.



MORFOLOGIA MATEMÁTICA

■ Abertura.

■ Código no OpenCV.

C++: `void morphologyEx(InputArray src, OutputArray dst, int op, InputArray kernel, Point anchor=Point(-1,-1), int iterations=1, int borderType=BORDER_CONSTANT, const Scalar& borderValue=morphologyDefaultBorderValue())`

Python: `cv2.morphologyEx(src, op, kernel[, dst[, anchor[, iterations[, borderType[, borderValue]]]])` → dst

- Parameters:**
- **src** – Source image. The number of channels can be arbitrary. The depth should be one of CV_8U, CV_16U, CV_16S, CV_32F or ``CV_64F.
 - **dst** – Destination image of the same size and type as **src**.
 - **kernel** – Structuring element. It can be created using `getStructuringElement()`.
 - **anchor** – Anchor position with the kernel. Negative values mean that the anchor is at the kernel center.
 - **op** –

Type of a morphological operation that can be one of the following:

- **MORPH_OPEN** – an opening operation
- **MORPH_CLOSE** – a closing operation
- **MORPH_GRADIENT** – a morphological gradient
- **MORPH_TOPHAT** – “top hat”
- **MORPH_BLACKHAT** – “black hat”

MORFOLOGIA MATEMÁTICA

- Abertura.
 - Exemplo de código no OpenCV.

```
import cv2
import numpy as np

img = cv2.imread('j.png',0)
kernel = np.ones((5,5),np.uint8)
opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
```



MORFOLOGIA MATEMÁTICA

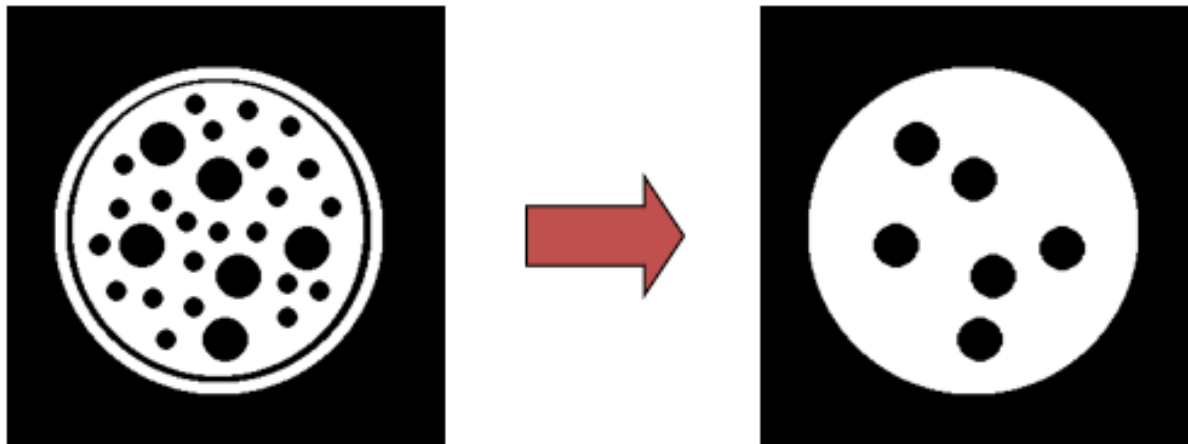
- Fechamento.
 - Define-se fechamento de um conjunto A por B como a dilatação de A por B seguida de uma erosão por B.

$$A \cdot B = (A \oplus B) \ominus B$$

- Aplicações:
 - Suaviza seções do contorno.
 - Preenche buracos.
 - Liga falhas no contorno.

MORFOLOGIA MATEMÁTICA

- Fechamento.
 - B = estrutura de disco 21 x 21.



MORFOLOGIA MATEMÁTICA

■ Fechamento.

■ Código no OpenCV.

