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

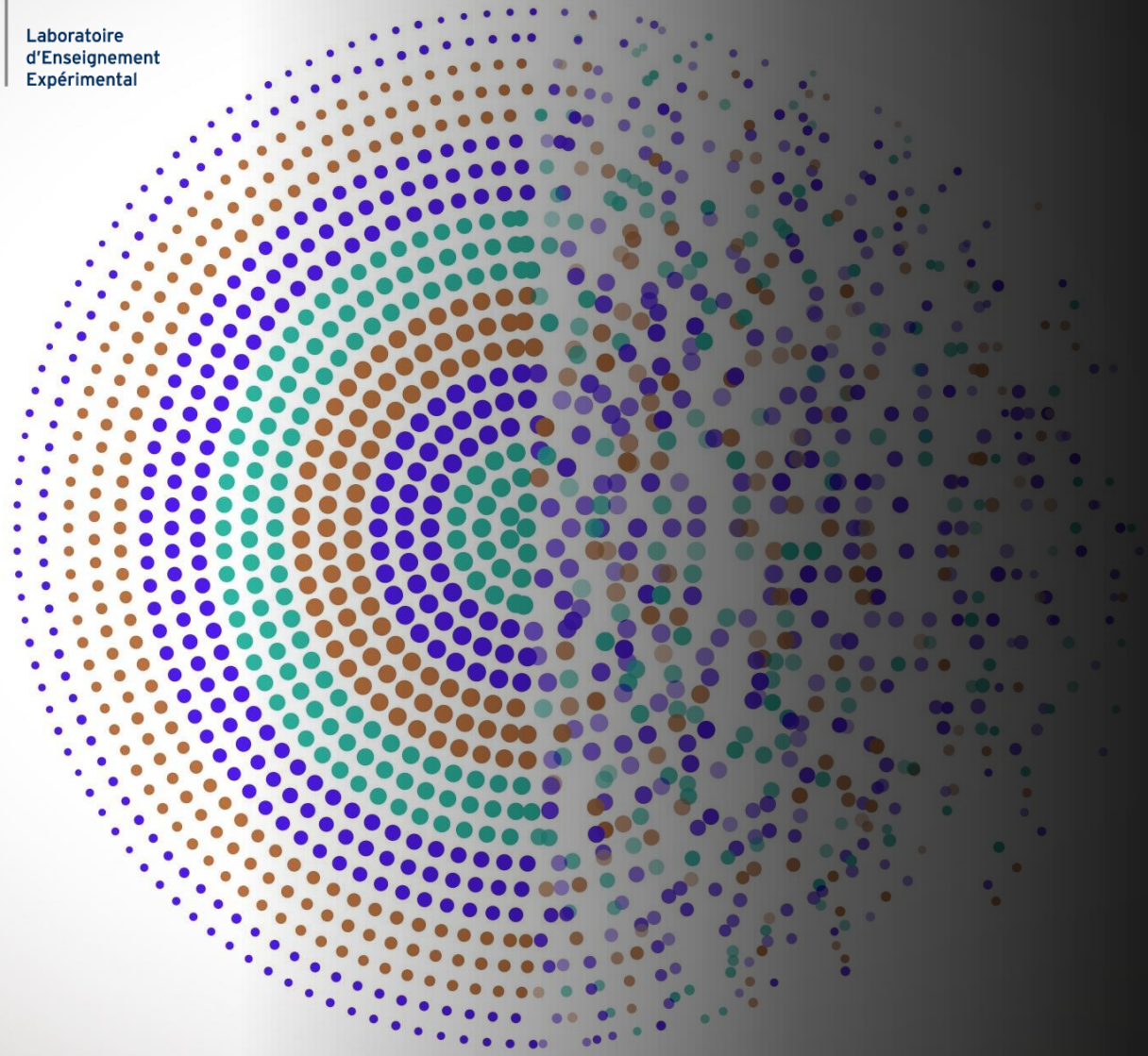


Image processing with OpenCV

Institut d'Optique – Engineers Training

Semester 6 – Digital Interface

Julien VILLEMEJANE

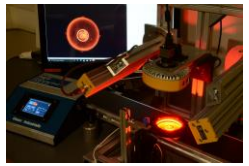


Image processing / OpenCV

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

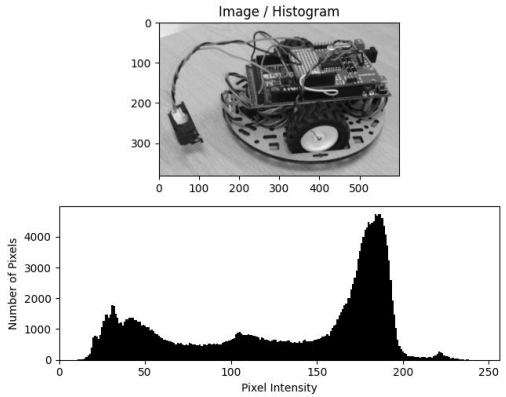
- At the end of this training, the learners will be able to:

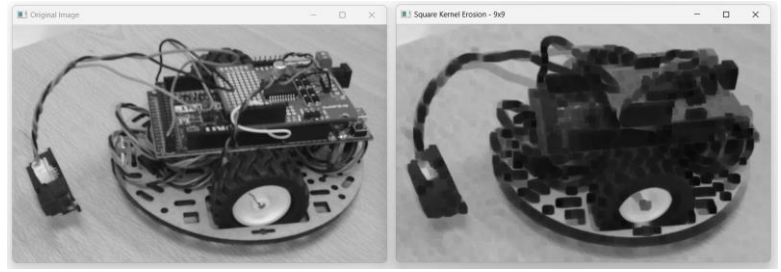
Use basic building blocks of OpenCV

Open an image
Create the histogram of an image

Apply basic transforms on images
- blur, filters
- erosion, dilation, opening, closing

Display the contour of objects





LEnsE.TECH

https://iogs-lense-training.github.io/image-processing/

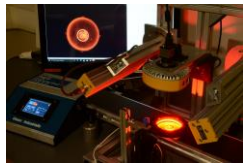
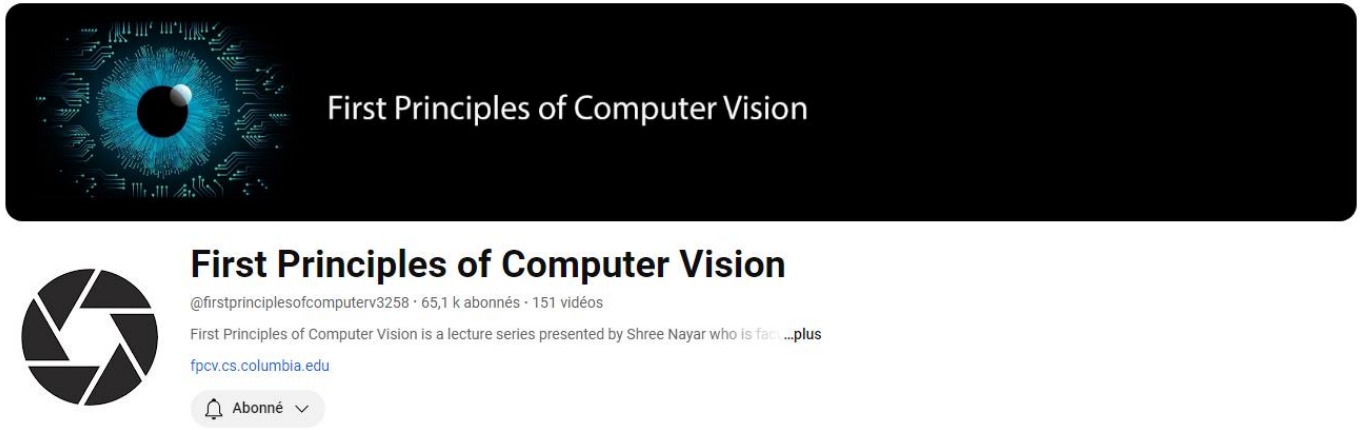


Image processing

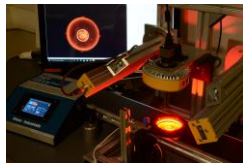
What is image processing for ?

