INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

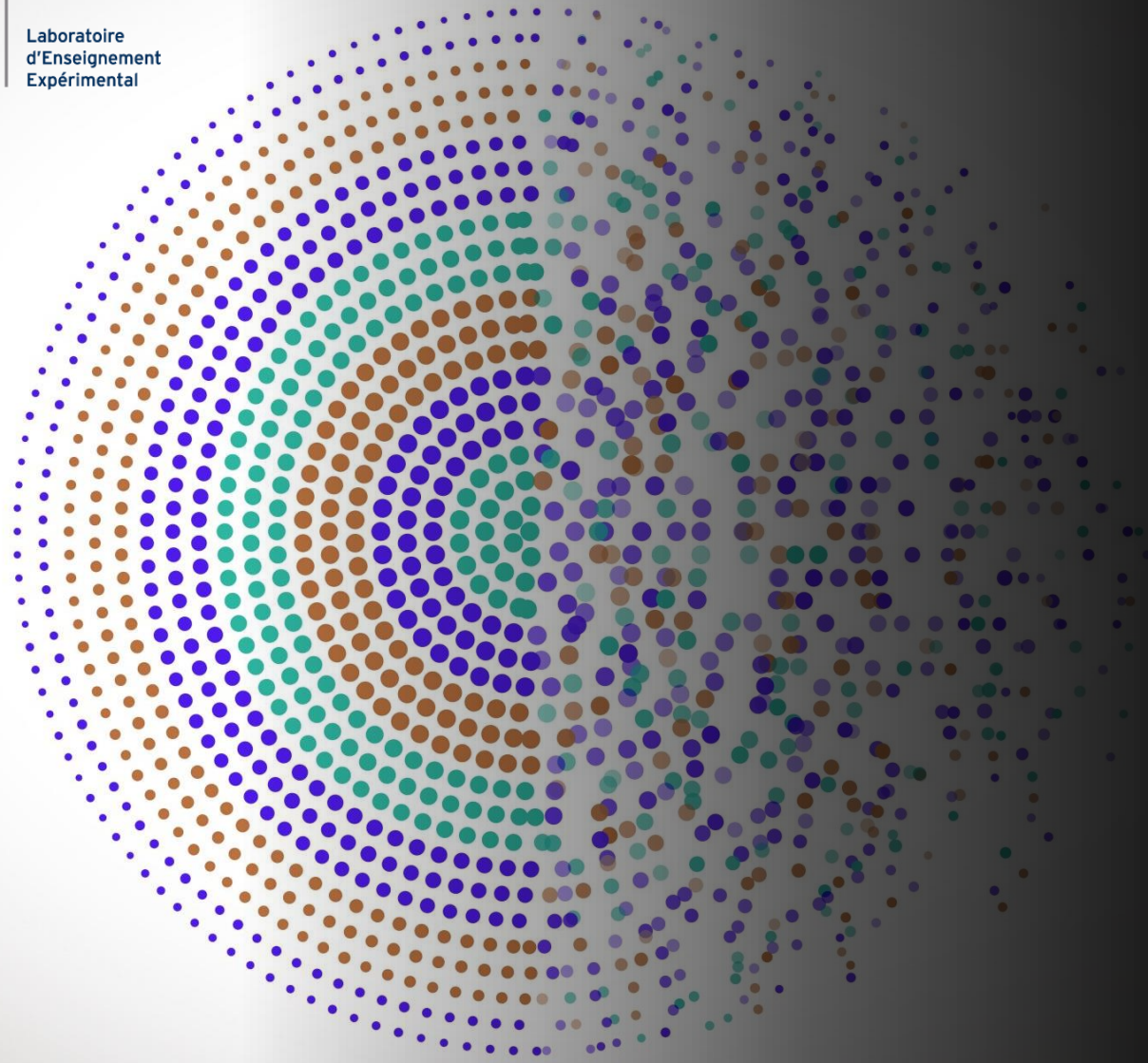
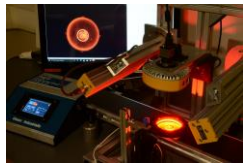


Image processing

Institut d'Optique – Engineers Training

Semester 6 – Digital Interface

Julien VILLEMEJANE



Goal of processing an image

Image from the camera

- **Noise**
- Bad contrast
- Inhomogeneous Lighting
- …

Desired image with objects with **well-defined contours**

- Homogeneous zones
- Transition zones

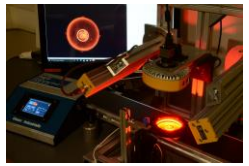


Image processing

Steps for processing an image

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

Acquisition

Light, Camera…

Pre Processing

Noise Reduction / Filtering
Contrast Enhancement
Normalization

Ensuring clarity and reducing unwanted information
Making features of interest stand out
Standardizing the image scale or intensity

Segmentation

Thresholding
Edge Detection
Region of Interest Selection

Isolating objects of ROI / Separating product from background
Identifying boundaries and contours
Focusing only on relevant portions of the image

Feature Extraction

Geometric Features
Texture Analysis
Color Analysis

Extracting measurements (size, shape, position…)
Recognizing patterns, symbols, points of interest

Classification

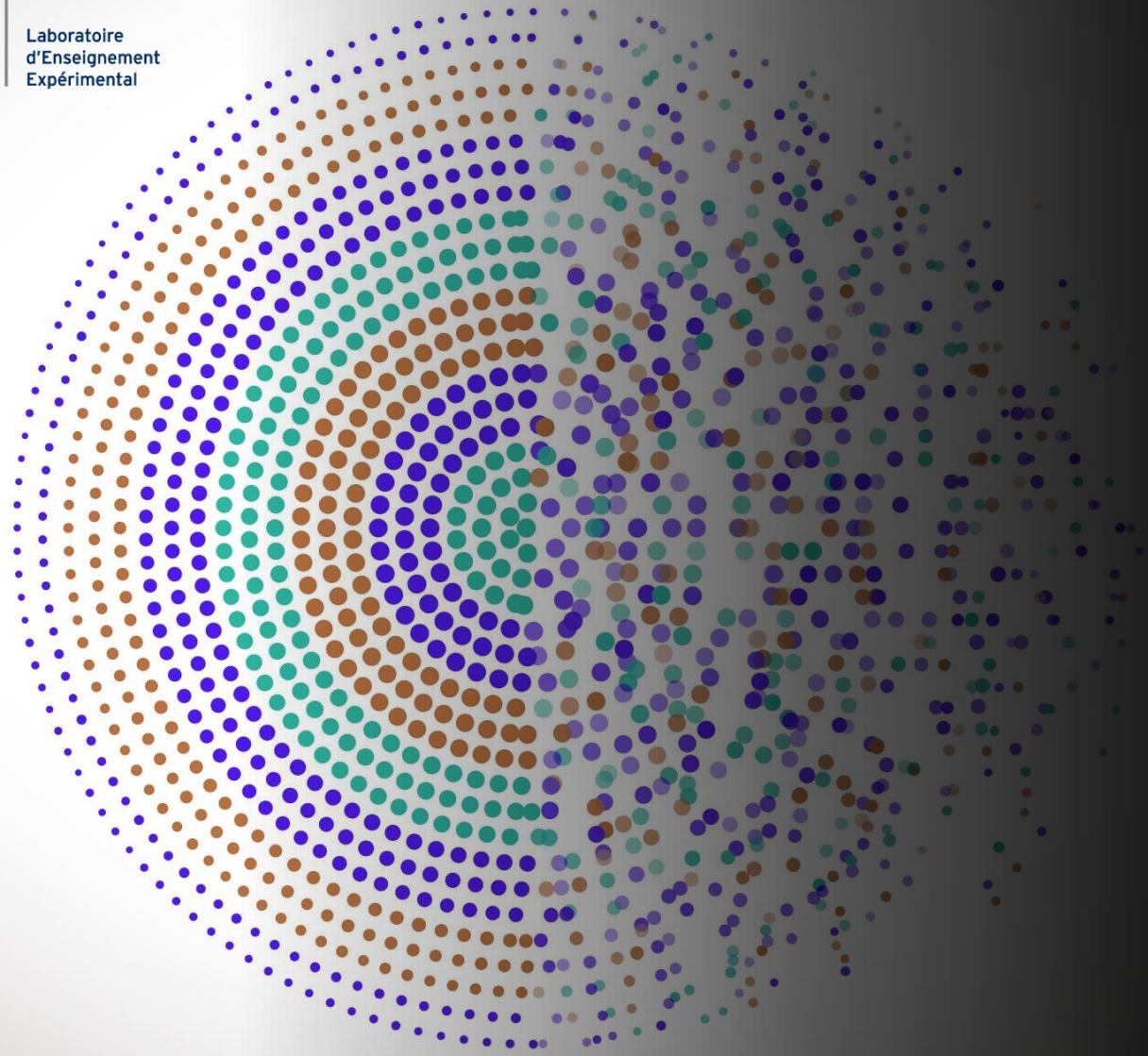
Object detection
Template Matching
Classification