C++: `void morphologyEx(InputArray src, OutputArray dst, int op, InputArray kernel, Point anchor=Point(-1,-1), int iterations=1, int borderType=BORDER_CONSTANT, const Scalar& borderValue=morphologyDefaultBorderValue())`

Python: `cv2.morphologyEx(src, op, kernel[, dst[, anchor[, iterations[, borderType[, borderValue]]]])` → dst

- Parameters:**
- **src** – Source image. The number of channels can be arbitrary. The depth should be one of CV_8U, CV_16U, CV_16S, CV_32F or ``CV_64F.
 - **dst** – Destination image of the same size and type as **src**.
 - **kernel** – Structuring element. It can be created using `getStructuringElement()`.
 - **anchor** – Anchor position with the kernel. Negative values mean that the anchor is at the kernel center.
 - **op** –

Type of a morphological operation that can be one of the following:

- **MORPH_OPEN** – an opening operation
- **MORPH_CLOSE** – a closing operation
- **MORPH_GRADIENT** – a morphological gradient
- **MORPH_TOPHAT** – “top hat”
- **MORPH_BLACKHAT** – “black hat”

MORFOLOGIA MATEMÁTICA

- Fechamento.
 - Exemplo de código no OpenCV.

```
import cv2
import numpy as np

img = cv2.imread('j.png',0)
kernel = np.ones((5,5),np.uint8)
closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
```



MORFOLOGIA MATEMÁTICA

- Propriedades da abertura e fechamento.
 - Abertura:
 - $A \circ B$ é um subconjunto (subimagem) de A .
 - Se C é um subconjunto de D , então $C \circ B$ é um subconjunto de $D \circ B$.
 - $(A \circ B) \circ B = A \circ B$.
 - Fechamento:
 - A é um subconjunto de $A \cdot B$.
 - Se C é um subconjunto de D , então $C \cdot B$ é subconjunto de $D \cdot B$.
 - $(A \cdot B) \cdot B = A \cdot B$.

MORFOLOGIA MATEMÁTICA

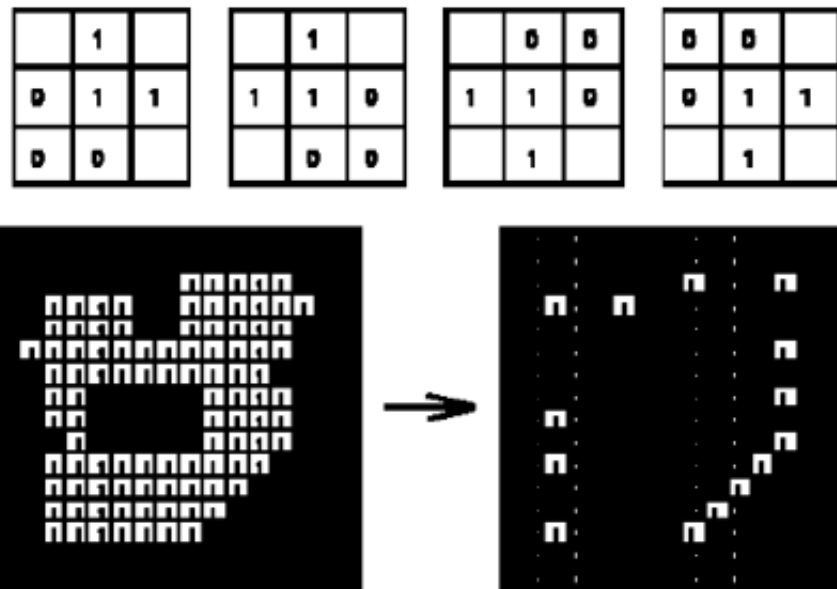
- Transformada acerto ou erro (hit or miss).
 - É uma ferramenta de detecção de formas baseada em morfologia matemática.
 - Definições:
 - B1 está associado com o objeto ($B1 = D$).
 - B2 está associado ao fundo ($B2 = W - D$).

$$A \otimes B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

- O hit or miss contém os pontos que satisfazem as duas condições simultâneas:
 - Os pontos de B1 que tocam A.
 - Os pontos de B2 que tocam A^c , ou seja, os pontos que não tocam A.

