Institut d'Optique Graduate School
Interfaçage Numérique

INTERFAÇAGE NUMÉRIQUE

Travaux Pratiques

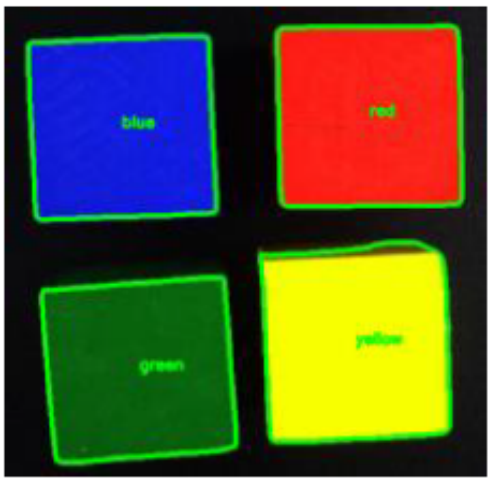
Semestre 6

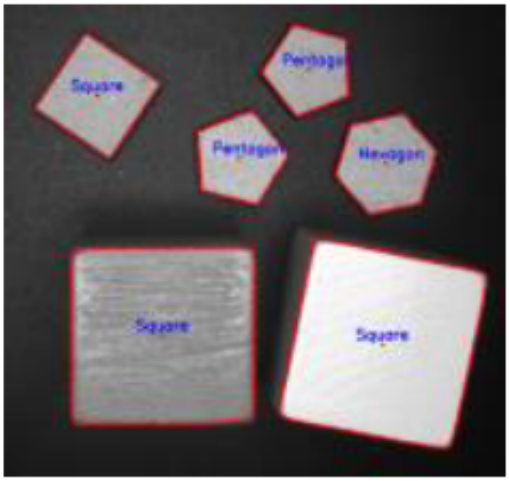
Vision Industrielle

De la physique de la chaîne d'acquisition à l'exploitation logicielle

---

4 séances

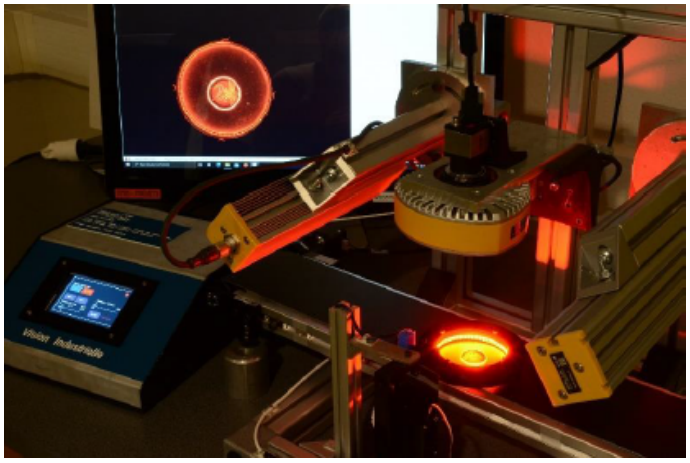




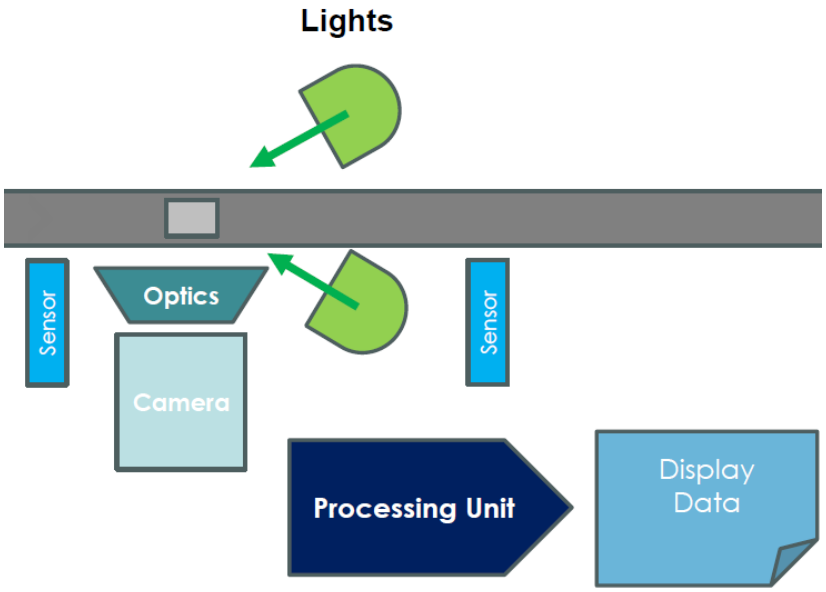
Ce sujet est disponible au format électronique sur le site du LEnsE - https ://lense.institutoptique.fr/ dans la rubrique Année / Première Année / Interfaçage Numérique S6 / Bloc 2 Vision Industrielle.