Identifying and labelling objects
Checking correct shape, size or orientation
Categorizing objects to specific groups (defective or non-def.)

Decision

Tolerances, pass/fail, real-time feedback…

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

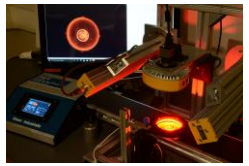


Digital Images and Processing

Institut d'Optique – Engineers Training

Semester 6 – Digital Interface

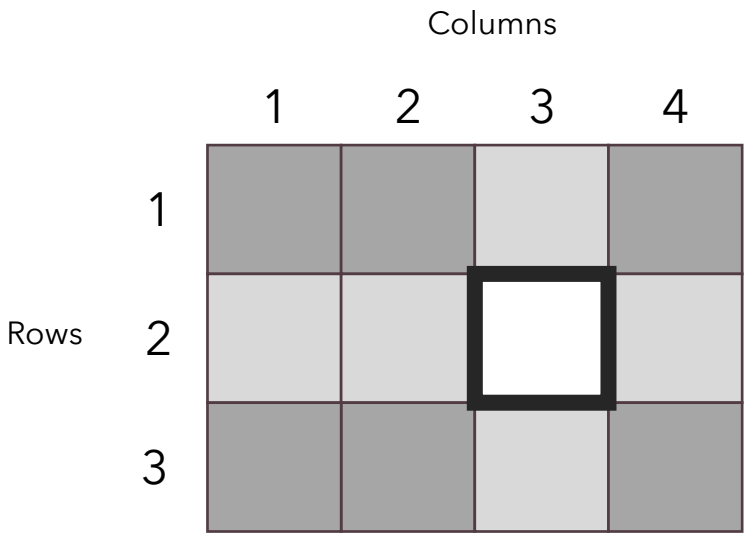
Julien VILLEMEJANE

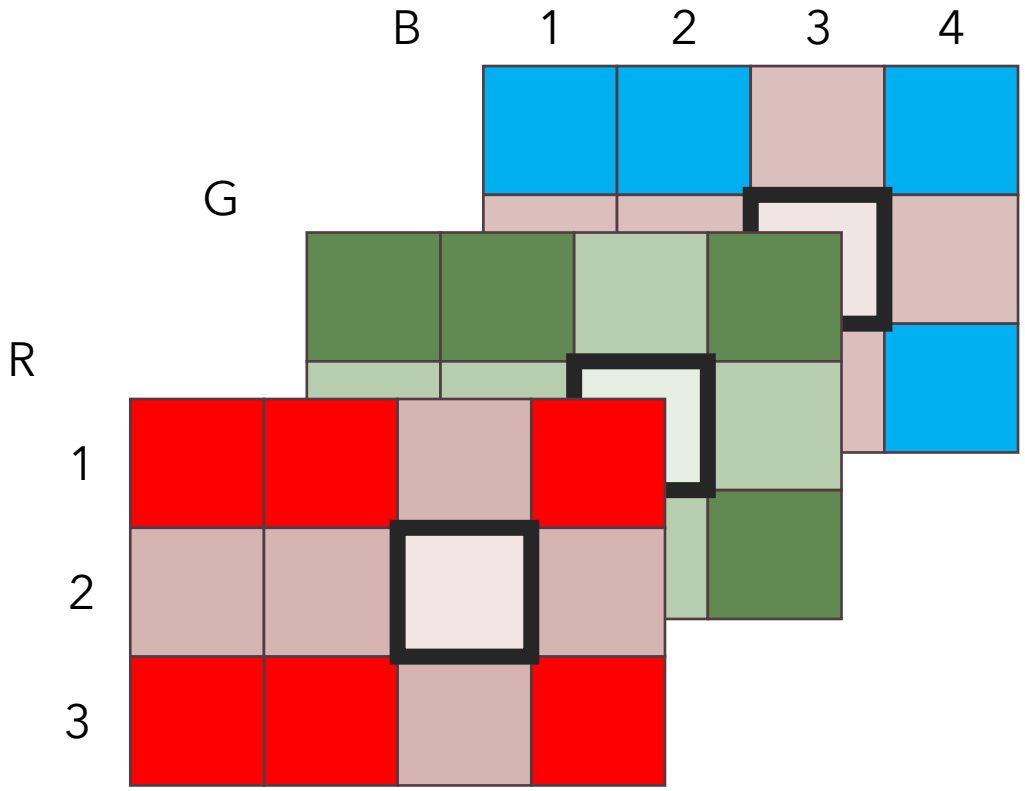


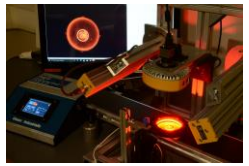
Digital Images / Resolution

Nb of pixels = h x v

GrayScale



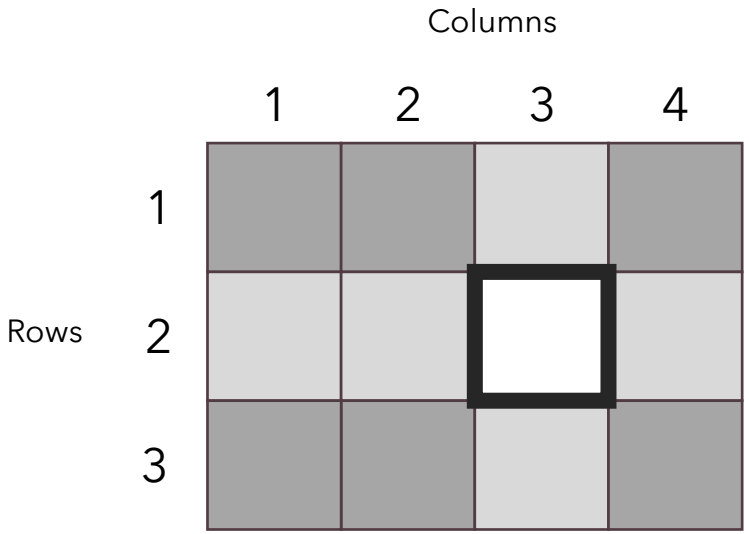




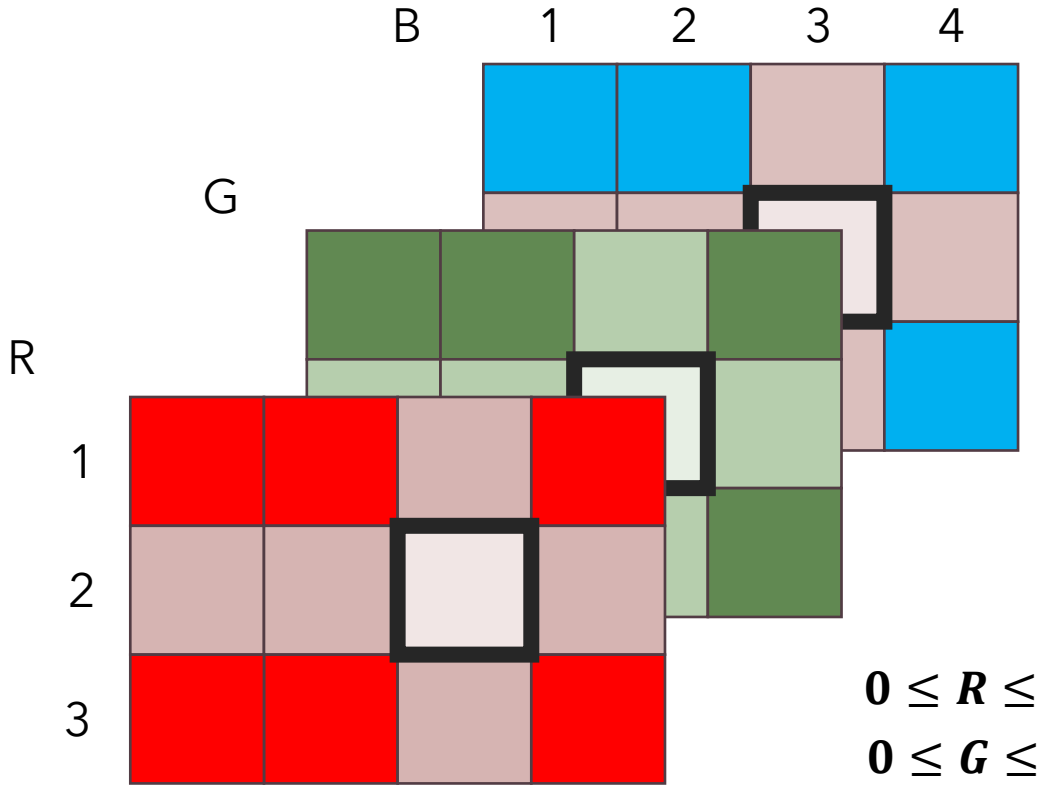
Digital Images / Depth

Nb of pixels = **h** x **v**

GrayScale



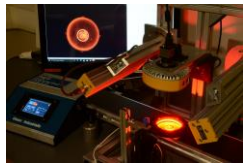
Each pixel is converted into **n bits**.



RGB : 3 channels of 8 bits

$$0 \leq R \leq 255$$
$$0 \leq G \leq 255$$
$$0 \leq B \leq 255$$

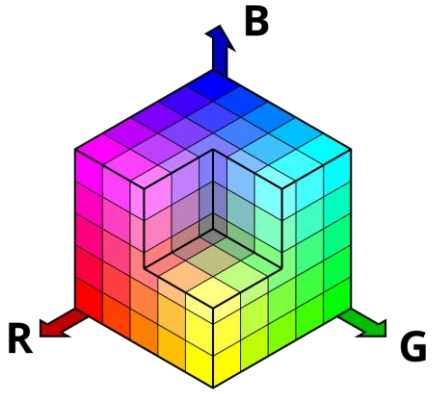
R=200, G=100, G=50



Digital Images / Color Spaces

RGB

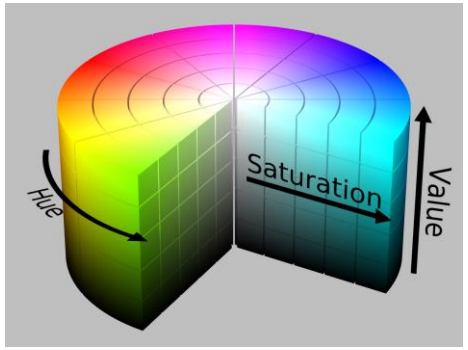
Used primarily in **electronic displays** like computer screens, cameras, and scanners. The combination of these three primary colors at various intensities can produce any color.



HSV

Used in **image editing**. It separates image's color from its brightness.

Hue : type of color
Saturation : intensity of the color
Value : Brightness of the color



CMYK

LAB

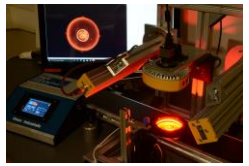
YUV

Color Space

Model for **representing colors** in a consistent and reproducible way

