# Computer Vision

Numerical Image Understanding

# Plan

- Image analysis
    - Pixel notions
    - Image histogram
- Image filtering
    - Convolution filtering
    - Spatial frequency
- Mathematical morphology operations
    - Erosion / dilation
    - Opening / closing

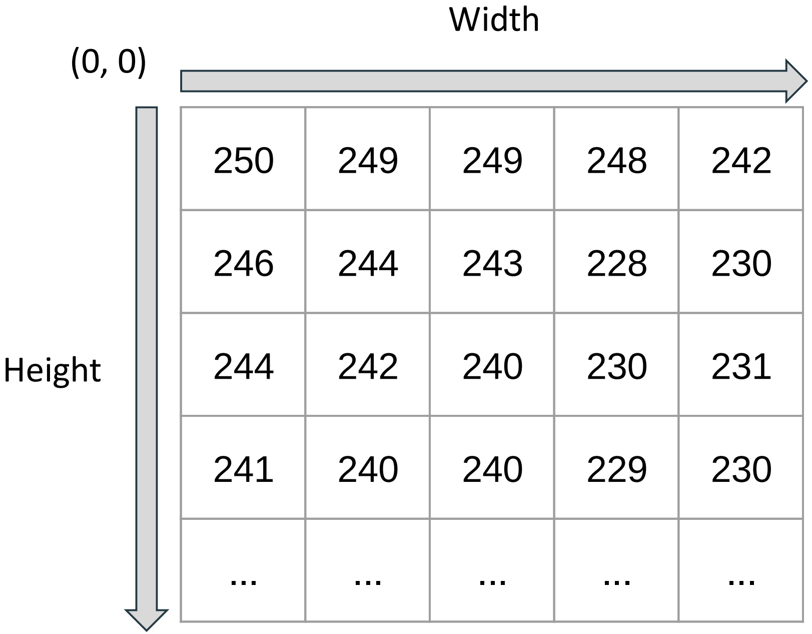
# Image Analysis

Pixel notions & histograms

# Pixel notions

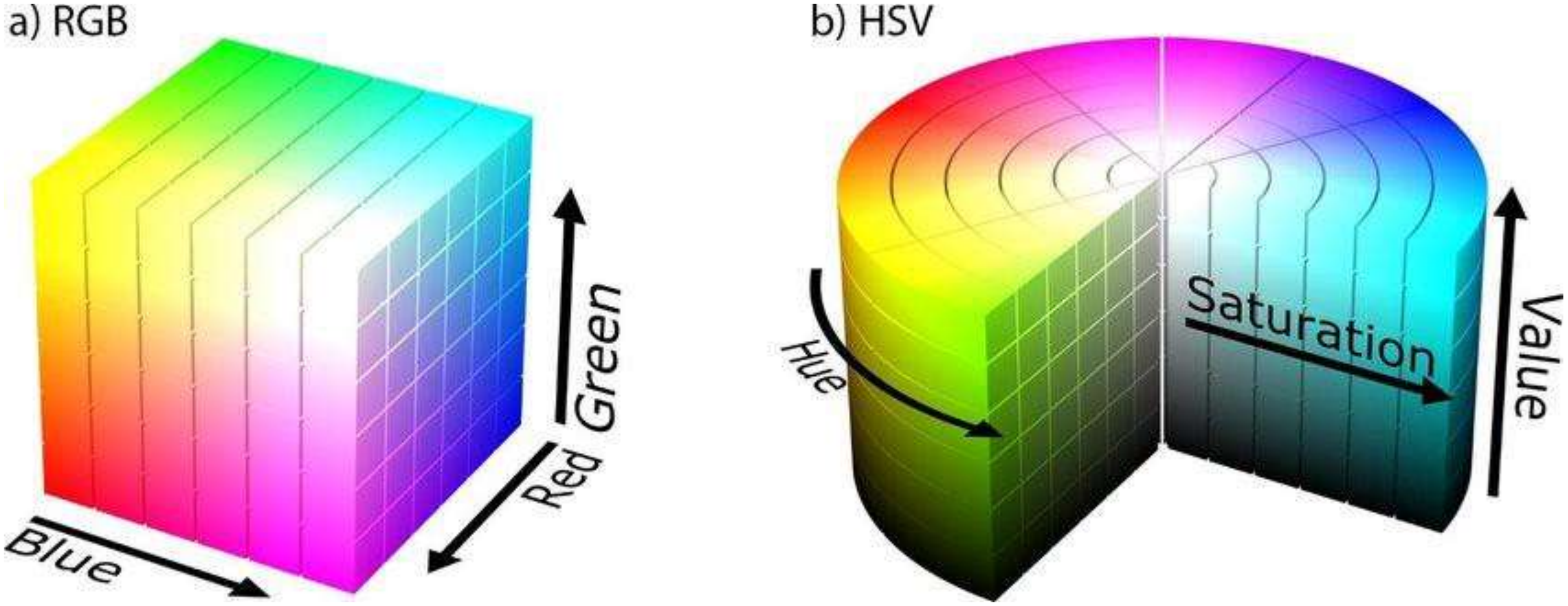
- Image sampling : **divide** image in **small areas** (pixels) containing a **value** (or a list of values)
- Coordinates : Line / **Width** & Column / **Height**
- Quantification : number of **possible values**
- Definition : W X H
- Resolution : Pixels / Length unit
- Compression

### Grayscale



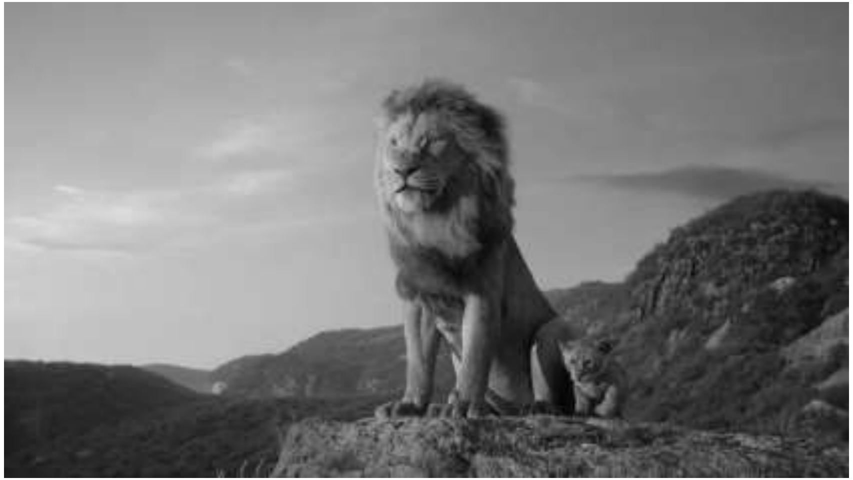
| | | | | |
|---|---|---|---|---|
| 250 | 249 | 249 | 248 | 242 |
| 246 | 244 | 243 | 228 | 230 |
| 244 | 242 | 240 | 230 | 231 |
| 241 | 240 | 240 | 229 | 230 |
| ... | ... | ... | ... | ... |