Vision Industrielle

La **vision industrielle** est une technologie qui permet à des machines d'**analyser automatiquement des scènes** pour **contrôler**, **guider** ou **inspecter** des objets sur des processus de production. Elle repose sur l'utilisation de **caméras**, d'**optique**, d'**éclairages** spécifiques (ou contraints), de **capteurs** et d'algorithmes de **traitement d'image**.



Elle a pour but de **prendre des décisions automatiques** (ou aider l'être humain dans sa prise de décision) vis-à-vis d'un (ou plusieurs) objet(s) dans une scène spécifique : détecter des défauts ou des irrégularités, compter ou trier..., en rejetant ou validant automatiquement des produits, tout en assurant une constance de la qualité et de la répétabilité des opérations.



Ce sujet a été co-conçu par une équipe d'étudiant·es - Joséphine BECHU, Justine GABRIEL et Paul CHENEAU - lors d'un projet DEPhI de 2A - 2025-2026 - et l'équipe pédagogique d'Interfaçage Numérique.

Acquis d'Apprentissages Visés (AAV)

À la fin de cette série de 4 séances, les étudiant·es seront capable de :

— Décomposer et paramétrer une chaîne de vision industrielle complète (du capteur au traitement de l'image),
— Comprendre les compromis physiques et numériques de chaque maillon,
— Réaliser un prototype d'inspection ou de mesure simple (tri d'objets),
— Justifier leurs choix de configuration (résolution, focale, éclairage...)

Ressources

Un tutoriel sur les bases d'OpenCV est disponible à l'adresse suivante :
https://iogs-lense-training.github.io/image-processing/

Un **kit d'images** est disponible sur le site du LEnsE dans la rubrique *Année / Première Année / Interfaçage Numérique S6 / Bloc 2 Caméra, Images et Interfaces / Images et OpenCV / Kit d'images*.

Des **fichiers de fonctions** sont disponibles sur le site du LEnsE dans la rubrique *Année / Première Année / Interfaçage Numérique S6 / Bloc 2 Caméra, Images et Interfaces / Images et OpenCV / Répertoire vers codes à tester*.

Quelques **exemples** et explications sur les différents pré-traitements d'images est disponible sur le site du LEnsE dans la rubrique *Année / Première Année / Interfaçage Numérique S6 / Bloc 2 Caméra, Images et Interfaces / Images et OpenCV / Image Processing with OpenCV*.

Déroulement

Les sujets **TP1** et **TP2** se font en binôme et sont interchangeables (4 binômes commenceront par la TP1 lors de la première séance et les 4 autres binômes commenceront par le TP2).

Les sujets **TP3** et **TP4** se font par groupe de 4 étudiant·es.

TP1 - Banc de vision industrielle

Le **TP1** se fera sur un banc de vision industrielle simple incluant une caméra, un objectif (focale : xx mm), un éclairage Effilux Ring-RGB et un ordinateur.

- [20'] Prendre en main l'interface - caméra (temps d'intégration, histogramme) + éclairage RGB
- [20'] Tester les outils de base proposés dans l'interface (coupe, lissage, seuillage)
- [20'] Vérifier l'uniformité de l'éclairage sur objet uniforme - coupe dans l'image
- [30'] Valider la linéarité de la caméra sur objet uniforme - pour différents temps d'intégration (en blanc et R/G/B)
- [60'] Contrôler le champ de vision du système et sa résolution spatiale
  - placer une règle pour mesurer le FOV (Field of View)
  - placer une mire pour mesurer la plus petite taille résoluble
  - Modifier la profondeur binaire (8, 10, 12 bits) et voir l'impact sur la résolution spatiale / sur la qualité de l'image
  - Revenir aux caractéristiques de l'objectif optique
- [30'] Contrôler le contraste du système en plaçant des mires en fonction du temps d'intégration, de l'éclairage
- [60'] Analyser l'impact des propriétés des objets et des sources sur la valeur mesurée par la caméra
  - Étudier les réflectances des cubes et le spectre des sources (fournis)
  - Comparer les niveaux de gris obtenus sous différents éclairages des différents objets mis à disposition (cubes, formes...)