Each color space uses a different method for organizing and describing color, depending on the purpose or application

Images Source : Wikipedia



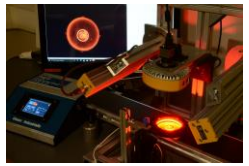
Digital Images / Color Spaces

Color Space

Model for **representing colors** in a consistent and reproducible way

Table 9 from
Segmentation of Images by Color Features: A Survey - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Advantages-and-disadvantages-of-color-spaces_tbl7_323632019 [accessed 10 Oct 2024]

Color space	Advantages	Disadvantages
RGB	Convenient for image acquisition and displaying;	Non-uniform illumination sensitive; Differences between colors is not linear
HSV, HSI	Based on human color perception; Robust before non-uniform illumination; The chromaticity is decoupled from the intensity	Non removable singularities
L*a*b*, L*u*v*	Efficient in measuring small color difference; The chromaticity is decoupled from the intensity;	Singularity problem as other nonlinear transformations
YUV, YCbCr	Efficient coding color information for TV signal.	Due to the linear transformation, correlation between the component channels exists, although not as high as the RGB space



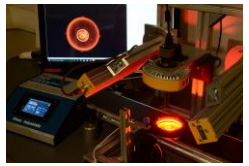
Pre-Processing

Acquisition

Pre Processing

Noise Reduction / Filtering
Contrast Enhancement
Normalization

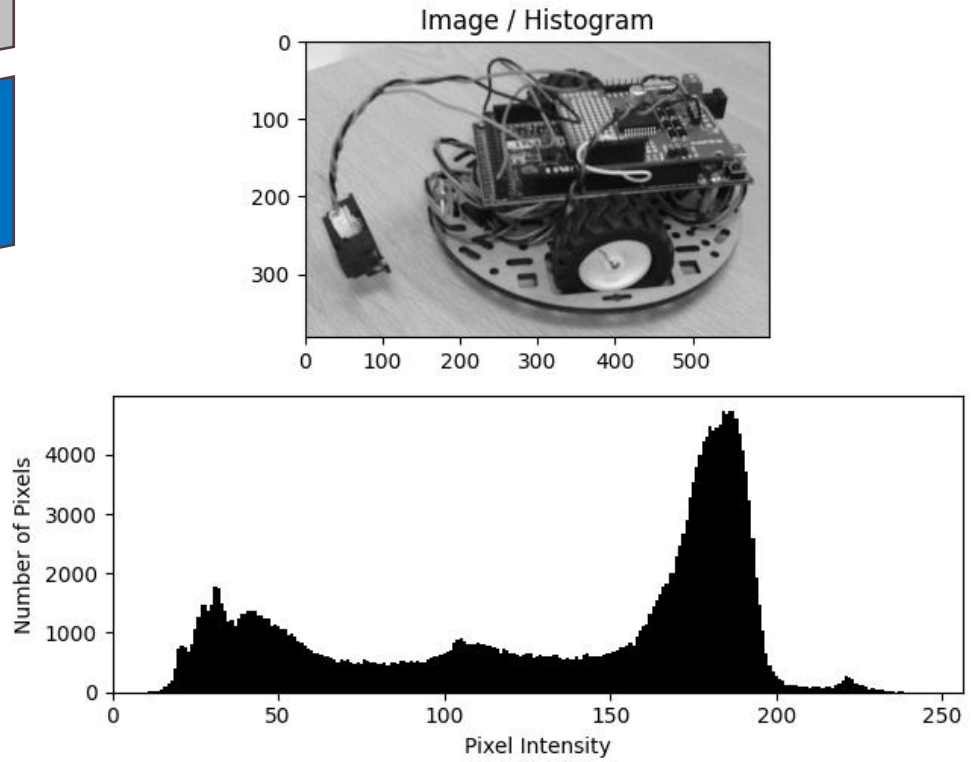
Ensuring clarity and reducing unwanted information
Making features of interest stand out
Standardizing the image scale or intensity