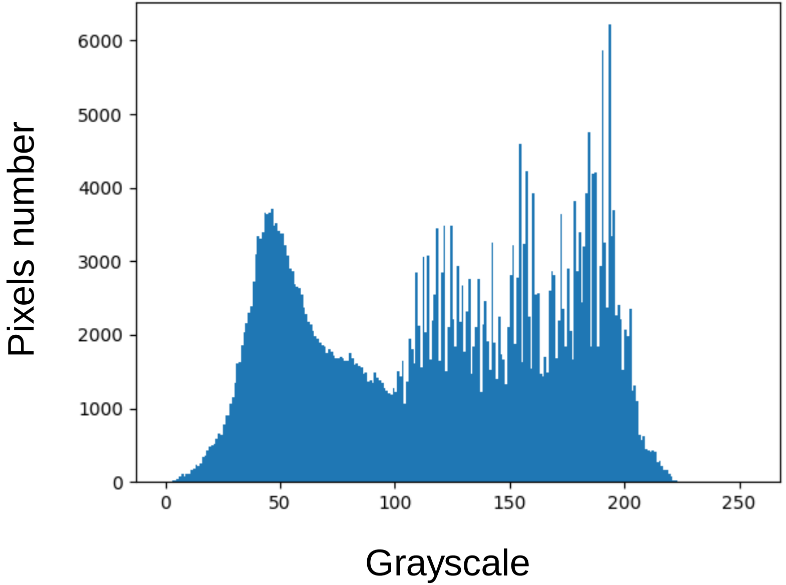
# Colors



# Image Histogram

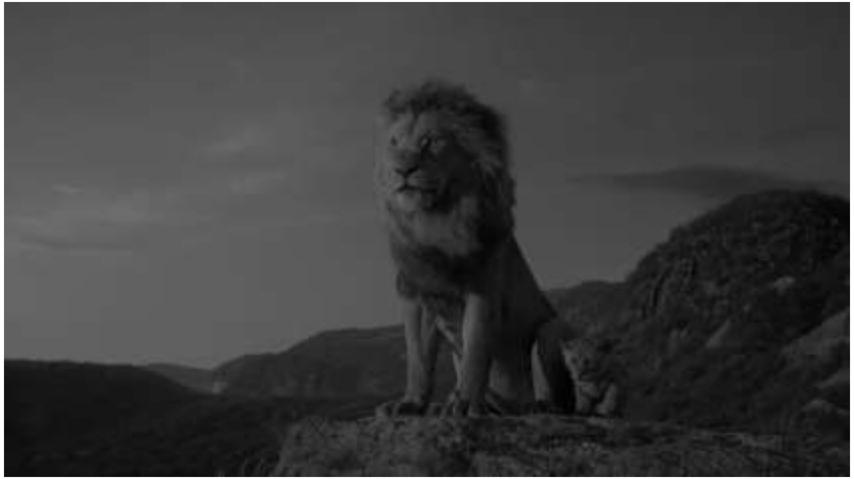
Original

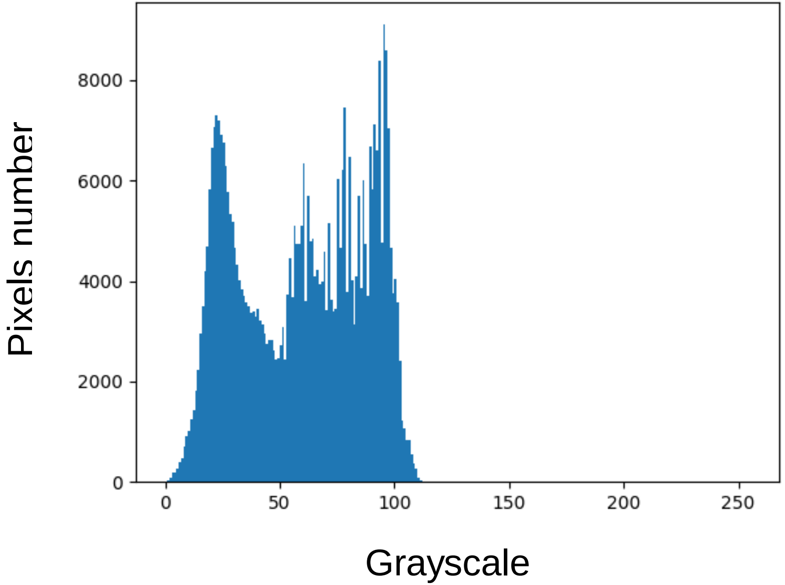




# Image Histogram

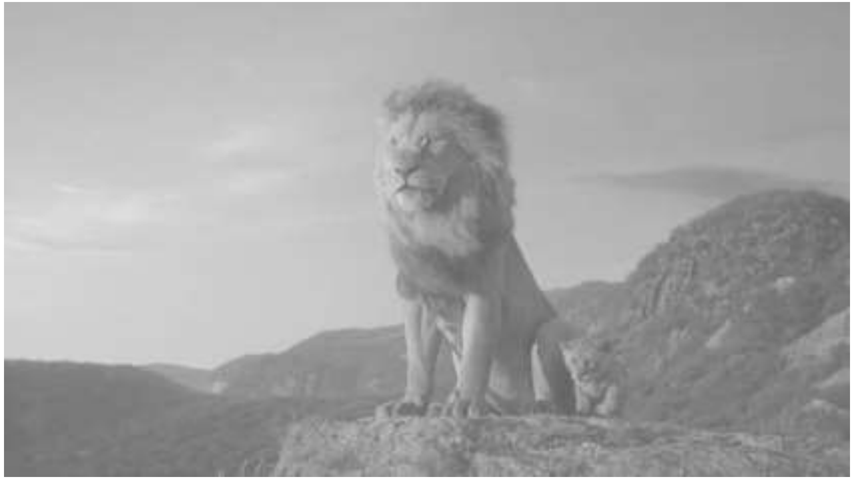
Under exposed

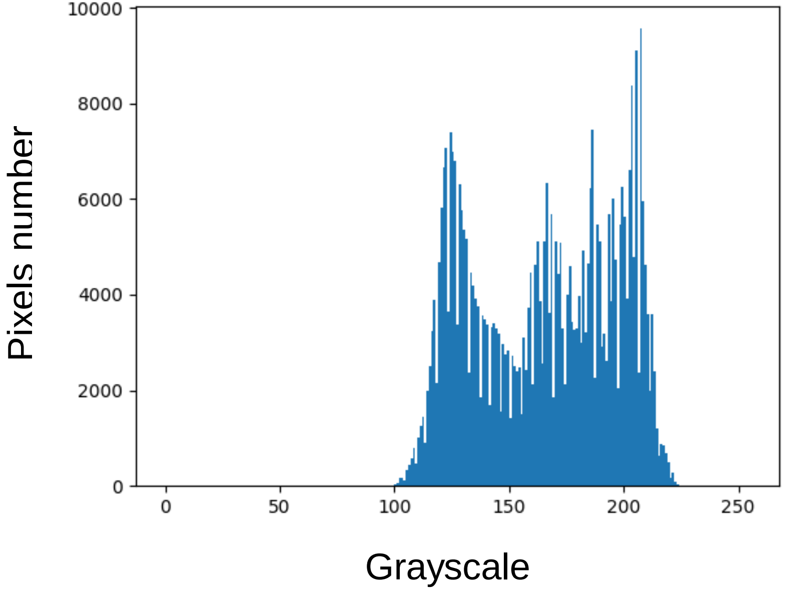




# Image Histogram

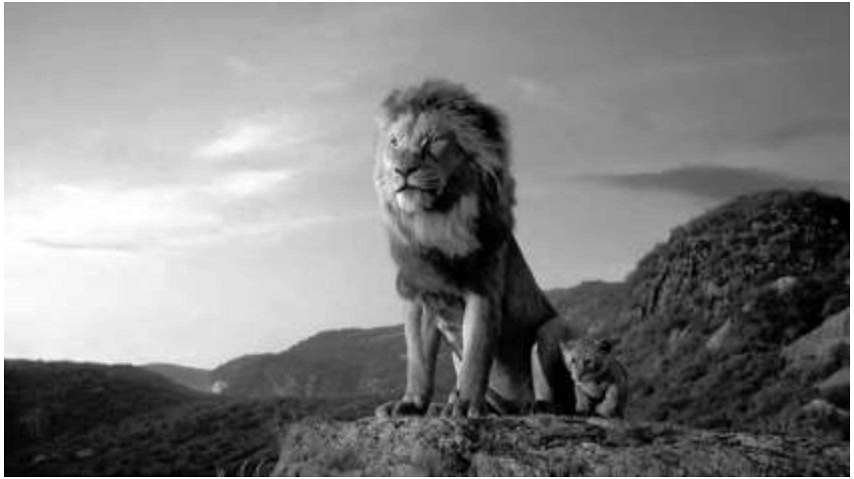
Over exposed

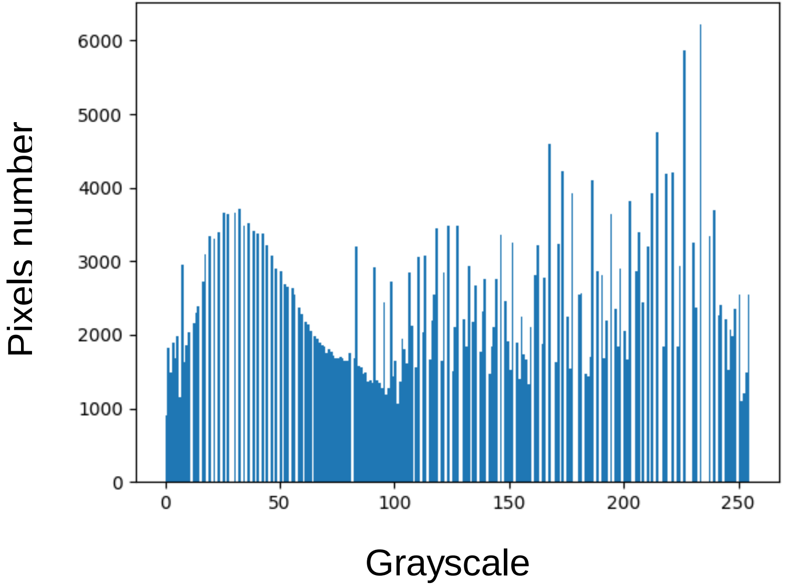




# Histogram Equalization

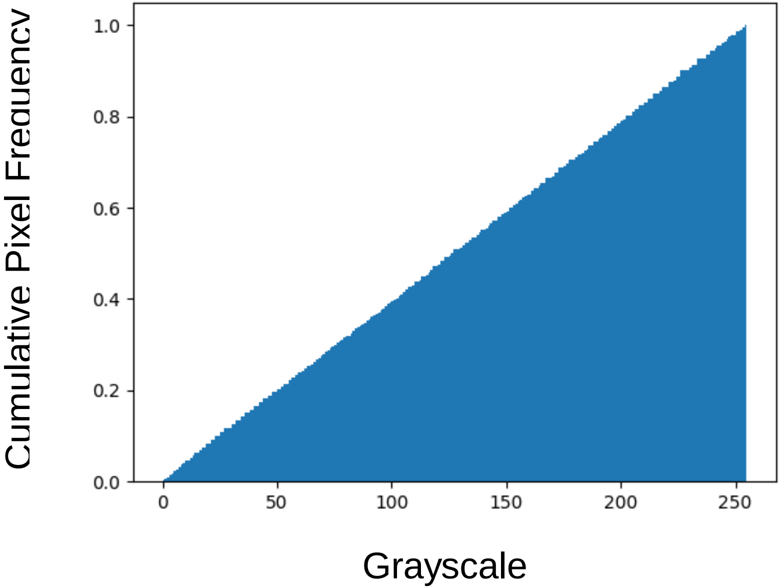
Contrast adjusted





# Histogram Equalization

Contrast adjusted