TP2 - Manipulations de base sous OpenCV

Codes fournis (à modifier ?)

- [40'] Ouvrir une image - différence Gray/RGB
- [20'] Calculer l'histogramme d'une image
- [20'] Améliorer numériquement une image / contraste-luminosité
- [20'] Appliquer un seuillage sur une image
- [30'] Appliquer une érosion et une dilatation sur une image
- [20'] Appliquer une ouverture et une fermeture sur une image
- [20'] Appliquer un gradient sur une image
- [30'] Calculer la FFT d'une image
- [20'] Appliquer un filtre moyenneur sur une image
- [20'] Appliquer un filtre passe-haut - Sobel

TP3 - Manipulations avancées sous OpenCV / Limites du banc de vision

Sous OpenCV :
— [40'] Détecter des formes
— [40'] Détecter des couleurs
— [40'] Recolorisation (matrice de changement de base)
Segmentation de l'image ?

Sur le banc de VI :
— [40'] Calibration de la chaîne sur les cubes de couleurs (couleur)
— [40'] Calibration de la chaîne sur des formes colorées
— [40'] Calibration de la chaîne pour des mesures de distance

TP4 - Détection d'objets - Contrôle qualité

Le but est à partir d'une série d'images (1 en RGB ou 3 en niveau de gris sous éclairage particulier) de détecter le nombre d'objets d'une forme et d'une couleur particulière et de valider si les pièces requises pour un assemblage sont bien présentes (simuler le packaging d'un produit fini - 6 pièces de forme et de couleurs distinctes par exemple).

La validation du packaging final devra se faire sans l'intervention de l'être humain mais un affichage propre des couleurs devra être fait pour le contrôle par le/la manipulateur·trice.

Bonus : mesure de la conformité de la taille des pièces ?

Objectifs du bloc

L'objectif principal de ce bloc est de découvrir les **différents paramètres impactant la qualité d'une prise d'image par une caméra** dans un environnement industriel ainsi que la **manipulation d'images numériques**, à l'aide de la bibliothèque **OpenCV**.

On s'intéressera à l'impact du **temps d'intégration**, de la **résolution** de la caméra et de l'**éclairage** (couleur et intensité) de la scène et des objets sur l'image résultante.

Cette étude sera complétée par un TP plus détaillé sur les bruits associés à l'utilisation des caméras CMOS en 2ème année du cycle ingénieur à Palaiseau.

Des cours et des projets autour du traitement d'images sont également proposés dans les prochaines années de formation, quelque soit le site. Ces deux séances sont une introduction à ces modules plus avancés.

*Le bloc **Images et OpenCV** (facultatif) de ce module permet d'aller également plus loin dans le pré-traitement des images et leur manipulation.*

Séance 1 / Banc de vision industrielle

Ce bloc de travaux pratiques utilise un **banc de vision industrielle** avec une lampe de type Effi-Ring RGB, une caméra Basler et une interface développée en **Python** (*PyQt6*) et qui utilise des fonctionnalités de la bibliothèque **OpenCV**.

Les documentations de la caméra et de l'éclairage sont disponibles aux adresses suivantes :

— Basler **a2A 1920 - 160ucBAS** : https://docs.baslerweb.com/a2a1920-160ucbas#specifications
— **Effi-Ring** : https://www.effilux.com/fr/produits/annulaire/effi-ring