OpenCV / Histogram of an image

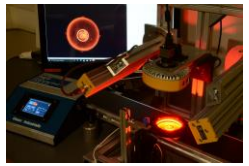
Acquisition

Pre Processing



Histogram

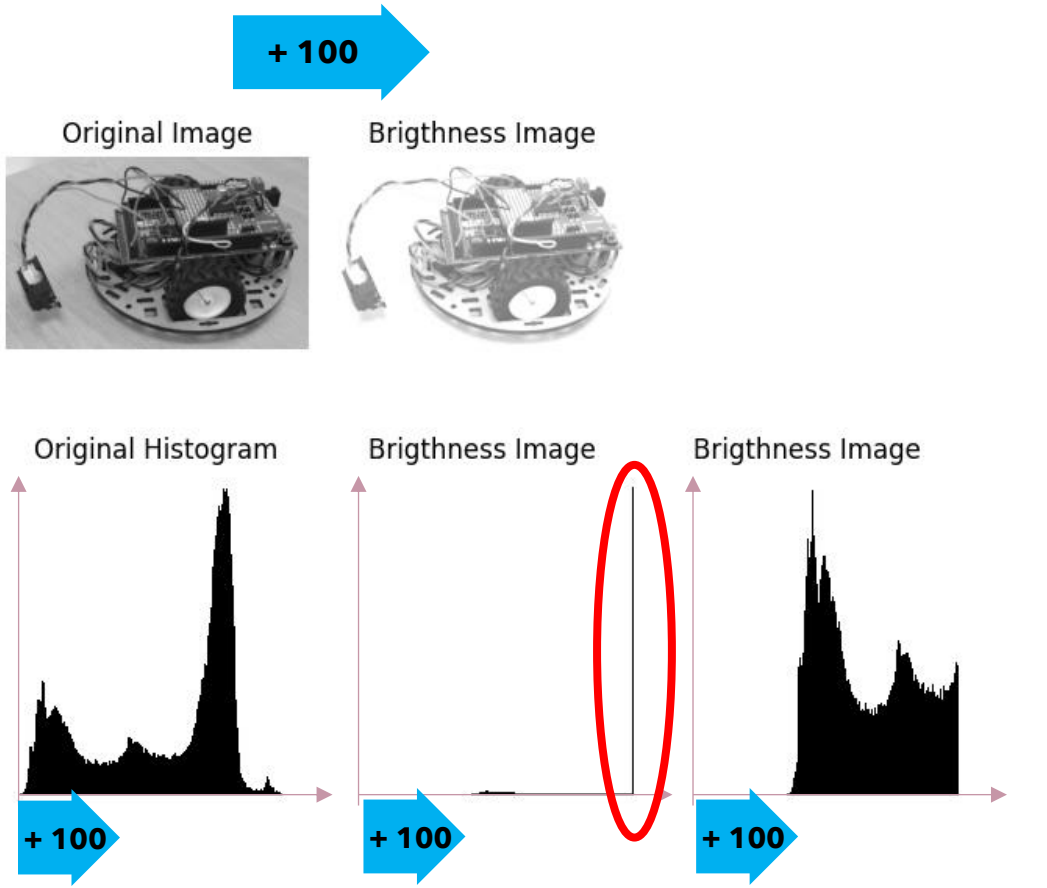
Graphical representation that shows the **distribution of pixel intensity values** in an image



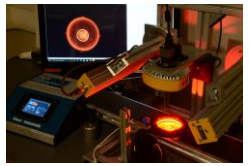
OpenCV / Contrast and Brightness

Acquisition

Pre Processing



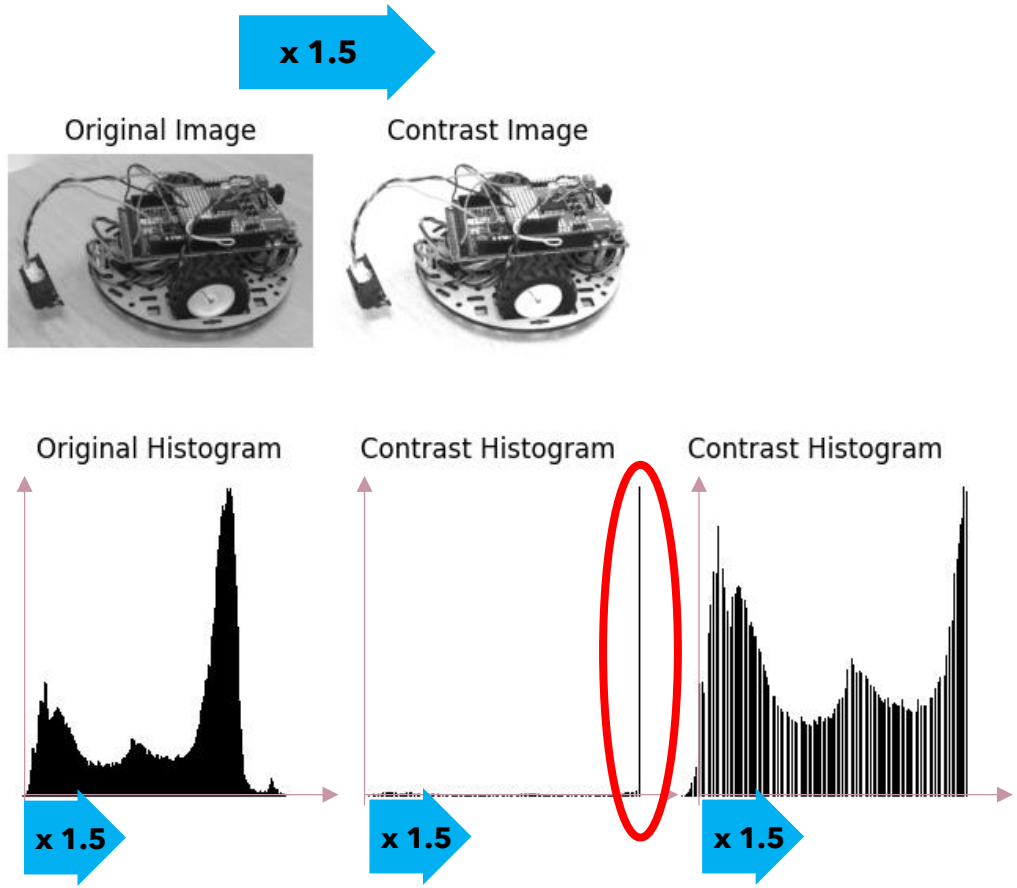

```
new_img = cv2.convertScaleAbs(image, beta=100)
```



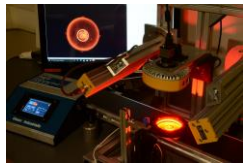
OpenCV / Contrast and Brightness

Acquisition

Pre Processing




```
new_img = cv2.convertScaleAbs(image, alpha=1.5)
```



OpenCV / Convolution

Acquisition

Pre Processing

Convolution (filter)

kernel

-1	0	-2
1	5	1
-2	0	-1

original image

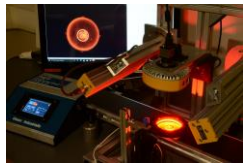
5	8	4	2	3	1	5
9	5	1	8	7	6	2
5	7	1	5	6	8	7
5	8	2	8	4	3	3
5	6	6	7	2	5	1

5	8	4	2	3	1	5
9	5	1	8 x -1	7 x 0	6 x -2	2
5	7	1	5 x 1	6 x 5	8 x 1	7
5	8	2	8 x -2	4 x 0	3 x -1	3
5	6	6	7	2	5	1