# Classical transformations

- Brightness: intensity measure (dark vs. fade / bright)
- Contrast : difference in brightness between dark and bright pixels
    - Histogram equalization

- Geometrical: translation / zoom / rotation
- Image compression

- Filtering : Noise removal / adding

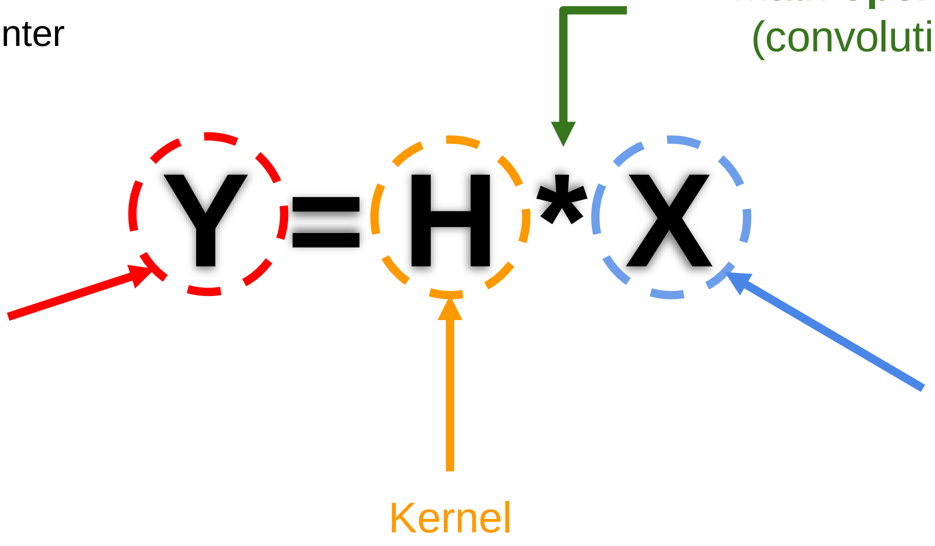
# Image Filtering

Convolution & Spatial Frequency

# Convolution (or linear) Filtering

Defined by:

- The support (shape of the kernel)
- The pixel in the center
- An algorithm



$$Y = H * X$$

Math **operator** (convolution)