Why is image processing required in a machine vision chain ?
For industrial inspection applications ?

What are the **main processes** ?
What is the goal of each of them ?
What is the ideal **workflow** ?



https://www.youtube.com/@firstprinciplesofcomputerv3258



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

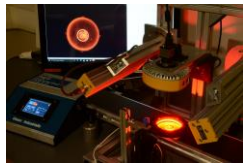


Image processing

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

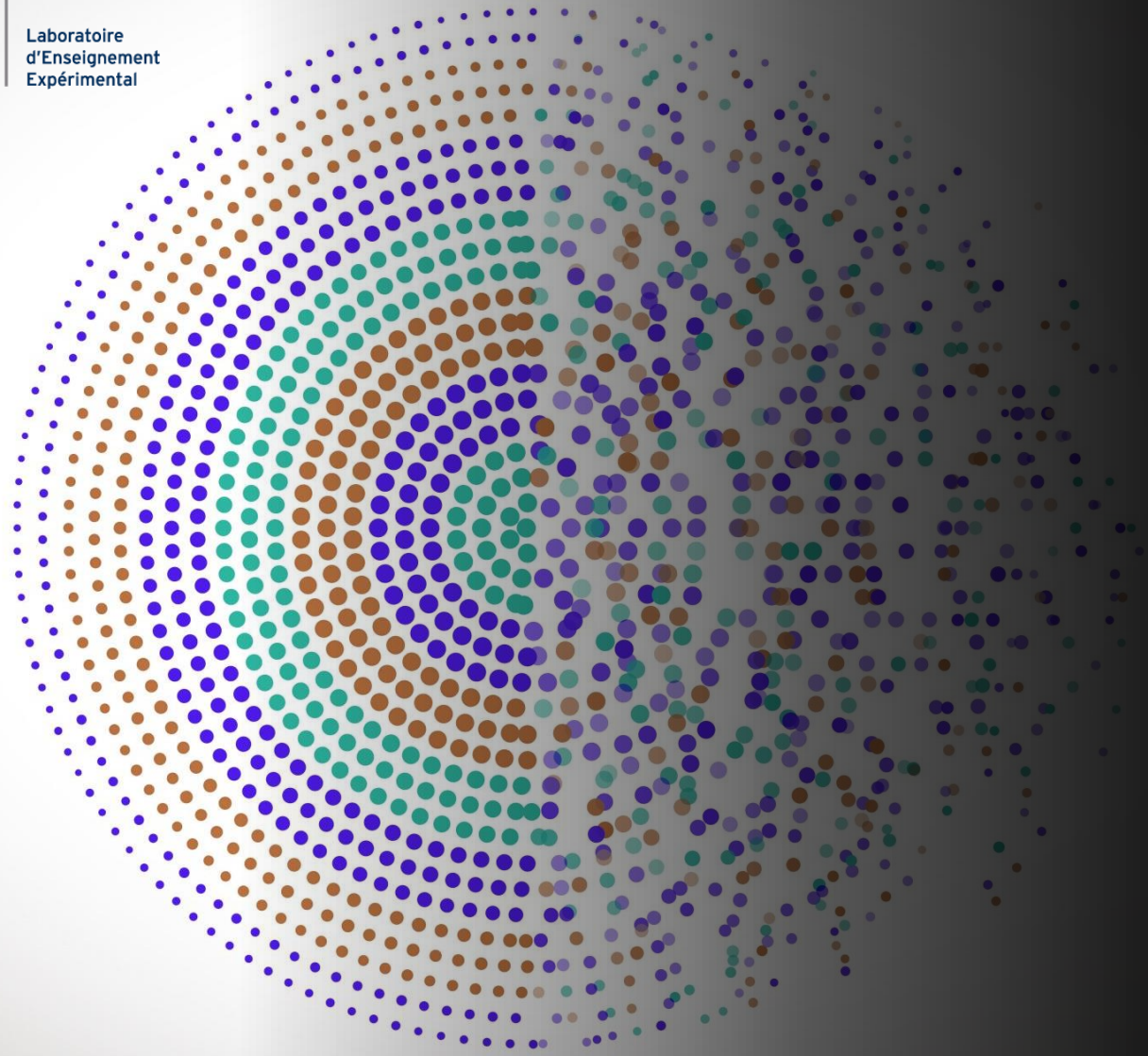
Extracting measurements (size, shape, position…)
Recognizing patterns, symbols, points of interest

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

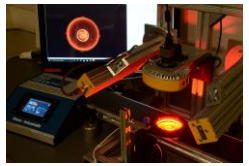


OpenCV

Institut d'Optique – Engineers Training

Semester 6 – Digital Interface

Julien VILLEMEJANE



What is OpenCV ?

Open Source Computer Vision Library

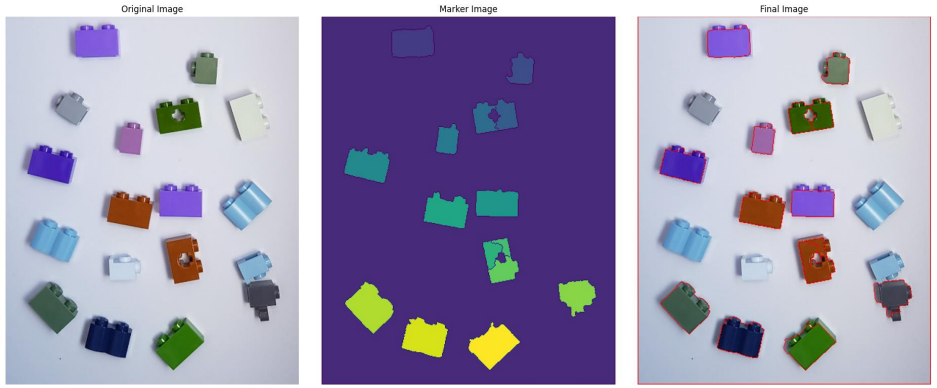
An open source **computer vision** and machine learning software **library**

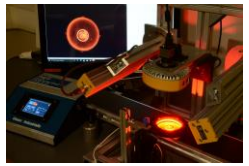
supporting multiple programming languages such as Python, C++, Java, and MATLAB

OpenCV

http://opencv.org

Image Processing	*Filtering, Edge detection, Image transformations…*
Object Recognition	*Detecting objects in images and videos*
CV Algorithms	*Motion tracking, 3D reconstruction, Augmented reality*
Machine Learning	*Image classification, Pattern recognition, Scene Understanding*