R = -8 + 0 - 12 + 5 + 30 + 8 - 16 + 0 - 3
R = 4

filtered image

				4		



OpenCV / Convolution Kernel *(or Structuring Element)*

Acquisition

Pre Processing

kernel = cv2.***getStructuringElement***(cv2.MORPH_xx, (M,N))

Cross Kernel

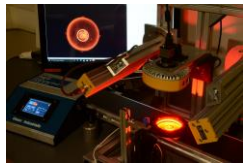
0	0	1	0	0
0	0	1	0	0
1	1	1	1	1
0	0	1	0	0
0	0	1	0	0

cv2.**MORPH_CROSS**

Rect Kernel

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

cv2.**MORPH_RECT**



OpenCV / Erosion and Dilation

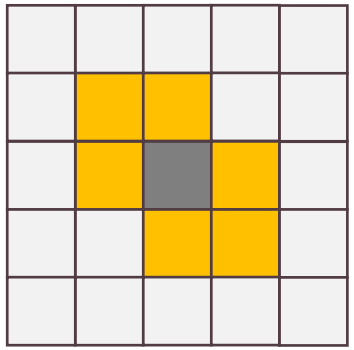
INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

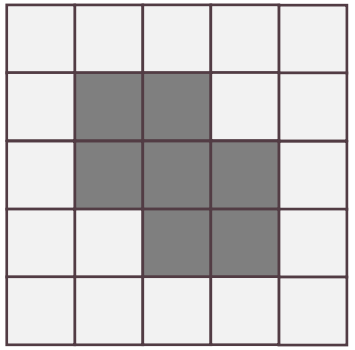
Acquisition

Pre Processing

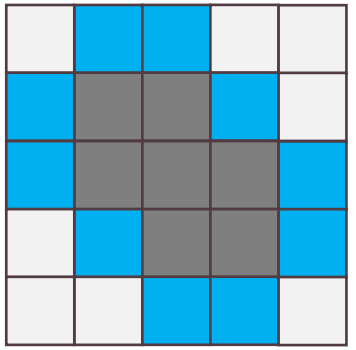
Original pixels

Removed pixels





Added pixels



kernel

0	1	0
1	1	1
0	1	0

Erosion

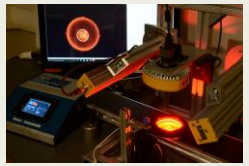
Shrinking the foreground by **removing pixels** to the boundaries of objects

https://www.youtube.com/watch?v=fmyE7DiaIYQ

Dilation

Enlarging the foreground by **adding pixels** to the boundaries of objects

https://www.youtube.com/watch?v=xO3ED27rMHs

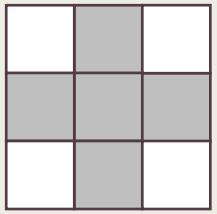


OpenCV / Erosion and Dilation - EXERCICES

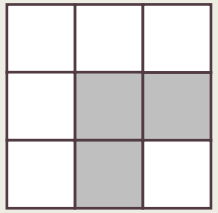
INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

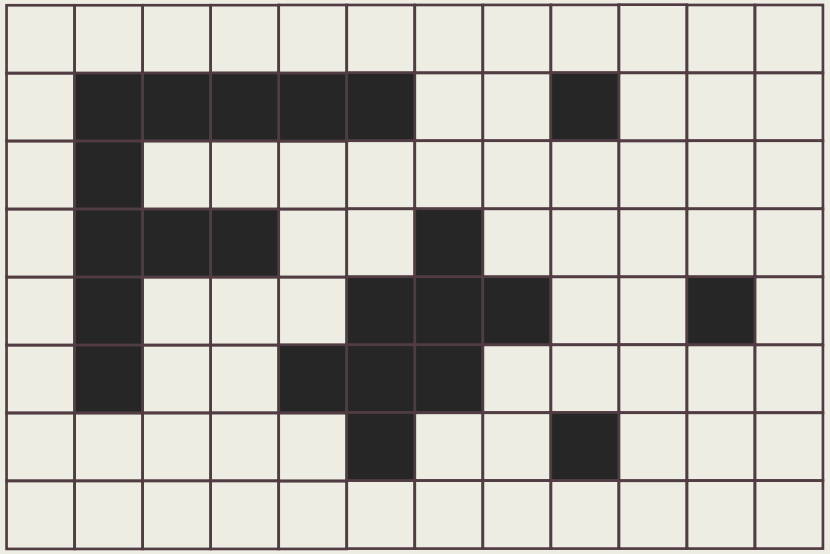
Erosion

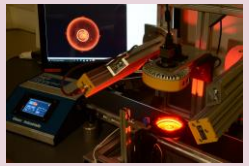
CROSS



CORNER





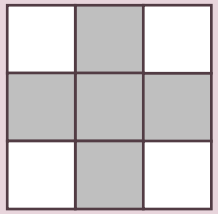


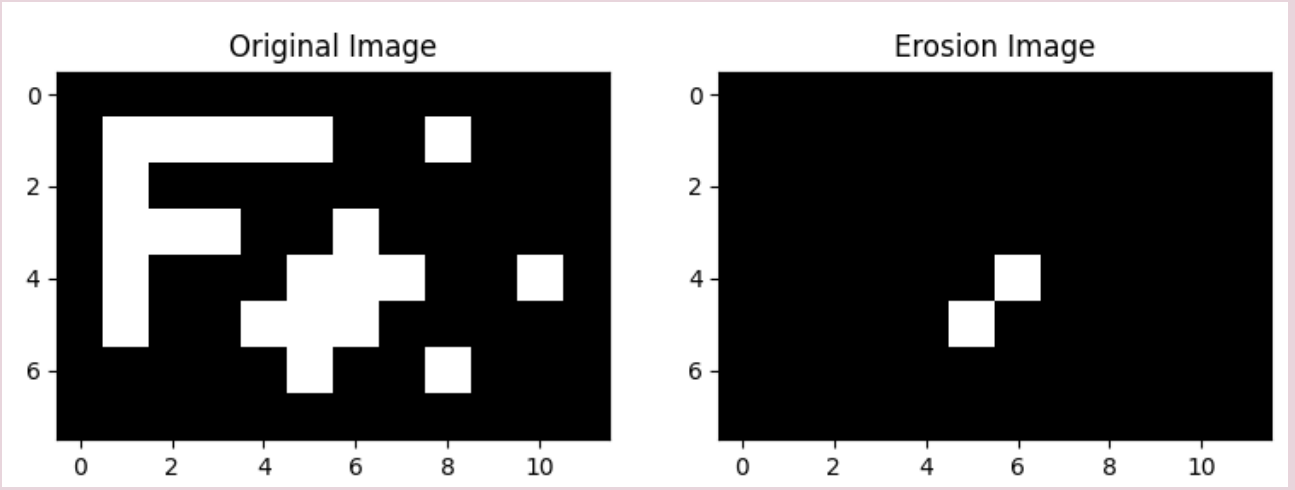
OpenCV / Erosion and Dilation - REPONSES