Image array **after** filtering

Kernel

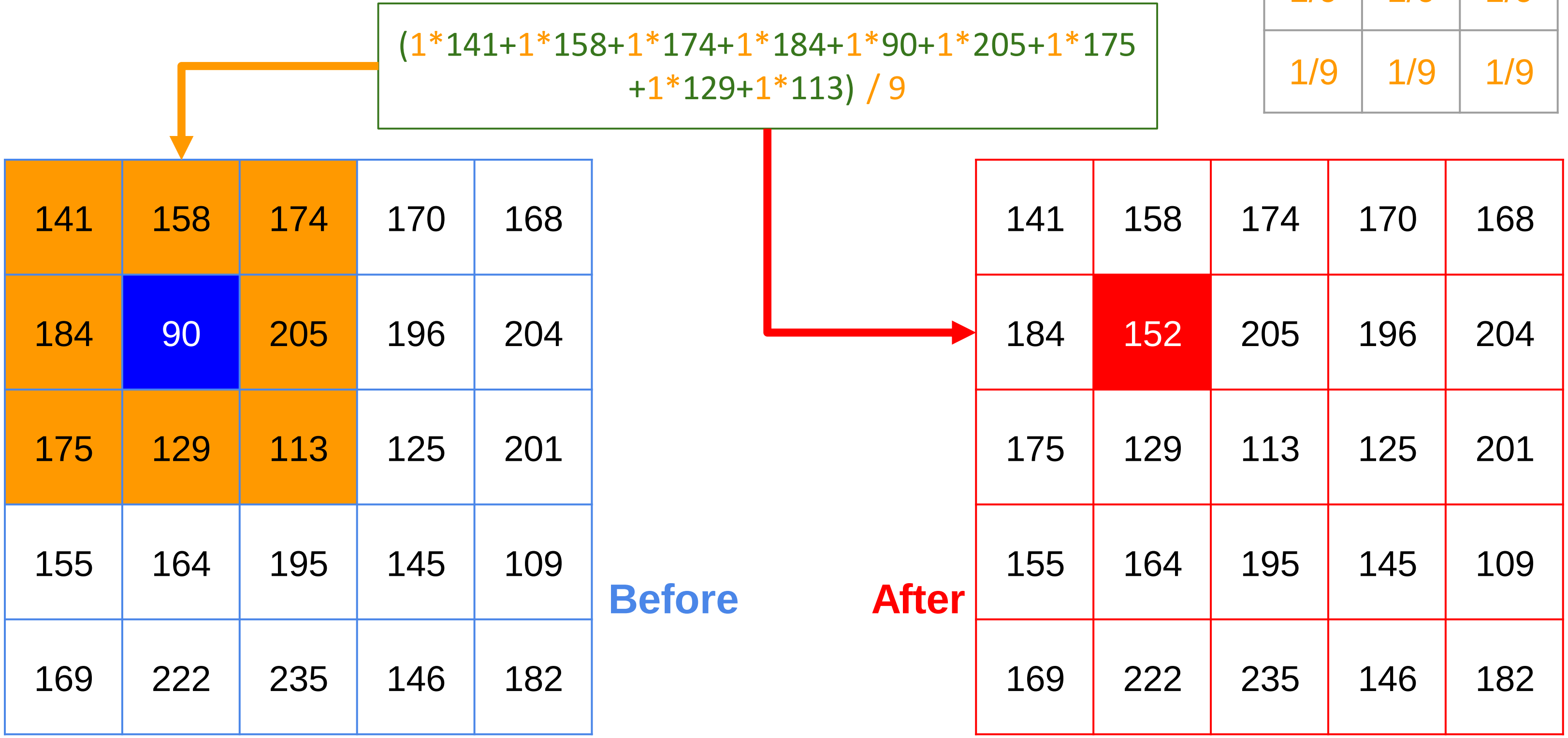
Image array **before** filtering

# Neighbors Averager Filter

Kernel

|  |  |  |
|---|---|---|
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

Convolution



(1*141+1*158+1*174+1*184+1*90+1*205+1*175 +1*129+1*113) / 9

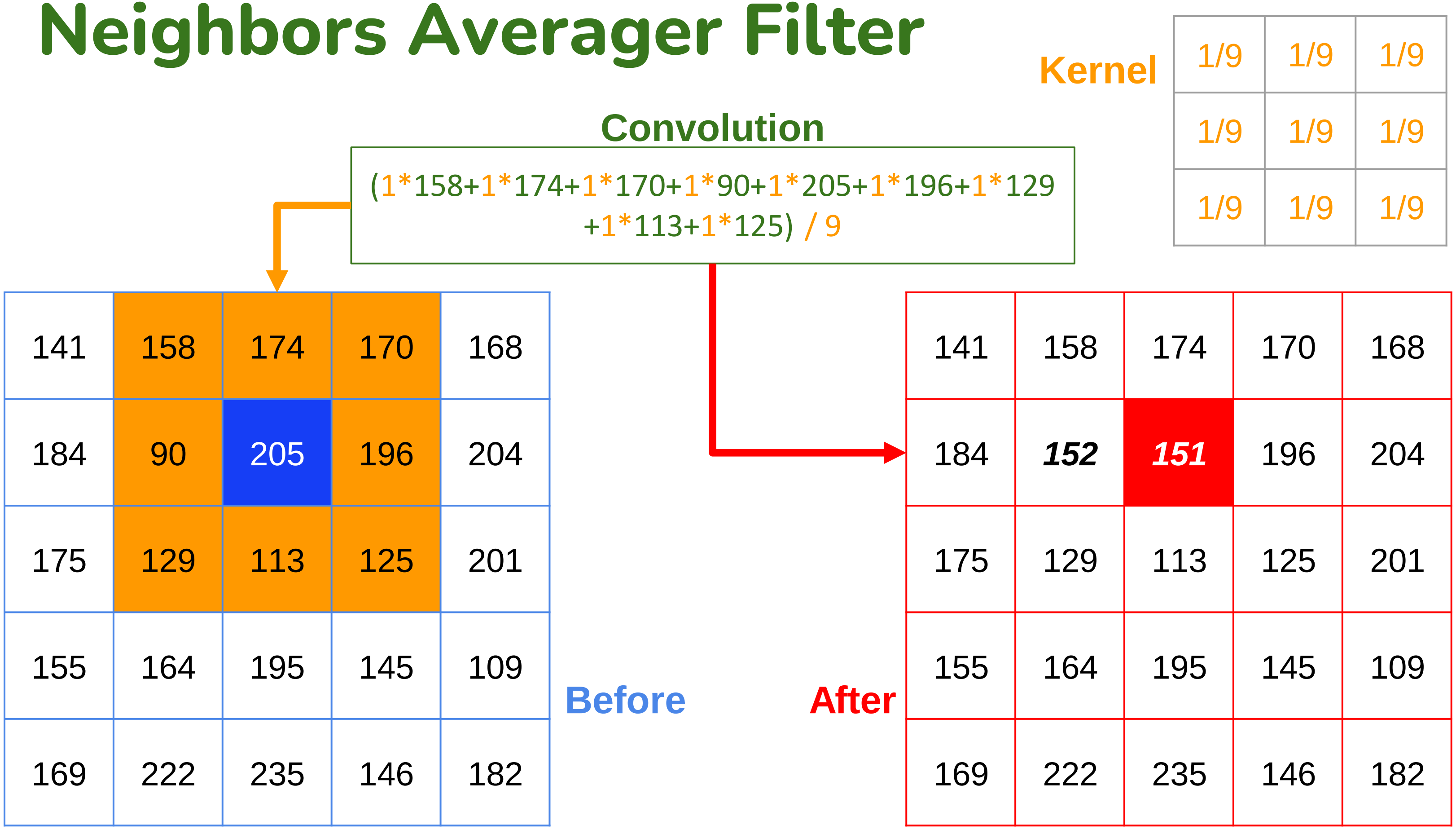
Before

|  |  |  |  |  |
|---|---|---|---|---|
| 141 | 158 | 174 | 170 | 168 |
| 184 | 90 | 205 | 196 | 204 |
| 175 | 129 | 113 | 125 | 201 |
| 155 | 164 | 195 | 145 | 109 |
| 169 | 222 | 235 | 146 | 182 |

After

|  |  |  |  |  |
|---|---|---|---|---|
| 141 | 158 | 174 | 170 | 168 |
| 184 | 152 | 205 | 196 | 204 |
| 175 | 129 | 113 | 125 | 201 |
| 155 | 164 | 195 | 145 | 109 |
| 169 | 222 | 235 | 146 | 182 |

# Neighbors Averager Filter



Kernel

| | | |
|---|---|---|
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

Convolution

(1*158+1*174+1*170+1*90+1*205+1*196+1*129 +1*113+1*125) / 9

| | | | | |
|---|---|---|---|---|
| 141 | 158 | 174 | 170 | 168 |
| 184 | 90 | 205 | 196 | 204 |
| 175 | 129 | 113 | 125 | 201 |
| 155 | 164 | 195 | 145 | 109 |
| 169 | 222 | 235 | 146 | 182 |