Python 3 and OpenCV

Installing OpenCV for Python 3

```
pip install opencv-python
```

Testing OpenCV importation in a script

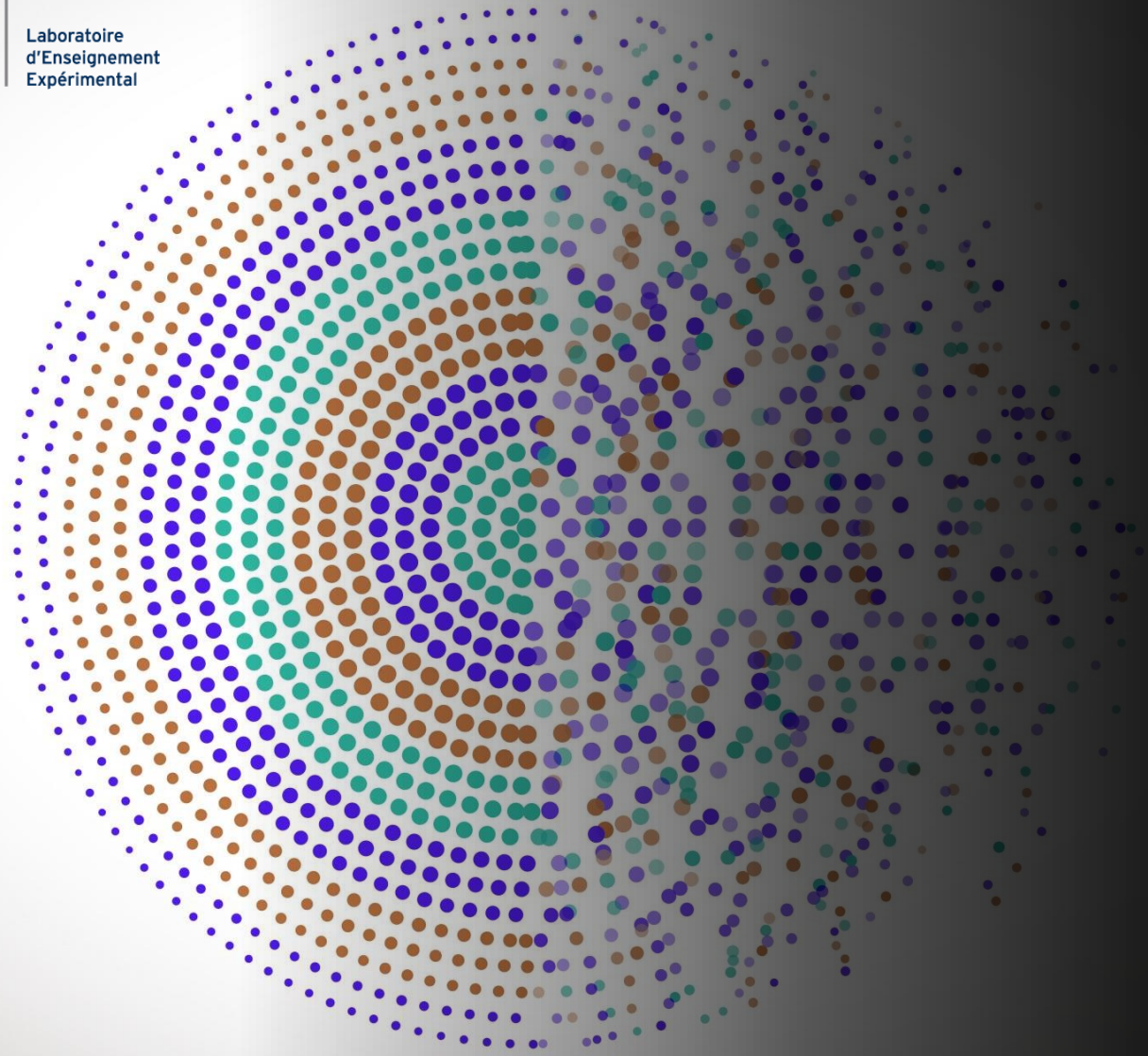
```
import cv2
Cv2.__version__
```

http://opencv.org

LEnsE.TECH

https://iogs-lense-training.github.io/image-processing/

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

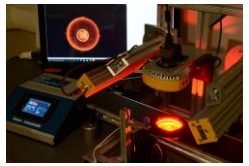


Digital Images and Processing

Institut d'Optique – Engineers Training

Semester 6 – Digital Interface

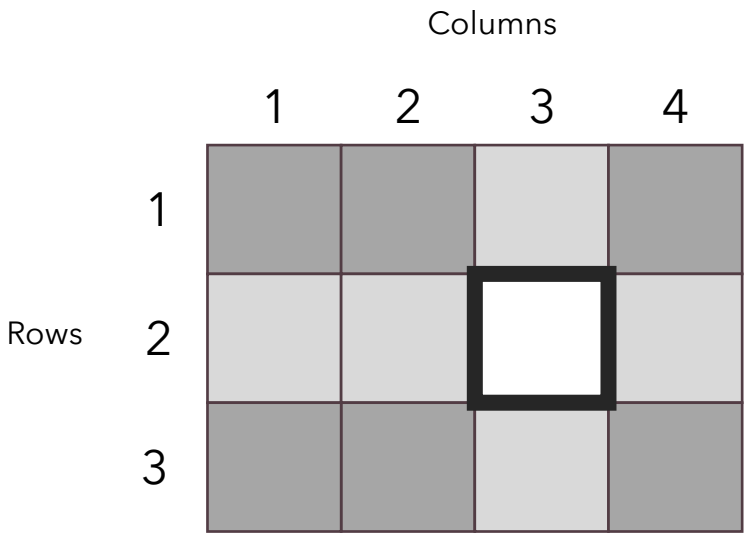
Julien VILLEMEJANE

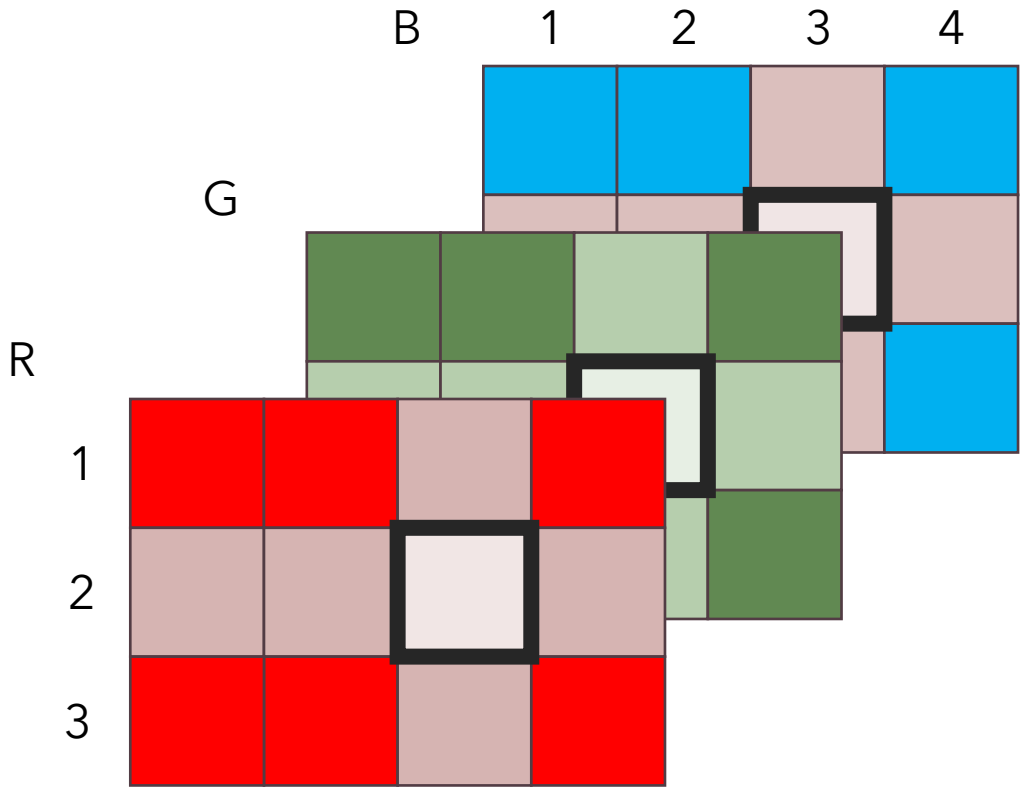


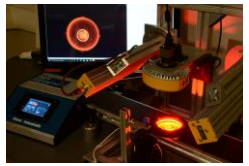
Digital Images / Resolution

Nb of pixels = h x v

GrayScale



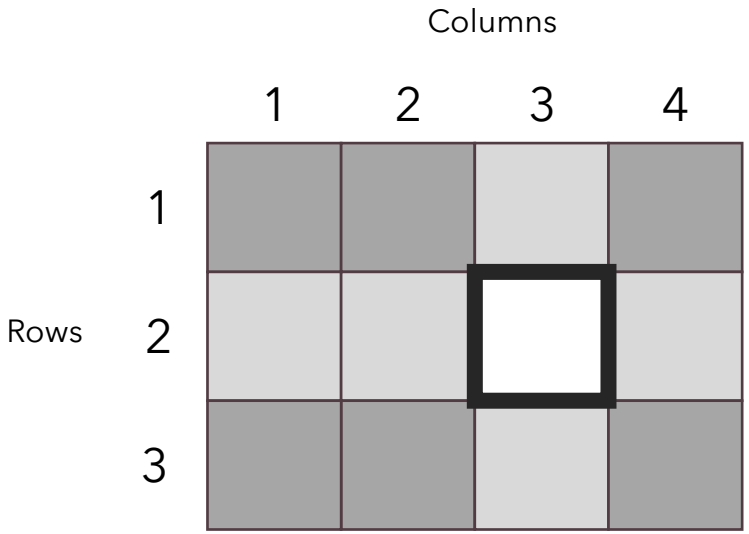




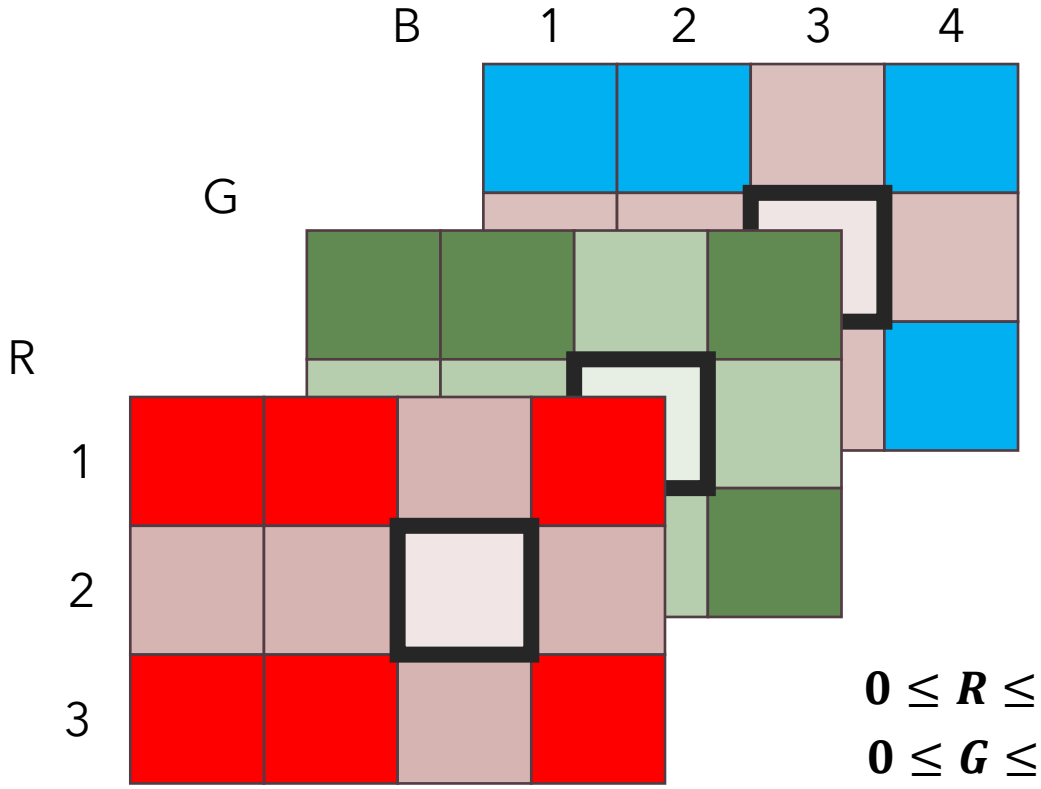
Digital Images / Depth

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

GrayScale



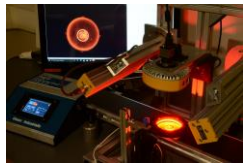
Each pixel is converted into **n bits**.



RGB : 3 channels of 8 bits