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

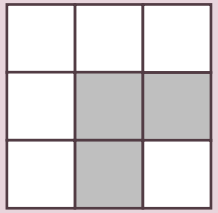
Erosion

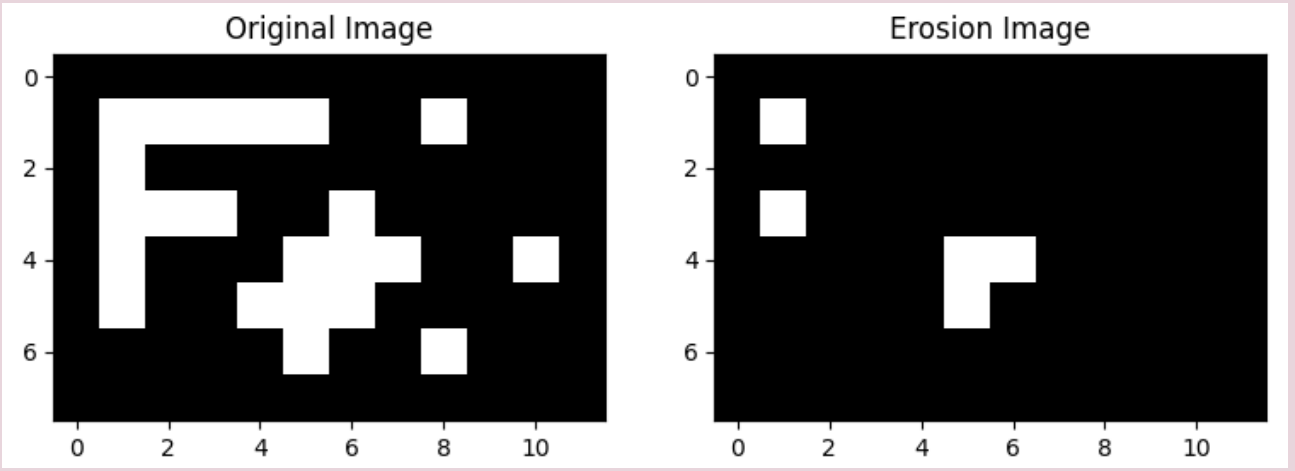
CROSS

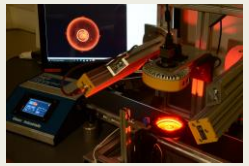




CORNER



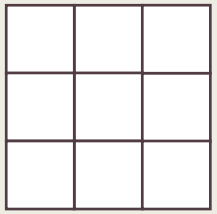




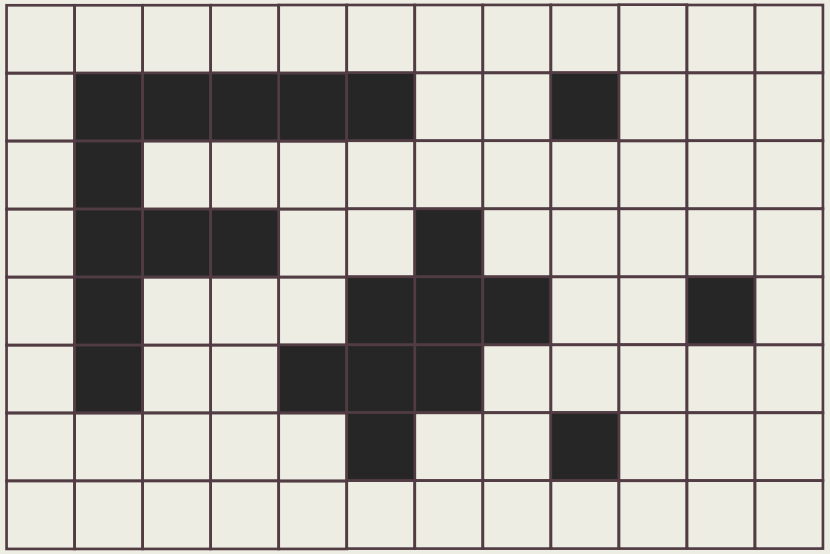
OpenCV / Erosion and Dilation - EXERCICES

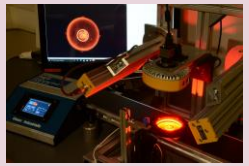
Erosion

KERNEL ??



How to design a kernel to detect horizontal lines ?



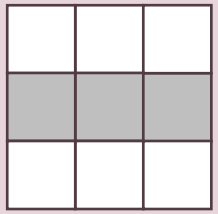


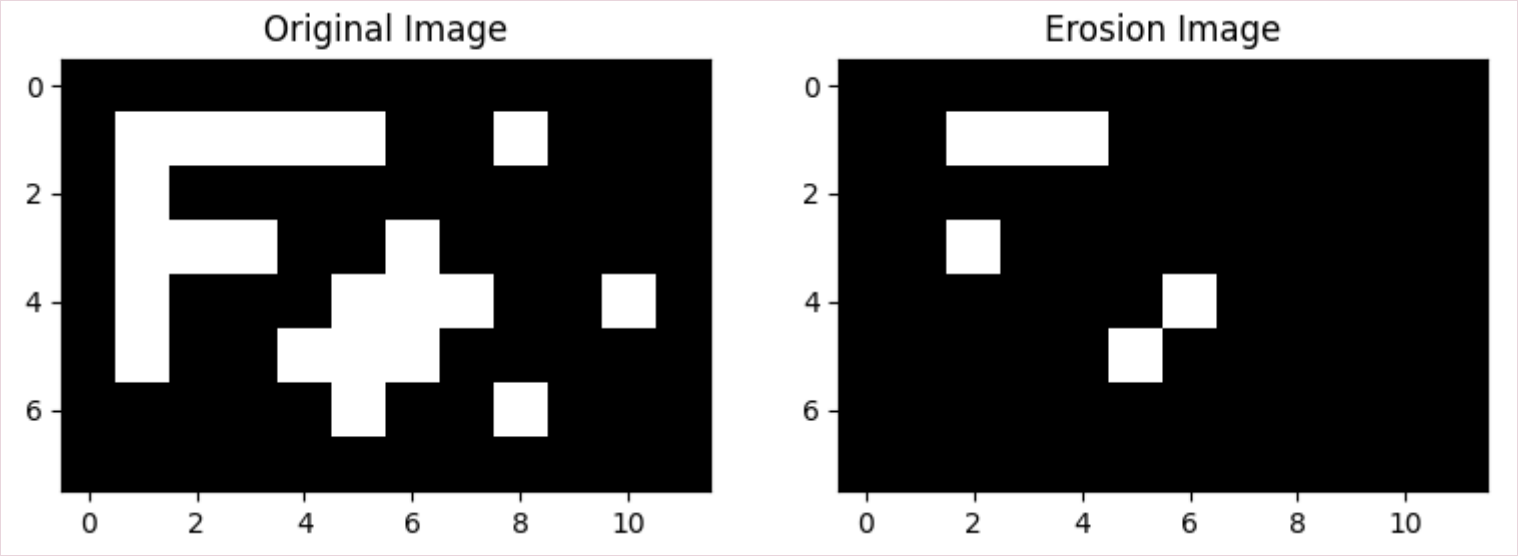
OpenCV / Erosion and Dilation - REPONSES

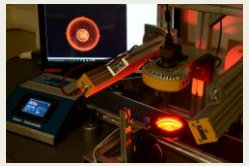
INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

Erosion

KERNEL



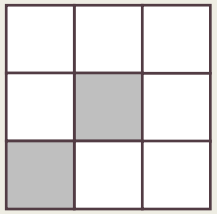




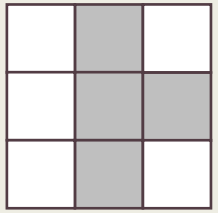
OpenCV / Erosion and Dilation - EXERCICES