Before

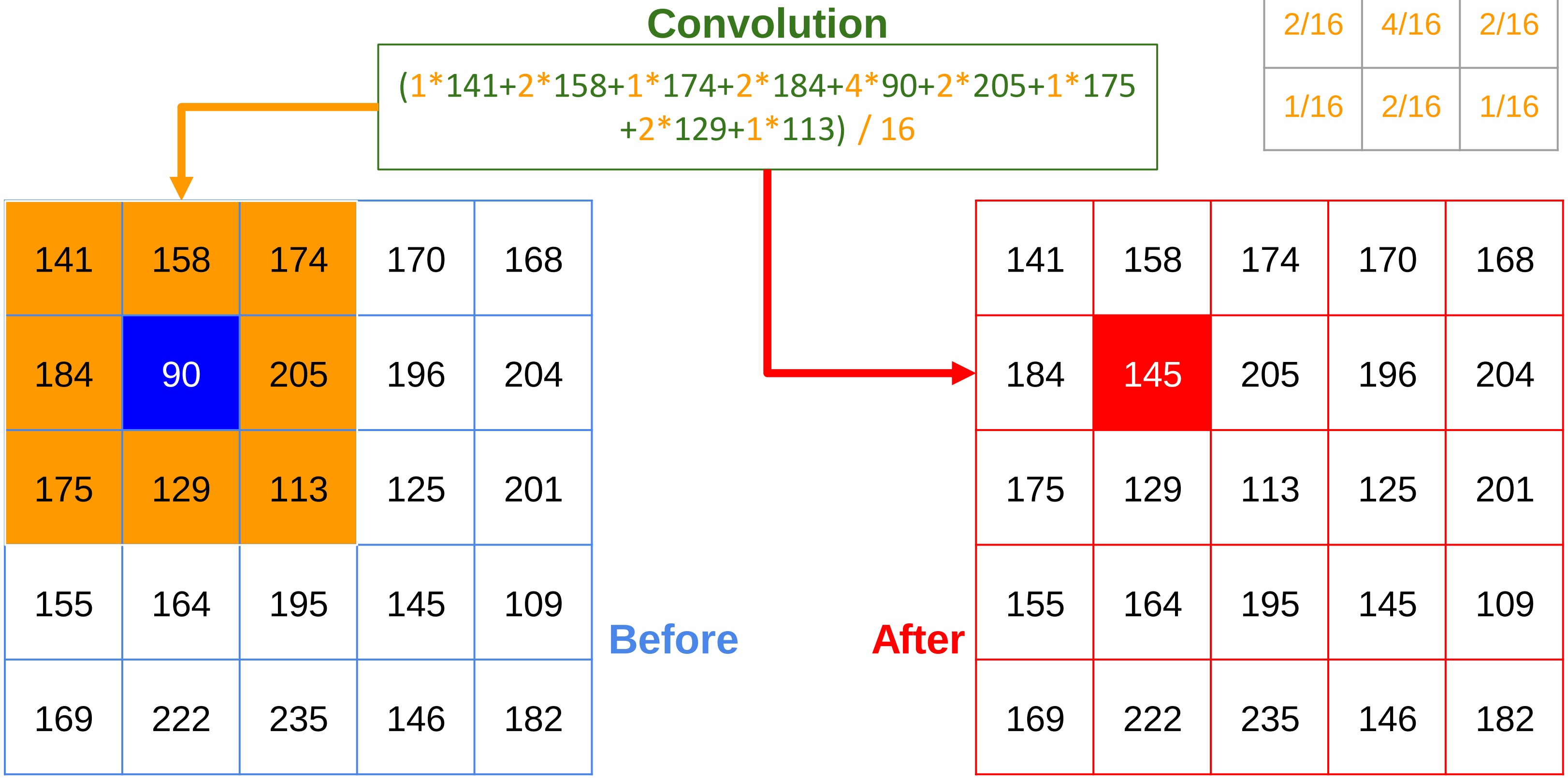
After

| | | | | |
|---|---|---|---|---|
| 141 | 158 | 174 | 170 | 168 |
| 184 | ***152*** | ***151*** | 196 | 204 |
| 175 | 129 | 113 | 125 | 201 |
| 155 | 164 | 195 | 145 | 109 |
| 169 | 222 | 235 | 146 | 182 |

# Gaussian Filter

Kernel

| | | |
|---|---|---|
| 1/16 | 2/16 | 1/16 |
| 2/16 | 4/16 | 2/16 |
| 1/16 | 2/16 | 1/16 |



Convolution

(1*141+2*158+1*174+2*184+4*90+2*205+1*175+2*129+1*113) / 16

Before

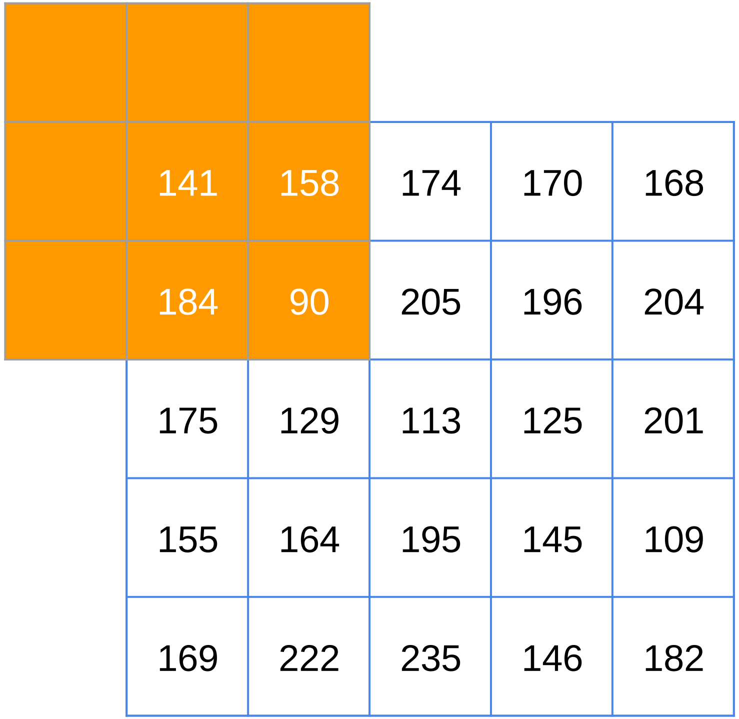
| | | | | |
|---|---|---|---|---|
| 141 | 158 | 174 | 170 | 168 |
| 184 | 90 | 205 | 196 | 204 |
| 175 | 129 | 113 | 125 | 201 |
| 155 | 164 | 195 | 145 | 109 |
| 169 | 222 | 235 | 146 | 182 |

After

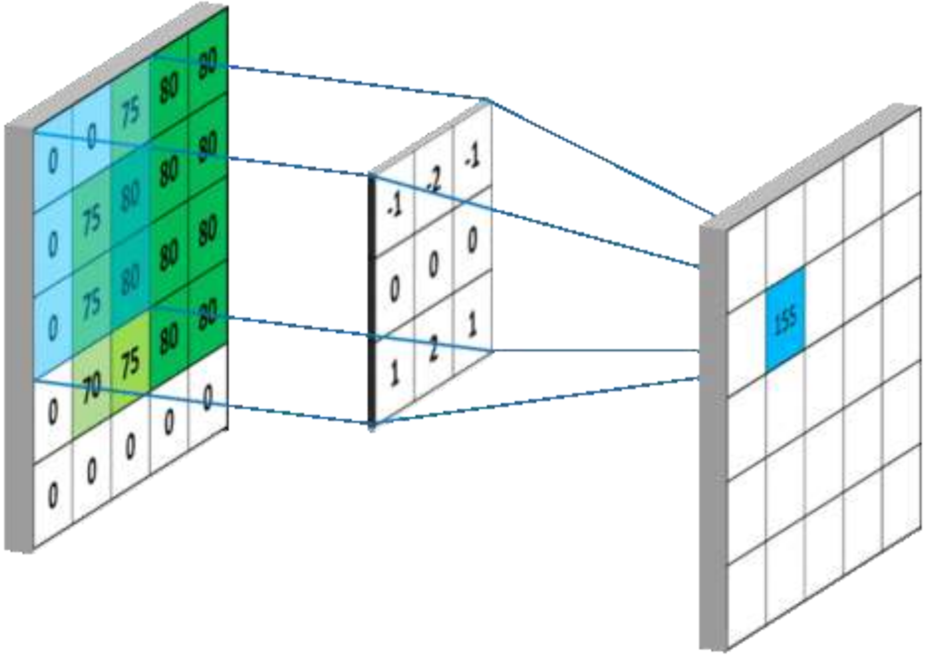
| | | | | |
|---|---|---|---|---|
| 141 | 158 | 174 | 170 | 168 |
| 184 | 145 | 205 | 196 | 204 |
| 175 | 129 | 113 | 125 | 201 |
| 155 | 164 | 195 | 145 | 109 |
| 169 | 222 | 235 | 146 | 182 |

# Borders management

- Zero padding
- Duplication
- Partial convolution



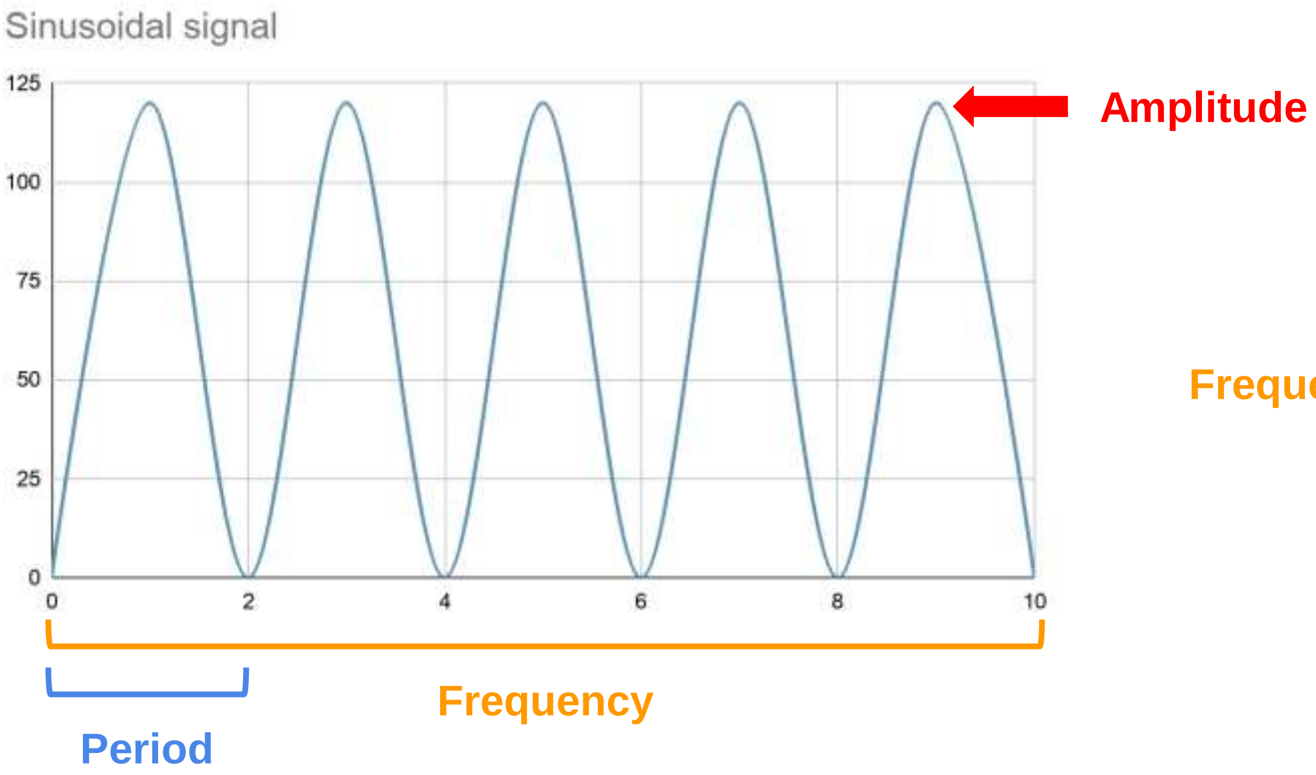
| | | | | | |
|---|---|---|---|---|---|
| | | | | | |
| | 141 | 158 | 174 | 170 | 168 |
| | 184 | 90 | 205 | 196 | 204 |
| | 175 | 129 | 113 | 125 | 201 |
| | 155 | 164 | 195 | 145 | 109 |
| | 169 | 222 | 235 | 146 | 182 |