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

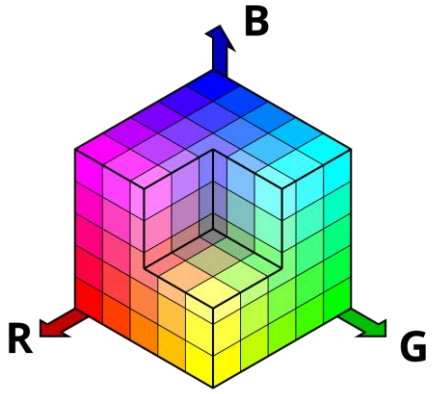
R=200, G=100, G=50



Digital Images / Color Spaces

RGB

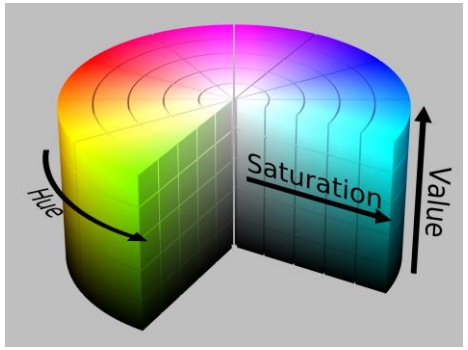
Used primarily in **electronic displays** like computer screens, cameras, and scanners. The combination of these three primary colors at various intensities can produce any color.



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

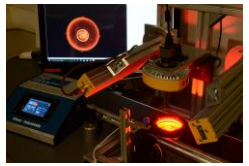
YUV

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



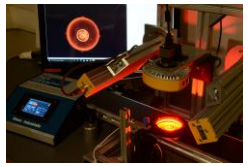
Digital Images / Color Spaces

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



OpenCV / Open and display an Image

Acquisition

import cv2

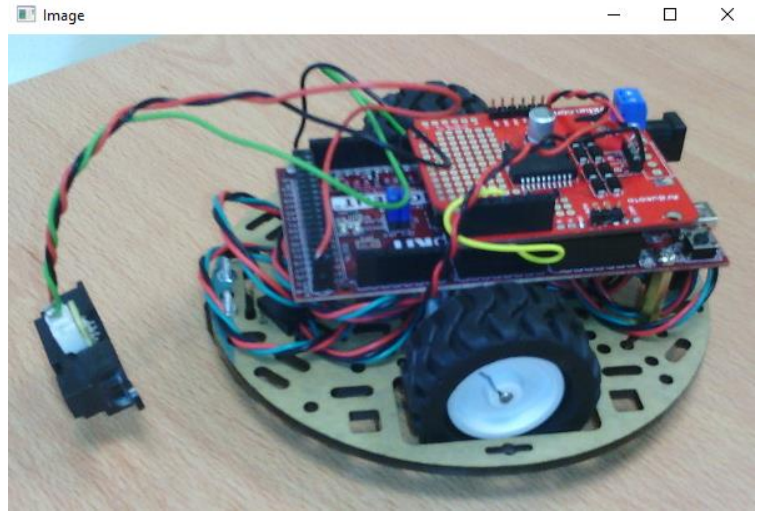
image_rgb = cv2.***imread***(*'path/to/image.png'*)
image_gray = cv2.***imread***(*'path/to/image.png'*, *cv2.IMREAD_GRAYSCALE*)

```
image = cv2.imread("../../_data/robot.pgm")
print(type(image))
```