Interfaçage Numérique

Travaux Pratiques

Semestre 6

Ressources

Bloc Vision Industrielle

Liste des ressources

— Image Processing / Key concepts

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

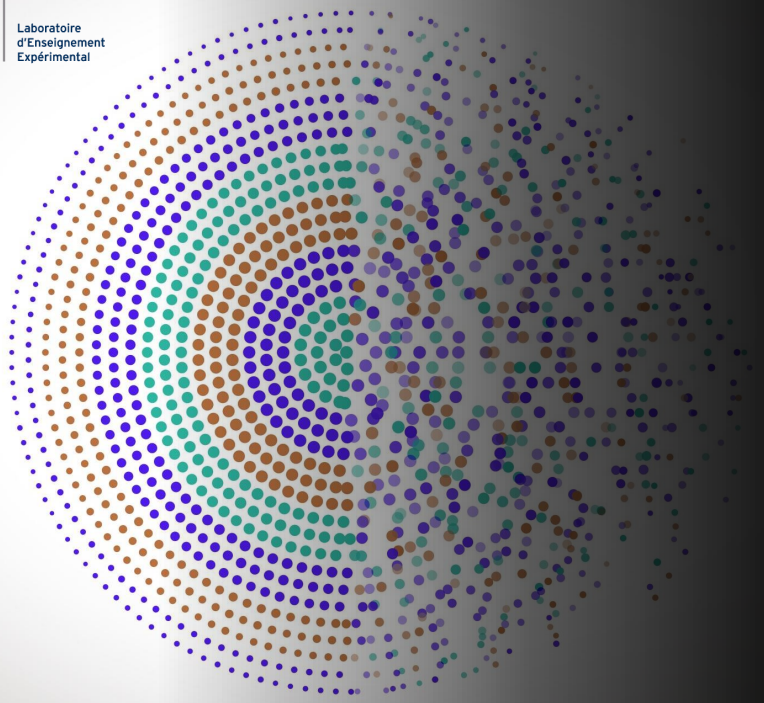
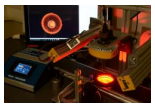


Image processing with OpenCV

Institut d'Optique – Engineers Training

Semester 6 – Digital Interface

Julien VILLEMEJANE



Goal of processing an image

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

Image from the camera

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

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

- Homogeneous zones
- Transition zones

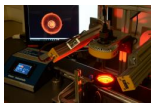


Image processing

Steps for processing an image

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…	

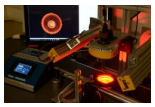


Image processing with OpenCV

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/

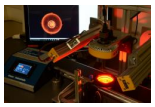
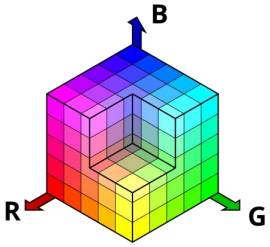


Image processing with OpenCV

Digital Images / Color Spaces

RGB

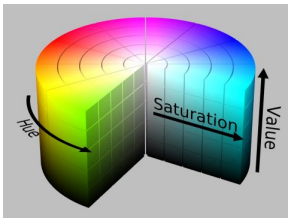
Used primarily in **electronic displays** like computer screens, cameras, and scanners. The combination of these three primary colors at various intensities can produce any color.



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

Images Source : Wikipedia

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

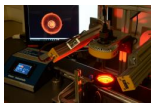


Image processing with OpenCV

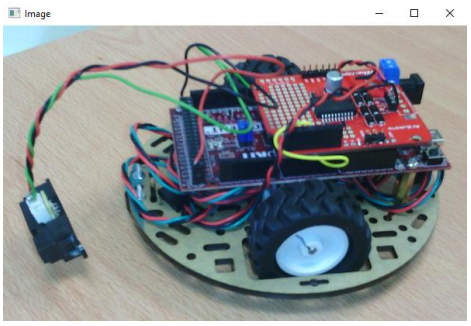
OpenCV / Open and display an Image

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)
```

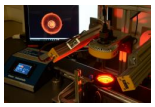
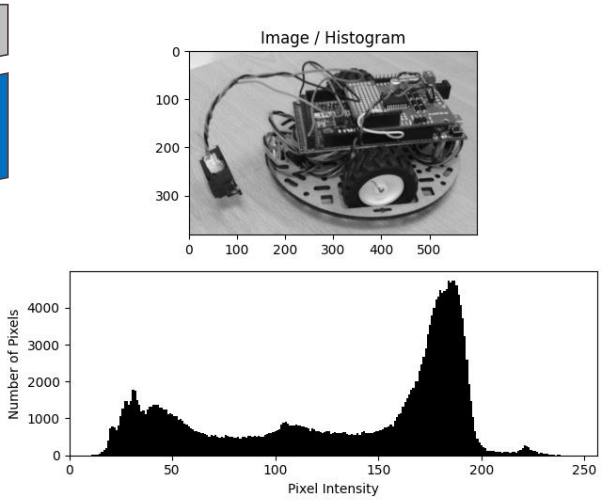


Image processing with OpenCV

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])
```