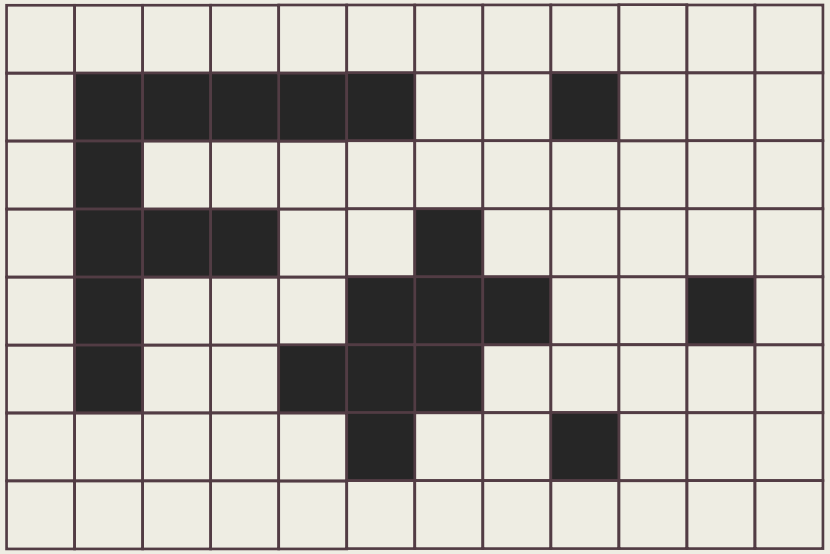
Dilation

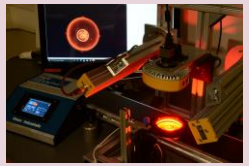
KER 1



KER 2





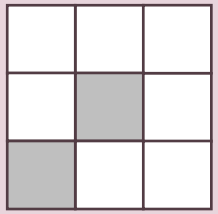


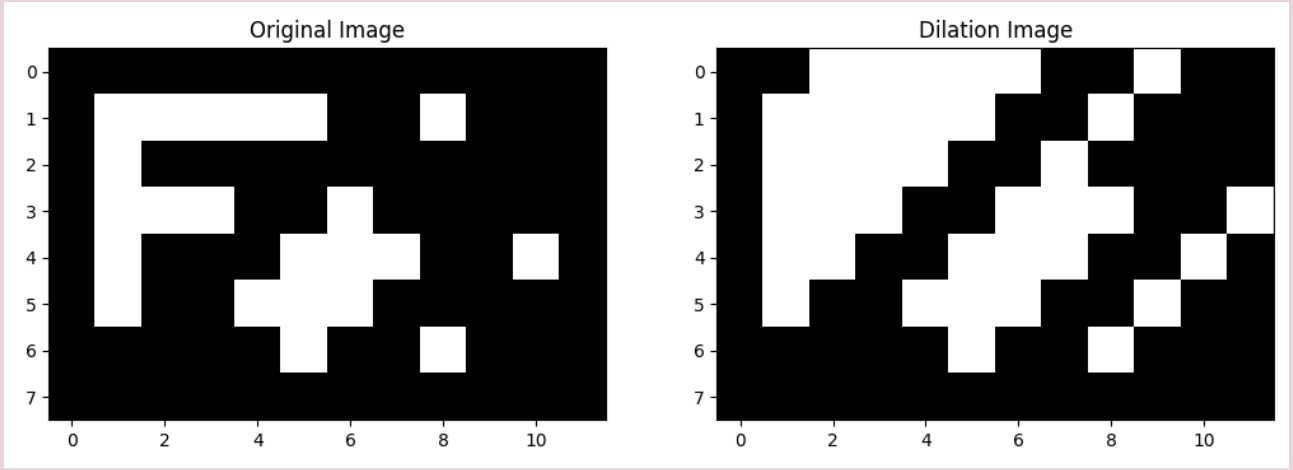
OpenCV / Erosion and Dilation - REPONSES