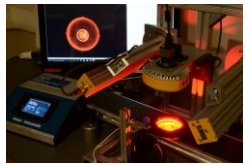
```
<class 'numpy.ndarray'>
```

```
print(image.shape)
```

```
(382, 600, 3)
```



cv2.***imshow***('*Image* ', image_rgb)
cv2.**waitKey**(0)



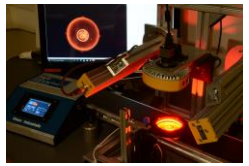
Pre-Processing

Acquisition

Pre Processing

Noise Reduction / Filtering
Contrast Enhancement
Normalization

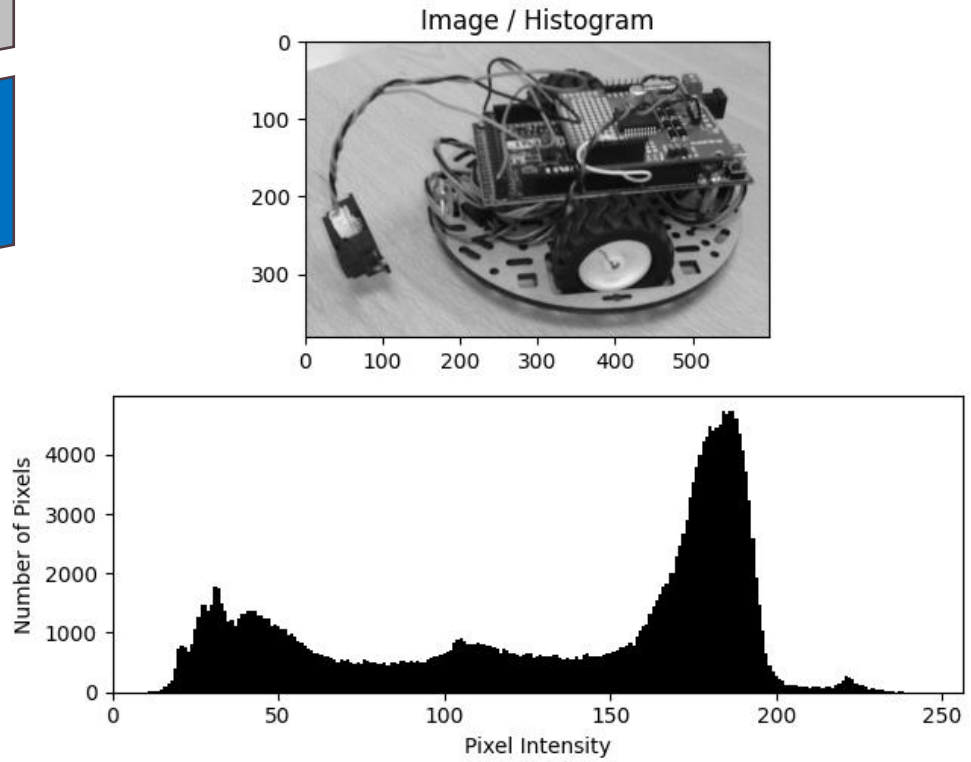
Ensuring clarity and reducing unwanted information
Making features of interest stand out
Standardizing the image scale or intensity



OpenCV / Histogram of an image

Acquisition

Pre Processing

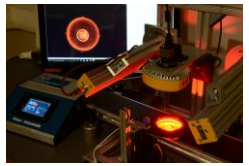


Histogram

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

cv2.**calcHist**([image], [chan], Mask, [bins_nb], [min, max])

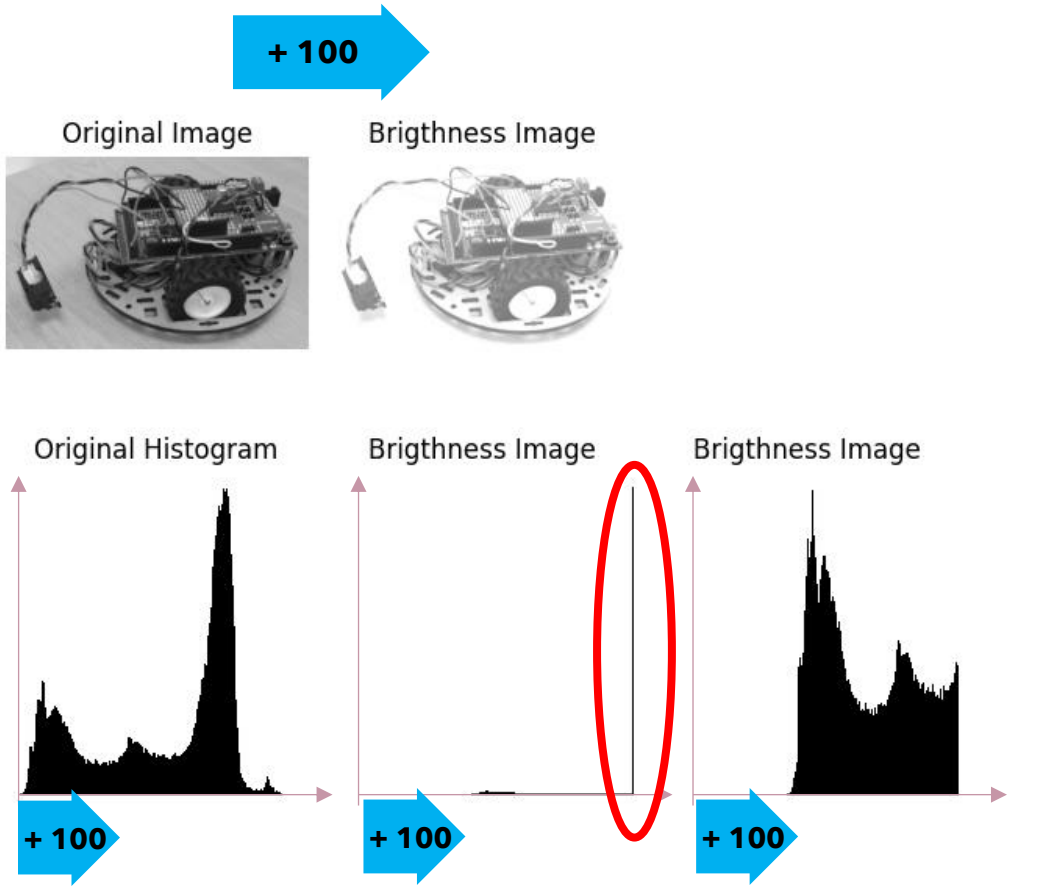
```
histogram = cv2.calcHist([image], [0], None, [256], [0, 256])
```