0 0 75 80 80
0 75 80 80 80
0 75 80 80 80
0 70 75 80 80
0 0 0 0 0
-1 -2 -1
0 0 0
1 2 1
155

# Filters

- **Averager** (linear) : spread the noise to the neighbors
- **Gaussian** (linear): preserves better the surrounding of the corrected noise
- **Median** (non-linear): great to correct isolated pixels
- **Bilateral** :
  - An operation on neighbors
  - Classical convolution kernel

# Spatial Frequency

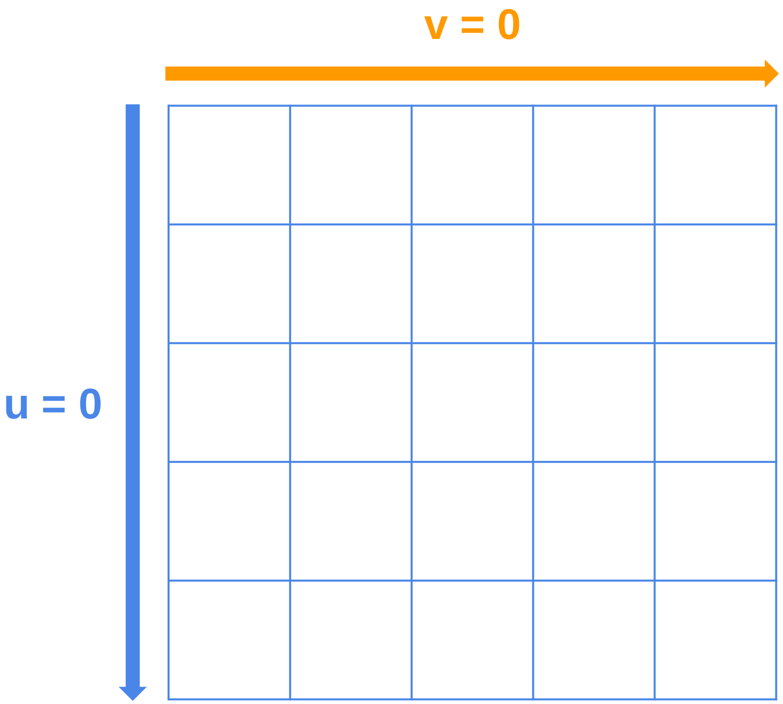


Period = 2

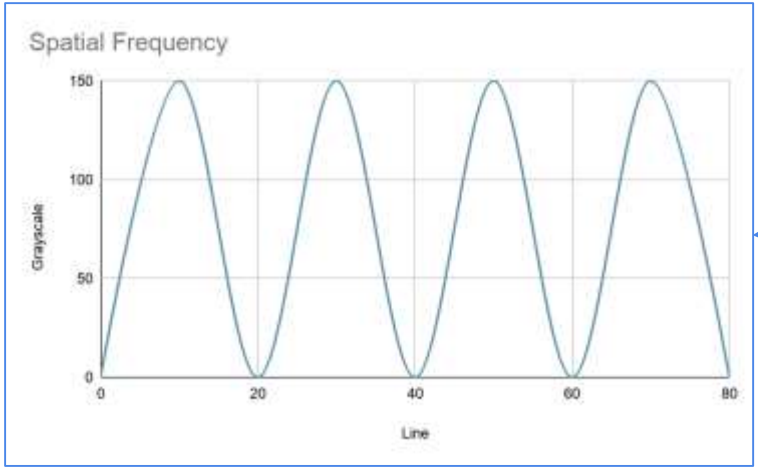
Frequency = 5/10 = 1/2

# Spatial Frequency

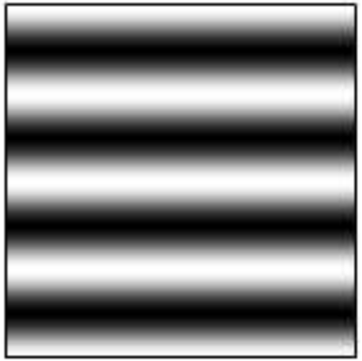
- 2 dimensions = **2 spatial frequencies**:
  - 1 following the u axis
  - 1 following the v axis