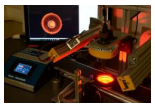
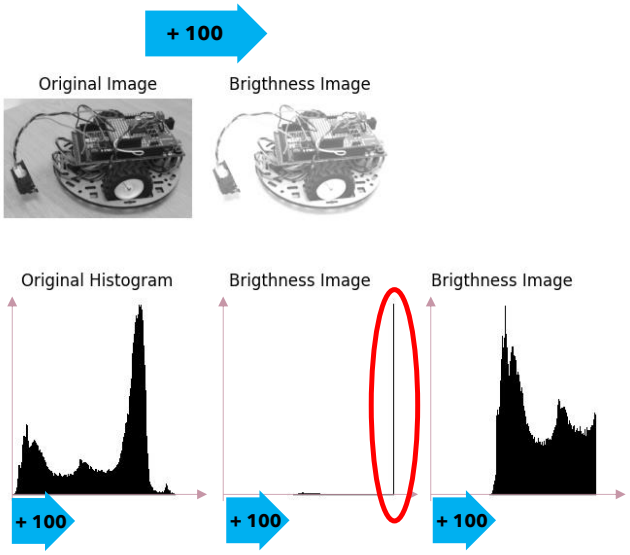


Image processing with OpenCV

OpenCV / Contrast and Brightness

Acquisition

Pre Processing



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

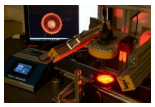
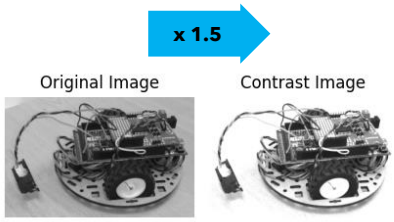


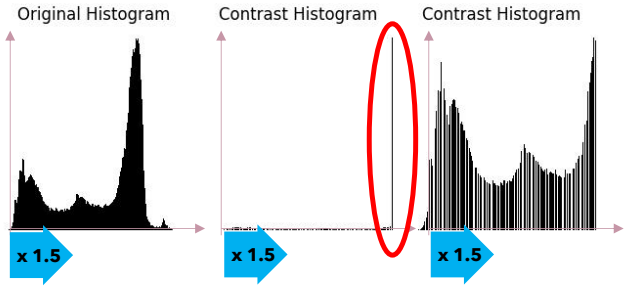
Image processing with OpenCV

OpenCV / Contrast and Brightness

Acquisition

Pre Processing





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

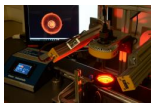


Image processing with OpenCV

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**

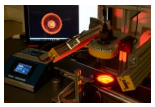
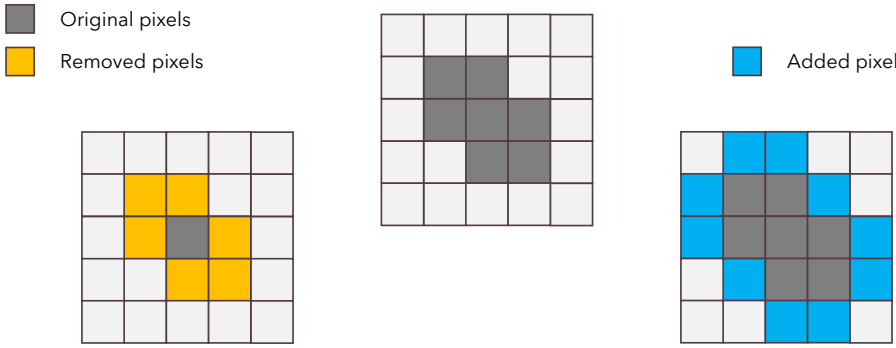


Image processing with OpenCV

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

https://www.youtube.com/watch?v=fmyE7DialYQ

Dilation

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

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

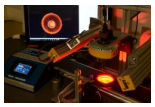
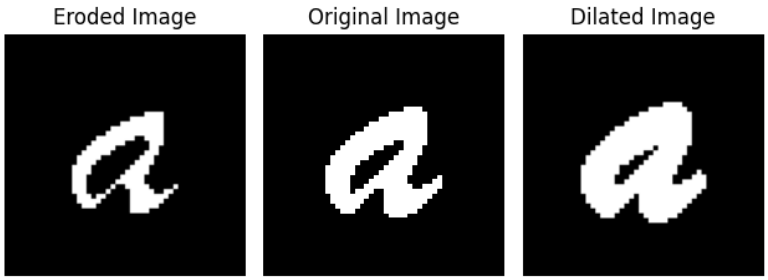