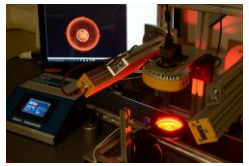
OpenCV / Contrast and Brightness

Acquisition

Pre Processing



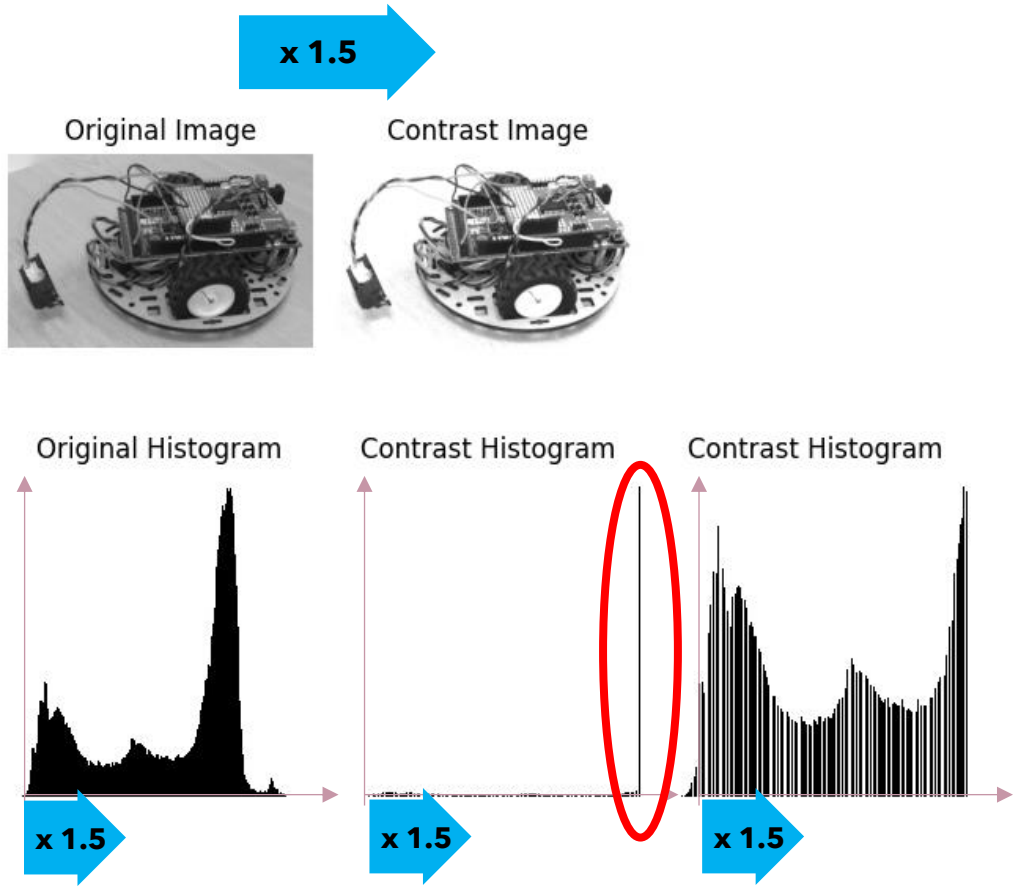

```
new_img = cv2.convertScaleAbs(image, beta=100)
```



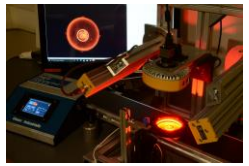
OpenCV / Contrast and Brightness

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

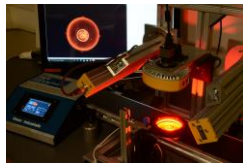
5	8	4	2	3	1	5
9	5	1	8	7	6	2
5	7	1	5	6	8	7
5	8	2	8	4	3	3
5	6	6	7	2	5	1

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

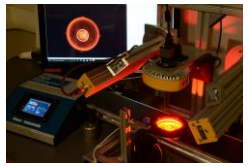
0	0	1	0	0
0	0	1	0	0
1	1	1	1	1
0	0	1	0	0
0	0	1	0	0

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

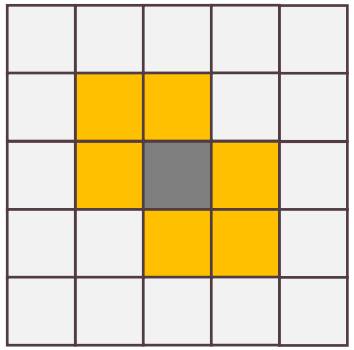
INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

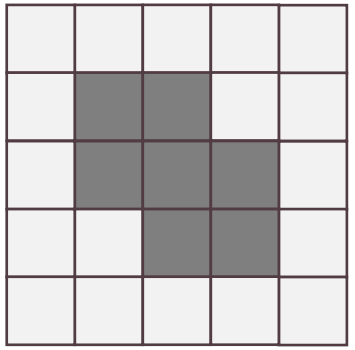
Acquisition

Pre Processing

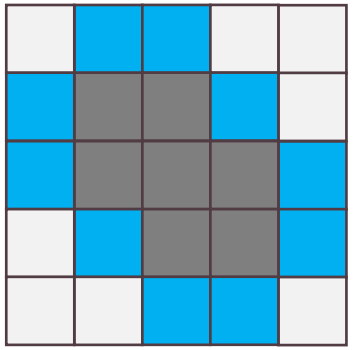
Original pixels

Removed pixels





Added pixels



kernel

0	1	0
1	1	1
0	1	0

Erosion

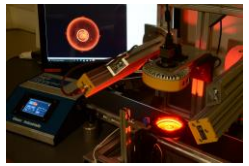
Shrinking the foreground by **removing pixels** to the boundaries of objects

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

Dilation

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

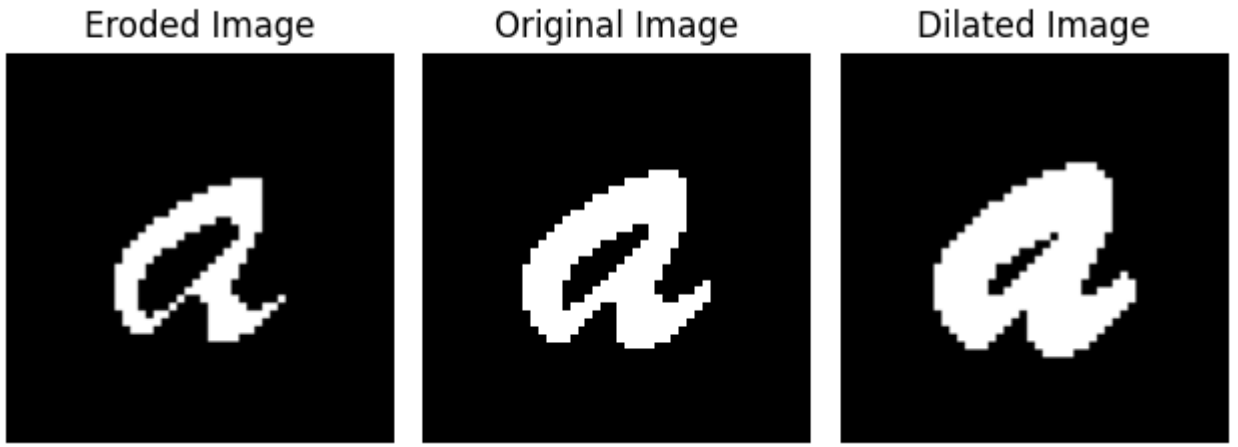
https://www.youtube.com/watch?v=xO3ED27rMHs



OpenCV / Erosion and Dilation

Acquisition

Pre Processing



kernel

0	1	0
1	1	1
0	1	0

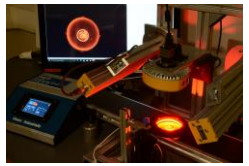
Erosion

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

Dilation

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

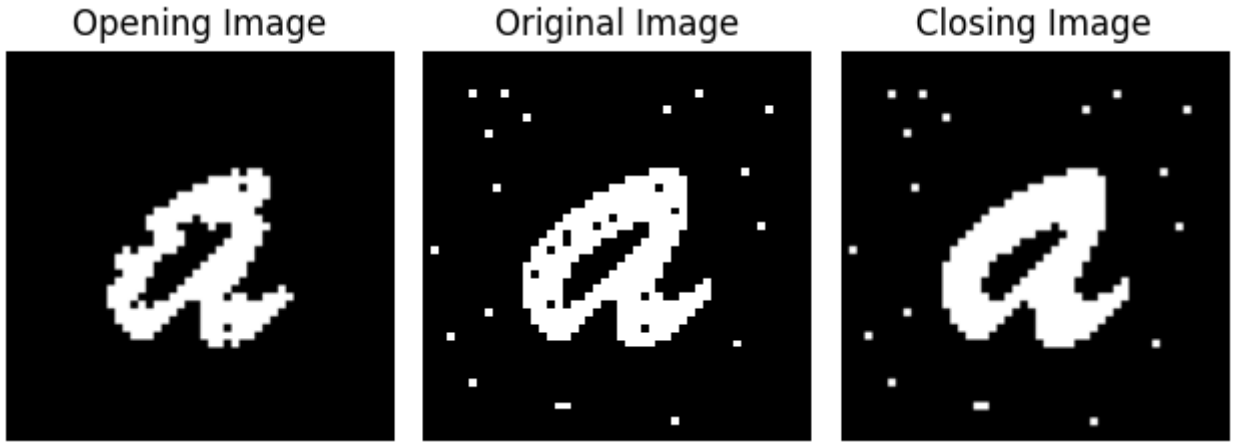
```
eroded_image = cv2.erode(image, kernel, iterations=1)
dilated_image = cv2.dilate(image, kernel, iterations=1)
```