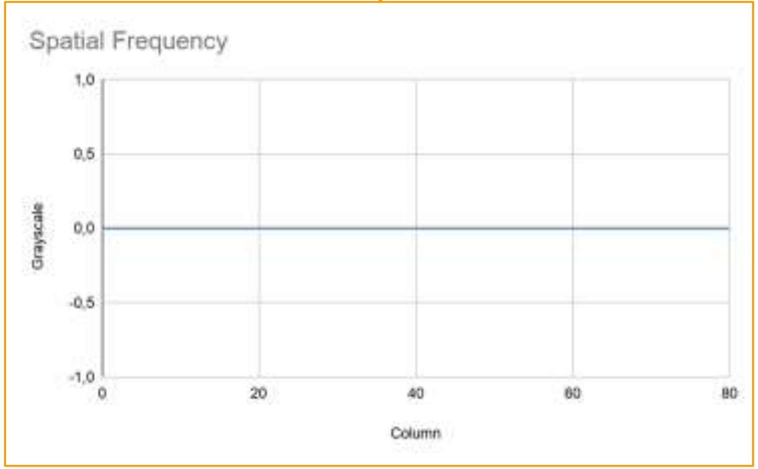
# Spatial Frequency



u = 4



V = 0



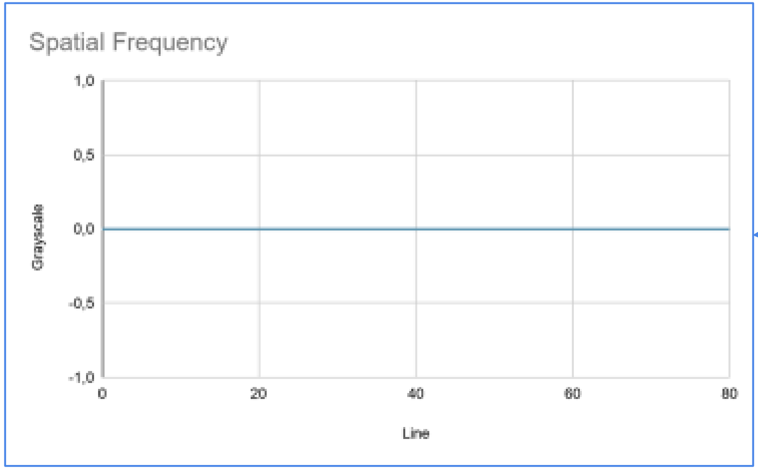
**Period(u) = Line/u = 20**

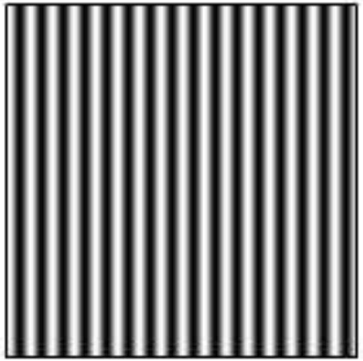
**Frequency(u) = u/Line = 0.2**

**Period(v) = Column/v = 0**

**Frequency(v) = v/Column = 0**

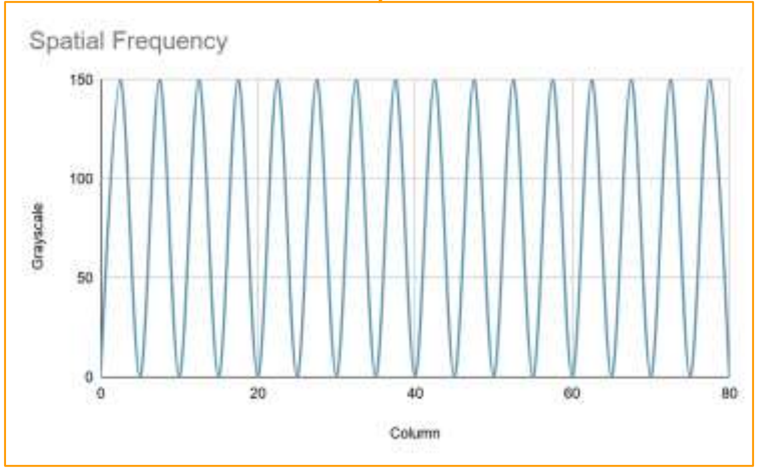
# Spatial Frequency





u = 0

V = 16



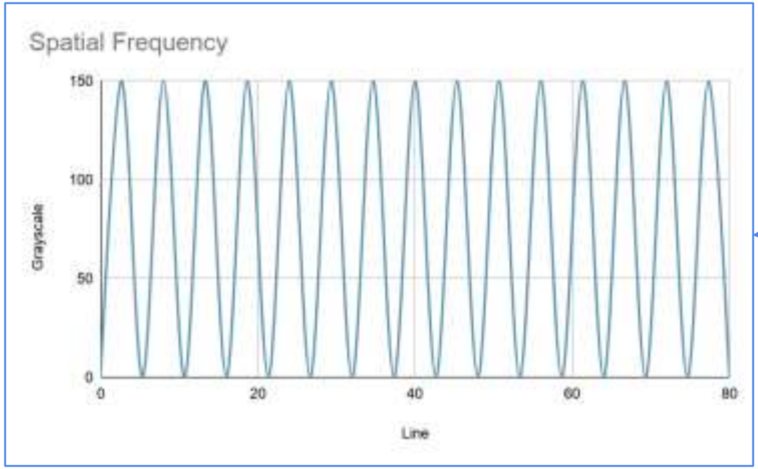
**Period(u) = Line/u = 0**

**Frequency(u) = u/Line = 0**

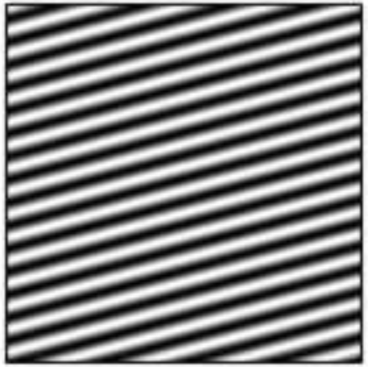
**Period(v) = Column/v = 5**

**Frequency(v) = v/Column = 0.2**

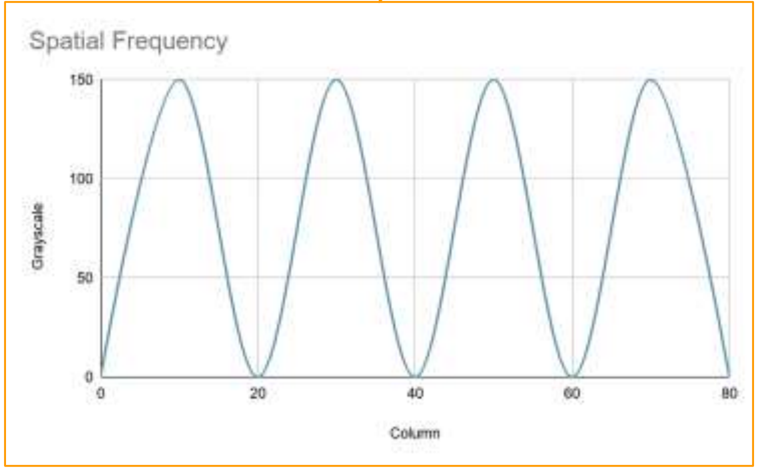
# Spatial Frequency



u = 15



V = 4



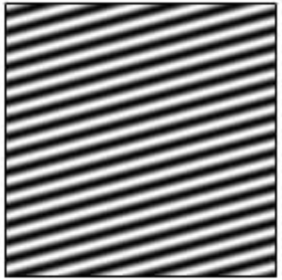
Period(u) = Line/u = 5.33

Frequency(u) = u/Line = 0.19

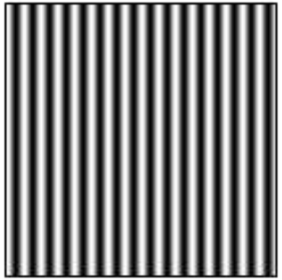
Period(v) = Column/v = 20

Frequency(v) = v/Column = 0.2

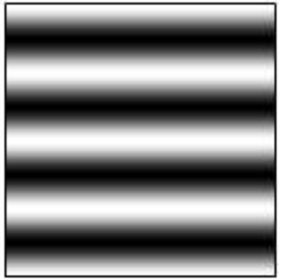
# Spatial Frequency



+



+



+

... (weighted sum of Line X Column images)

=

# Spatial Frequency Filtering

- **Low-pass** filter : keeps only low frequencies. The tinier the filter, the blurrier the image.
- **High-pass** filter: Keeps high frequencies. Useful to extract edges and remove content.
- **Enhancing** filter: keeps low frequencies and amplify high ones.

# Mathematical Morphology Operations

Erosion, dilation, opening & closing

# Mathematical Morphologie: Principle

- Using a **convolution kernel** of various shape.
- Change **the shape** of an item **inside** an image**.**
- **Remove** some **noise without** main image content **alteration**.

# Mathematical Morphologie: Erosion

## Convolution Kernel

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Central pixel** remains the same if **the whole kernel is included**. Else, the pixel changes to **the min** of the kernel area.

# Mathematical Morphologie: Erosion

**Convolution Kernel**

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

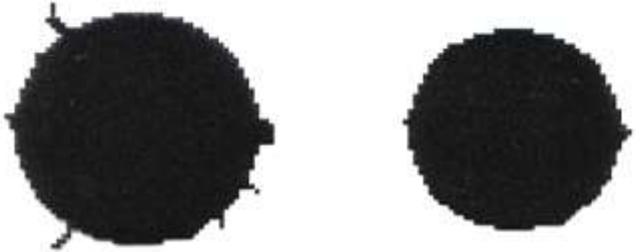
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Central pixel** remains the same if **the whole kernel is included**. Else, the pixel changes to **the min** of the kernel area.

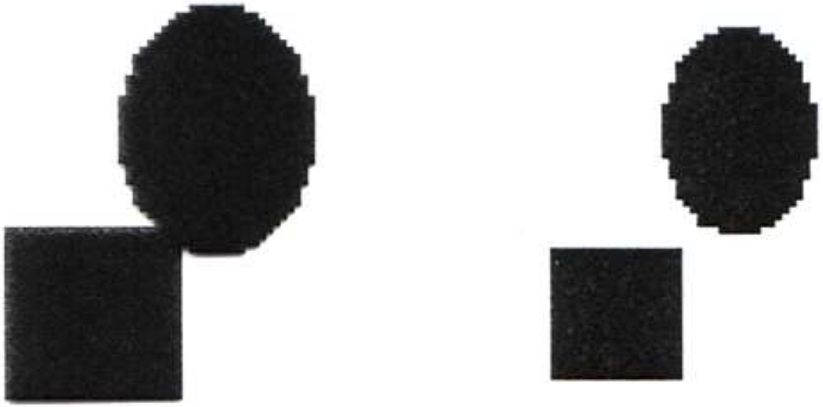
# Mathematical Morphologie: Erosion

Useful ? Why ?