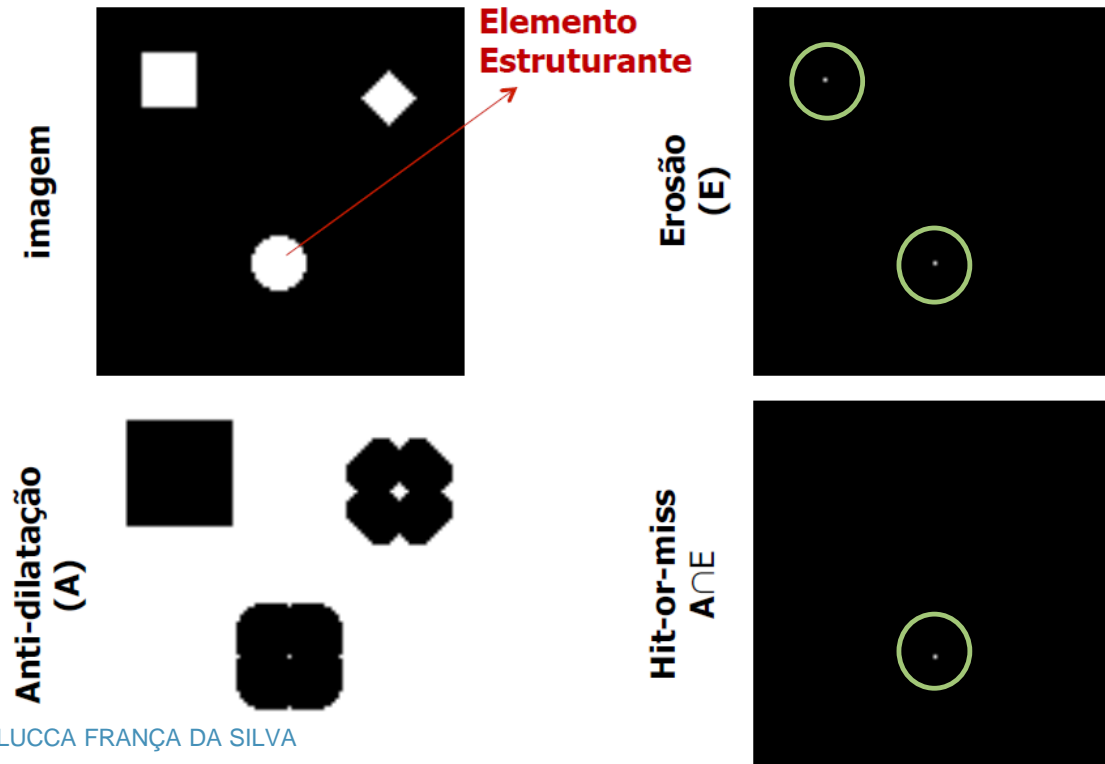
MORFOLOGIA MATEMÁTICA

- Transformada acerto ou erro (hit or miss).
 - Aplicação: detecção de cantos.



MORFOLOGIA MATEMÁTICA

- Transformada acerto ou erro (hit or miss).



MORFOLOGIA MATEMÁTICA

- Elementos estruturantes já definidos no OpenCV.

```
# Rectangular Kernel
>>> cv2.getStructuringElement(cv2.MORPH_RECT,(5,5))
array([[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]], dtype=uint8)

# Elliptical Kernel
>>> cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(5,5))
array([[0, 0, 1, 0, 0],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [0, 0, 1, 0, 0]], dtype=uint8)

# Cross-shaped Kernel
>>> cv2.getStructuringElement(cv2.MORPH_CROSS,(5,5))
array([[0, 0, 1, 0, 0],
       [0, 0, 1, 0, 0],
       [1, 1, 1, 1, 1],
       [0, 0, 1, 0, 0],
       [0, 0, 1, 0, 0]], dtype=uint8)
```

REFERÊNCIAS

- GONZALEZ, Rafael C.; WOODS, Richard C. **Processamento digital de imagens**. Pearson, 2011.
- PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações**. Thomson Learning, 2008.
- SILVA, Aristófanés. **Notas de aula da disciplina Processamento de Imagens da Universidade Federal do Maranhão**. 2018.
- BRAZ Jr, Geraldo. **Notas de aula da disciplina Visão Computacional da Universidade Federal do Maranhão**. 2018.