OpenCV / Opening and Closing morphological transforms

Acquisition

Pre Processing



kernel

0	1	0
1	1	1
0	1	0

Opening

Erosion then **Dilation**

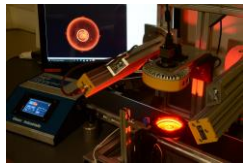
Removing small objects, in the background

Closing

Dilation then **Erosion**

Filling in small holes in the foreground

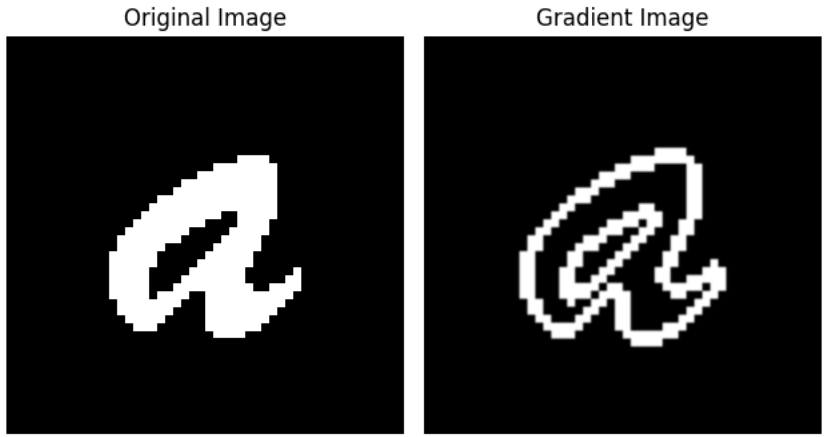
```
opening_image = cv2.morphologyEx(image, cv2.MORPH_OPEN, kernel)
closing_image = cv2.morphologyEx(image, cv2.MORPH_CLOSE, kernel)
```



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)