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

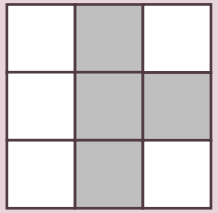
Erosion

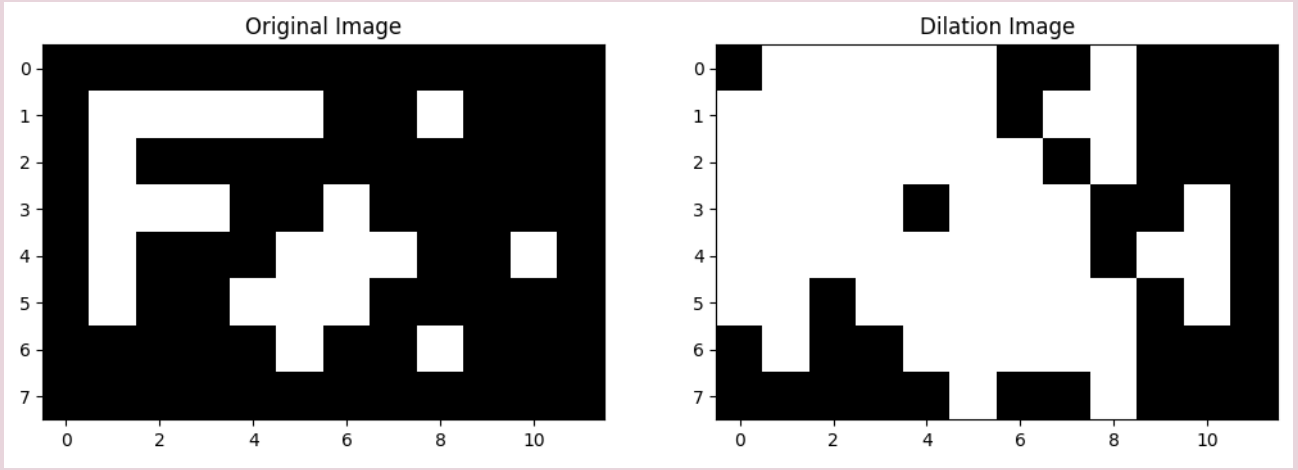
KER 1

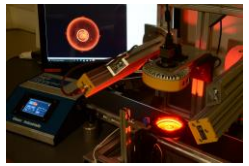




KER 2



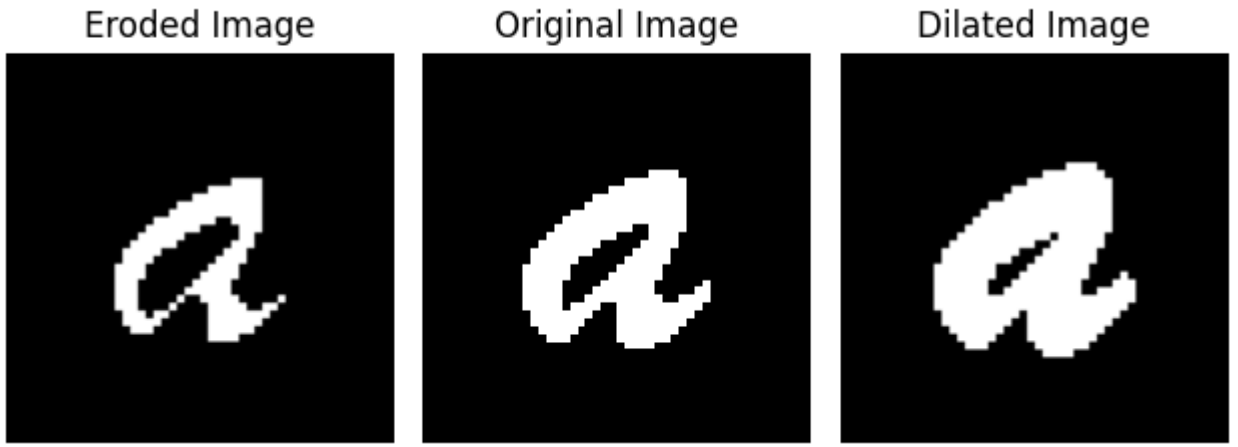




OpenCV / Erosion and Dilation

Acquisition

Pre Processing



kernel

0	1	0
1	1	1
0	1	0

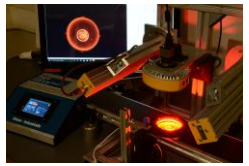
Erosion

Shrinking the foreground by **removing pixels** to the boundaries of objects

Dilation

Enlarging the foreground by **adding pixels** to the boundaries of objects

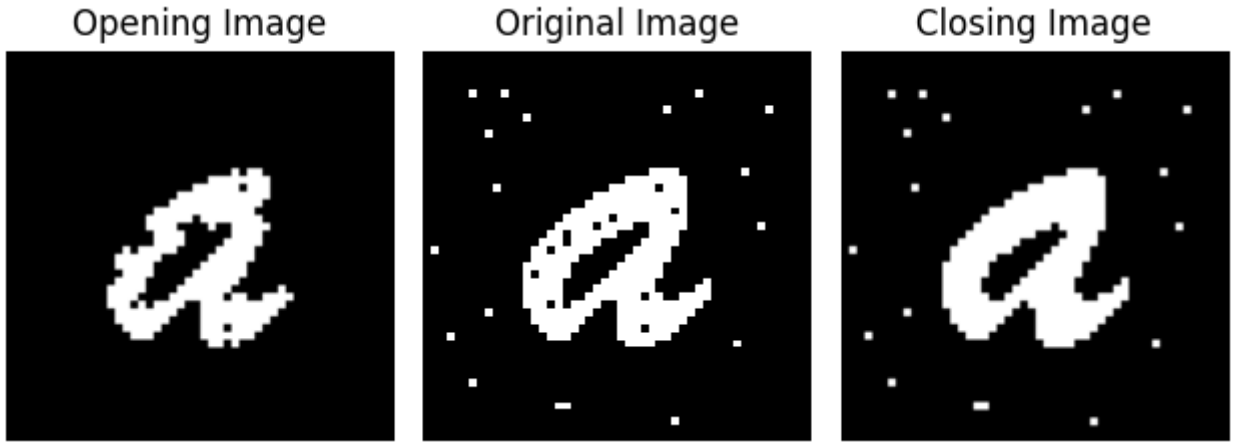
```
eroded_image = cv2.erode(image, kernel, iterations=1)
dilated_image = cv2.dilate(image, kernel, iterations=1)
```



OpenCV / Opening and Closing morphological transforms

Acquisition

Pre Processing



kernel

0	1	0
1	1	1
0	1	0

Opening

Erosion then **Dilation**

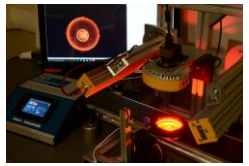
Removing small objects, in the background

Closing

Dilation then **Erosion**

Filling in small holes in the foreground

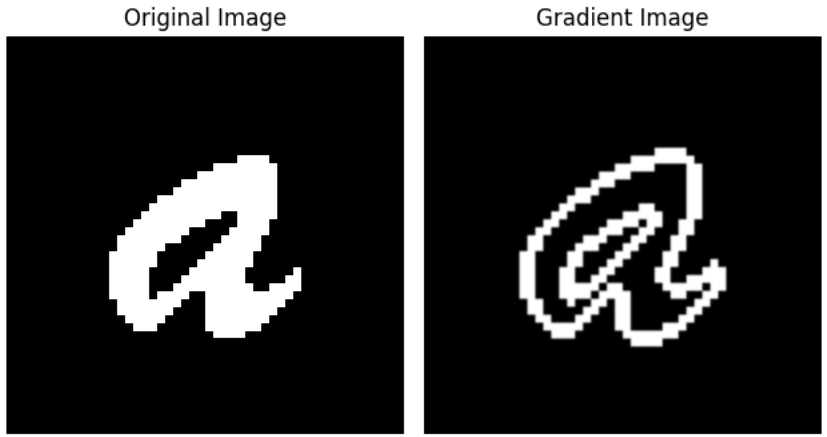
```
opening_image = cv2.morphologyEx(image, cv2.MORPH_OPEN, kernel)
closing_image = cv2.morphologyEx(image, cv2.MORPH_CLOSE, kernel)
```



OpenCV / Gradient morphological transforms

Acquisition

Pre Processing



kernel

0	1	0
1	1	1
0	1	0

Gradient

Difference between a **dilation** and an **erosion**

Unknown pixels classification : background or foreground ?

gradient_image = cv2.**morphologyEx**(image, ***cv2.MORPH_GRADIENT***, kernel)

cv2.imshow('Image Window', image)

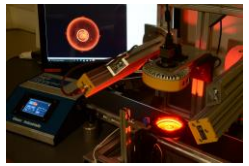


Image processing with OpenCV

OpenCV / Blur and mean

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

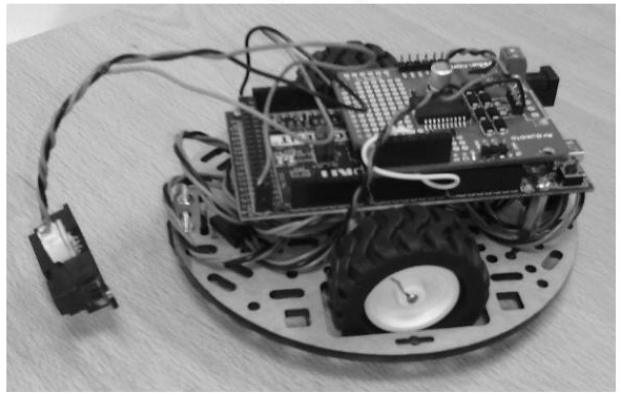
Acquisition

Pre Processing

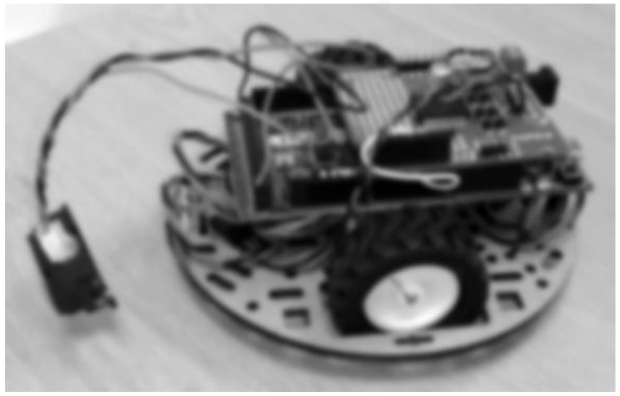
```
kernel_size = (N,M)

blurred_image_gauss = cv2.GaussianBlur(image, kernel_size, 0)
blurred_image_box = cv2.blur(image, kernel_size)
```

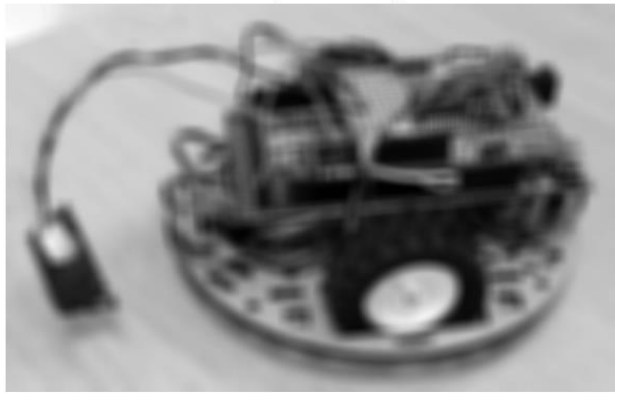
Original Image



Gaussian Blur Image



Median/Box Blur Image



1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Gaussian Kernel
(x 1/273)

Mean Kernel (x 1/(N*M))

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Removing irrelevant details

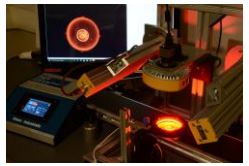


Image processing

Goal of processing an image

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

Acquisition

Pre Processing

Segmentation

Thresholding
Edge Detection
Region of Interest Selection

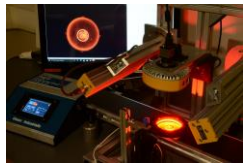
Feature Extraction

Geometric Features
Texture Analysis
Color Analysis

https://docs.opencv.org/4.x/d3/db4/tutorial_py_watershed.html

Isolating objects of ROI / Separating product from background
Identifying boundaries and contours
Focusing only on relevant portions of the image

Extracting measurements (size, shape, position…)
Recognizing patterns, symbols, points of interest



Fourier Transform and filtering

- Acquisition
- Pre Processing
- Segmentation
- Feature Extraction