Exemple 1



Exemple 2



# Mathematical Morphologie: Dilation

Convolution Kernel

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**For each pixel** that has a value of **1**, **apply** the **kernel**.

# Mathematical Morphologie: Dilation

**Convolution Kernel**

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

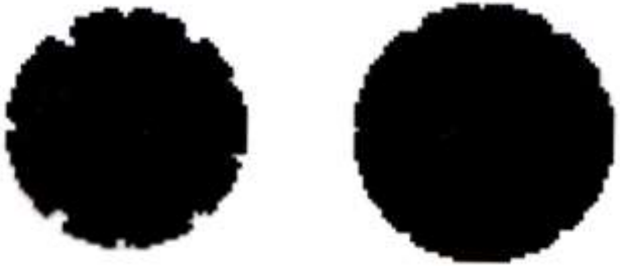
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**For each pixel** that has a value of **1**, **apply** the **kernel**.

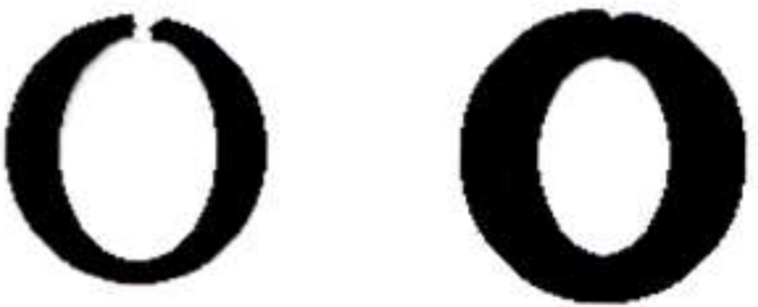
# Mathematical Morphologie: Dilation

Useful ? Why ?

Exemple 1



Exemple 2

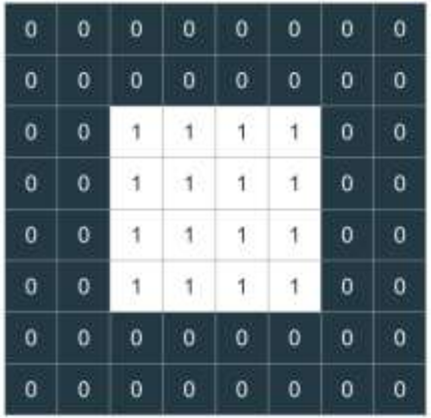


# Mathematical Morphologie: Opening

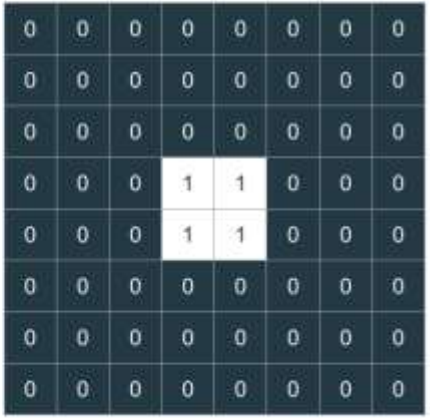
Opening = Erosion + Dilation

**Convolution Kernel**

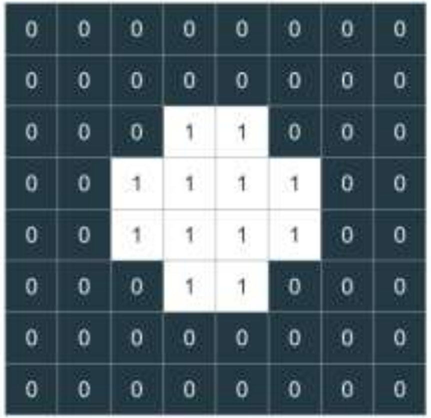
| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

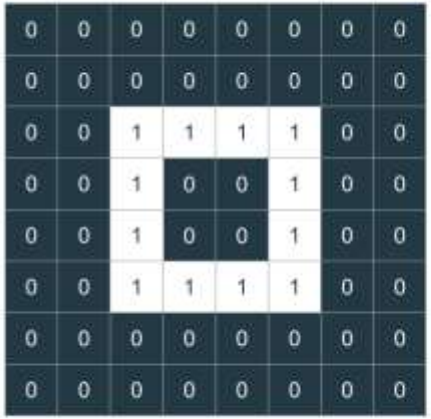
Useful to remove small size **elements** while keeping the global item shape.

# Mathematical Morphologie: Closing

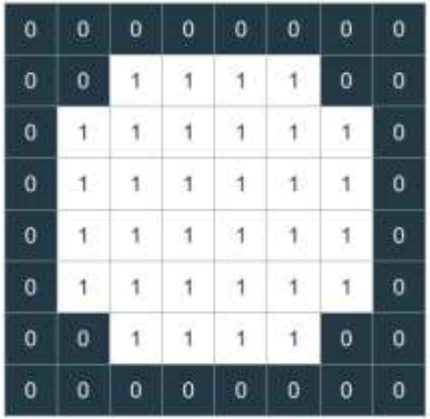
Closing = Dilation + Erosion

Convolution Kernel

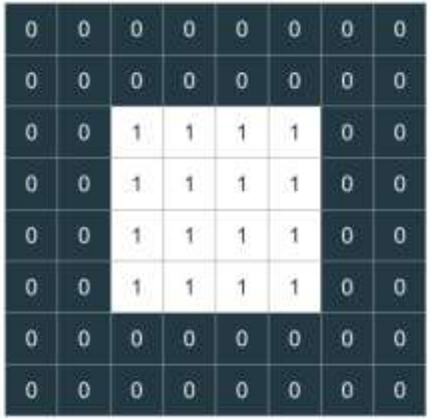
| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



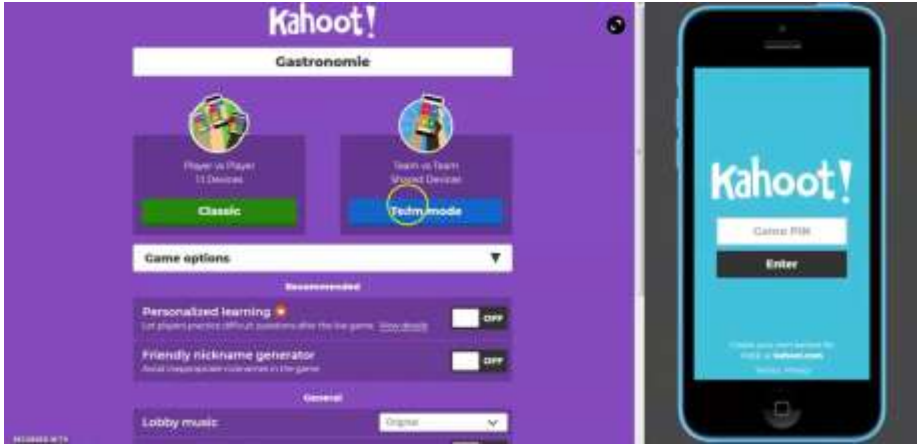
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Useful to remove small size **holes** while keeping the global item shape.

# Exercices

Coding games & images filtering

# Quizz Kahoot !



Personalized learning — OFF

Let players practice difficult questions after the live game. Show details

Friendly nickname generator — OFF

Avoid inappropriate nicknames in the game

Lobby music — Original

- ➢ Link: https://kahoot.it/
- ➢ Pin: xxxx

# Let’s play some Codingame !

STEP UP YOUR CODING GAME
The new way to improve your programming skills while having fun and getting noticed

- easy:
    - flip the sign
    - Reverse minesweeper
    - sudoku validator
    - lumen
    - pirate’s treasure
- medium:
    - forest fire
    - battleship

# Filtering: take your favorite image and...

- Detect objects inside of them
- Exemple: Coke can detector