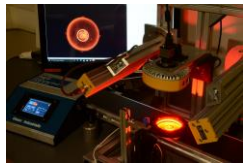


Image processing with OpenCV

OpenCV / Blur and mean

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

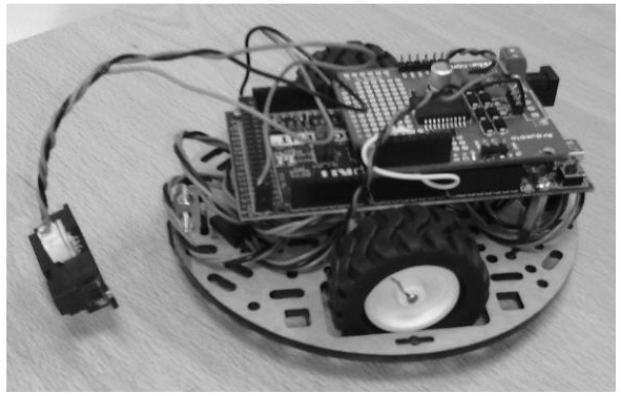
Acquisition

Pre Processing

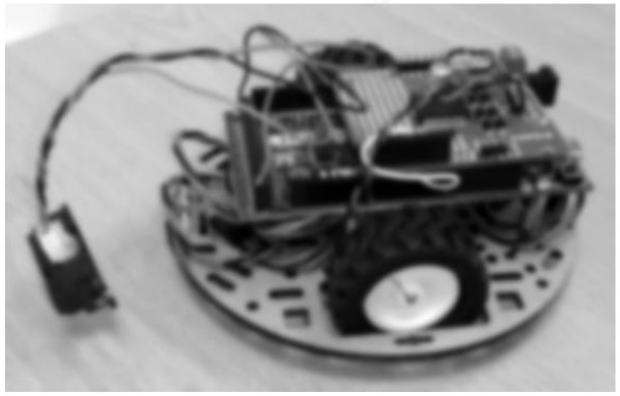
```
kernel_size = (N,M)

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

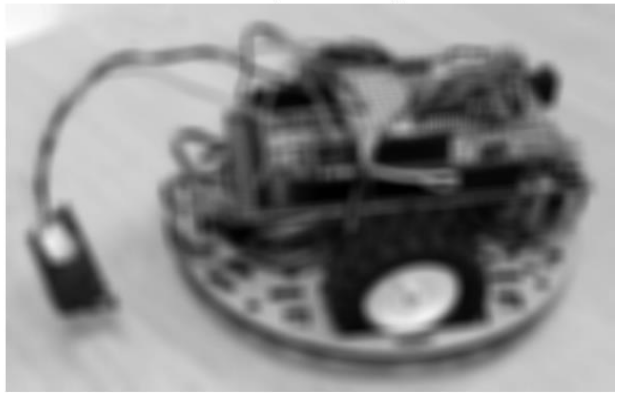
Original Image



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

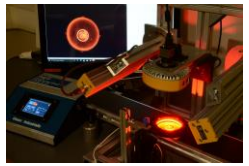


Image processing

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

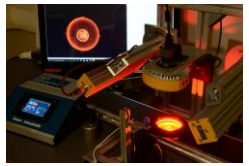
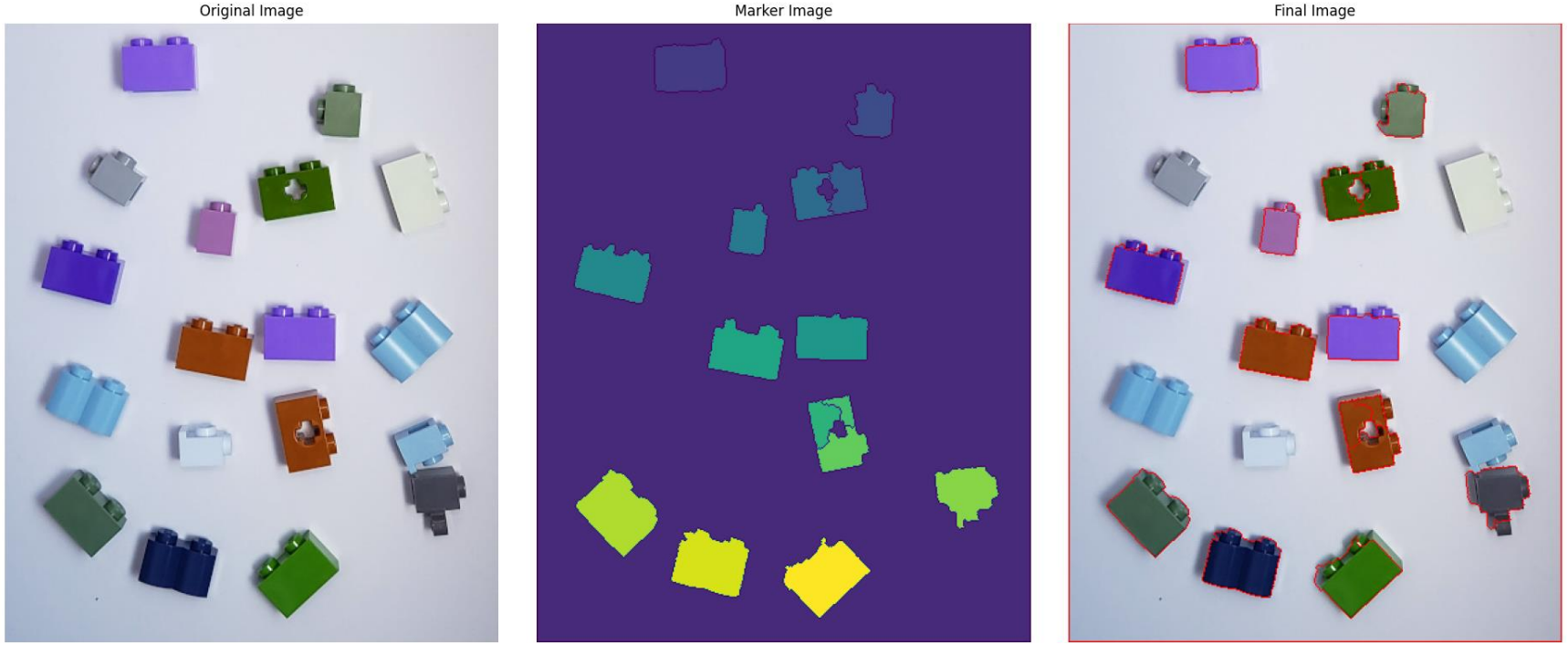


Image processing

Segmentation and Feature Extraction

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

- Acquisition
- Pre Processing
- Segmentation
- Feature Extraction



Watershed method

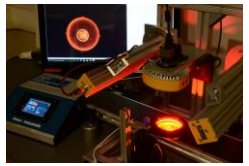


Image processing

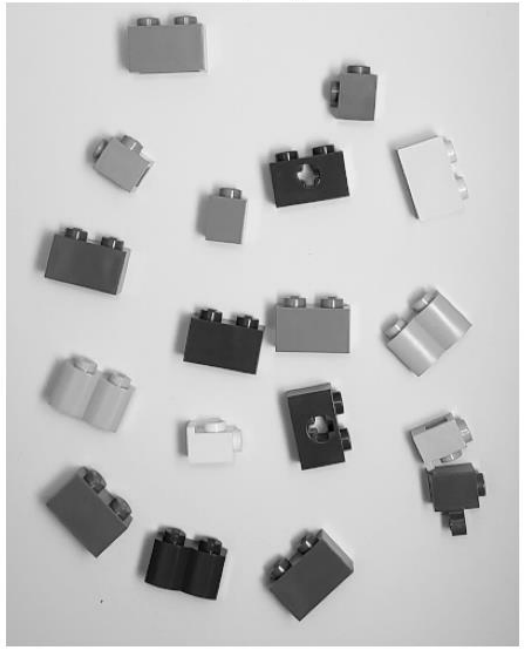
Segmentation and Feature Extraction

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

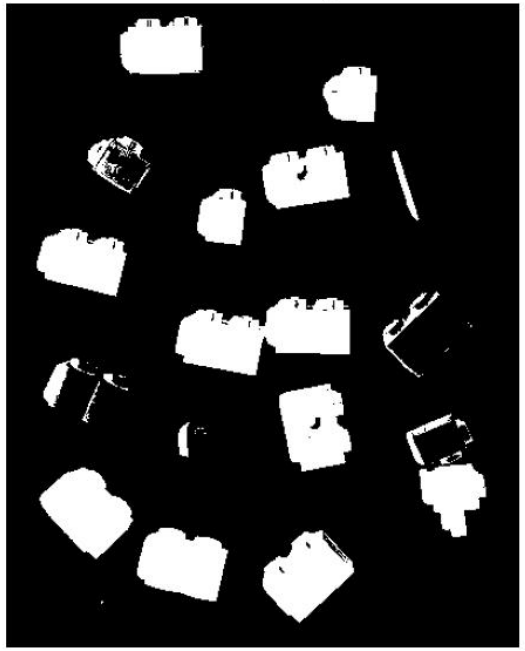
- Acquisition
- Pre Processing
- Segmentation
- Feature Extraction

Original RGB Image

Gray Image



Threshold Image



Watershed method / First step - Thresholding

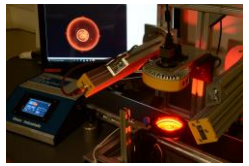


Image processing

Segmentation and Feature Extraction

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

- Acquisition
- Pre Processing
- Segmentation
- Feature Extraction

Watershed method / Second step – Sure BG/FG image

Watershed method / Third step – Labelling

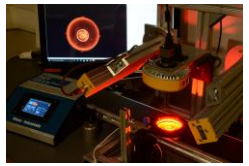
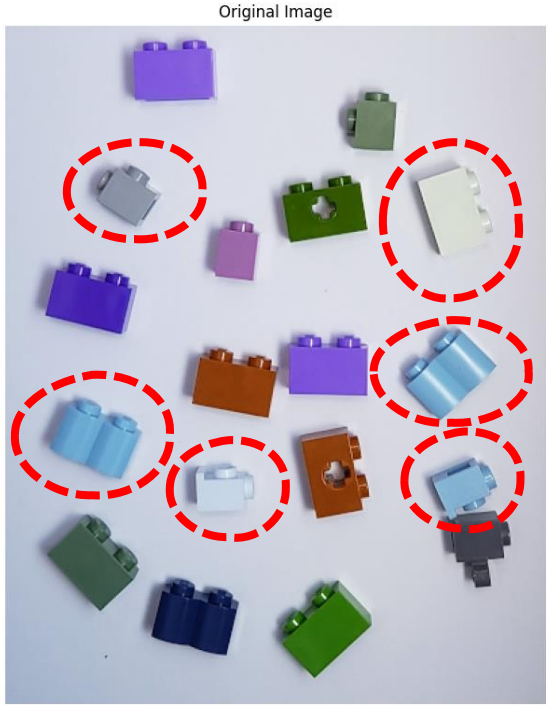


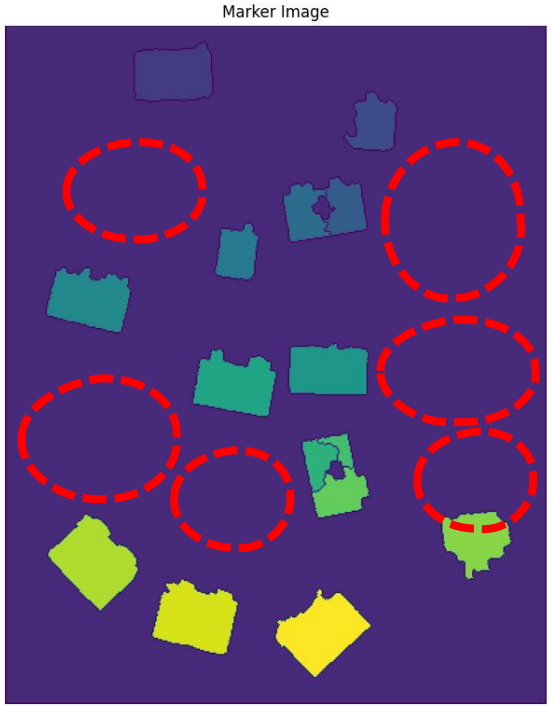
Image processing

Segmentation and Feature Extraction

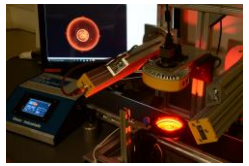
INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

- Acquisition
- Pre Processing
- Segmentation
- Feature Extraction





Watershed method



Fourier Transform and filtering

- Acquisition
- Pre Processing
- Segmentation
- Feature Extraction