Image processing with OpenCV

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)
```

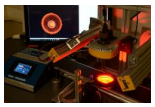
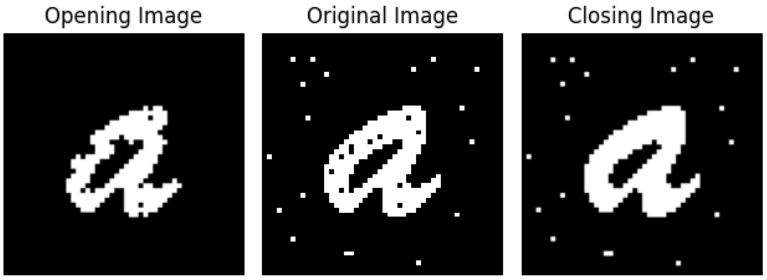


Image processing with OpenCV

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)
```

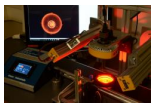
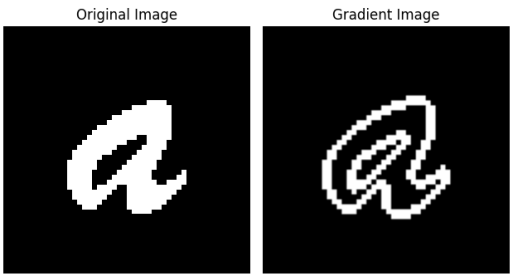


Image processing with OpenCV

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)

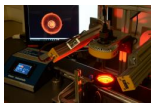


Image processing with OpenCV

OpenCV / Blur and mean

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

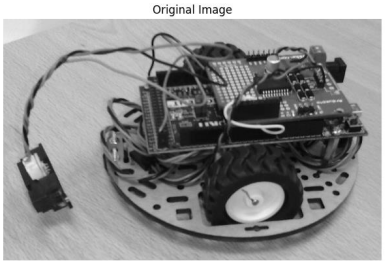
Acquisition

Pre Processing

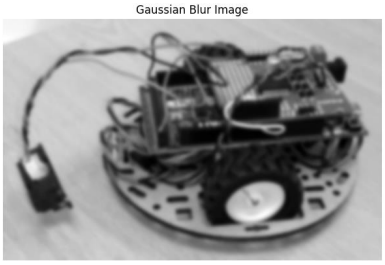
```
kernel_size = (N,M)

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

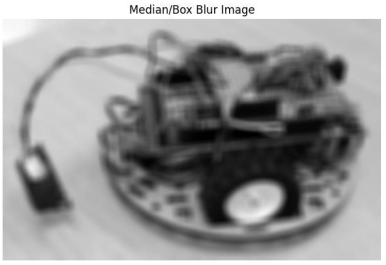
Original Image



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

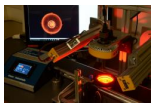


Image processing

Fourier Transform and filtering

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

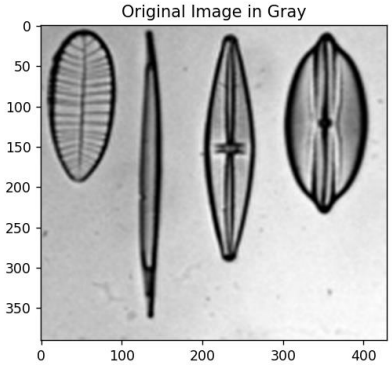
Acquisition

Pre Processing

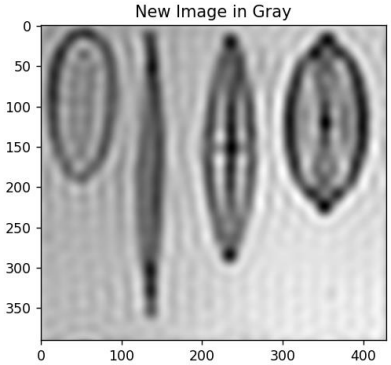
Segmentation

Feature Extraction

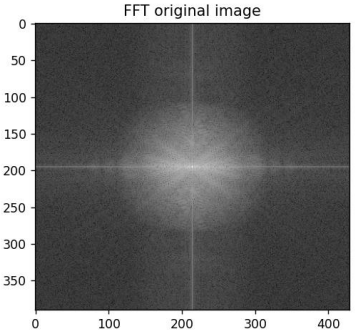
Original Image in Gray



New Image in Gray



FFT original image



Masked FFT