



## Master 1 Informatique

## Traitement d'image – 2024

**TD2 – Détection du model dans l'image et de contours**

**Le but** de ce TD est de manipuler et traiter une image à partir d'une bibliothèque OpenCV en Python. Nous allons appliquer les techniques suivantes sur les images :

- Opérations morphologiques sur l'image.
- Détection du model dans l'image.
- Calcule d'un gradient d'image.
- Détection des coins dans une image.

**Exercice 1. Operateurs de morphologie mathématique**

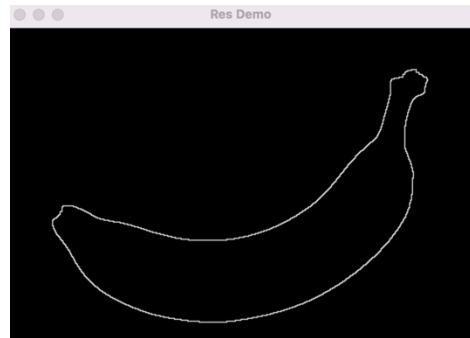
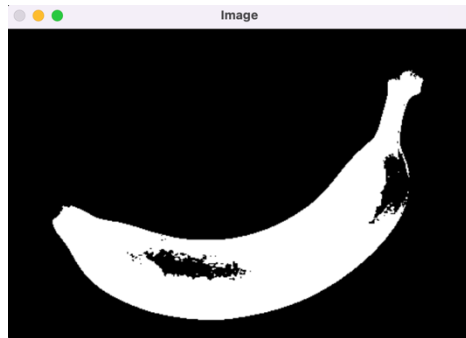
Utilisez les images [morph\\_1.pgm](#), [morph\\_2.pgm](#), [morph\\_3.pgm](#)

- 1) Prenez une image, seuillez cette image en testant plusieurs valeurs de seuils. Modifiez la fonction de seuillage afin que le fond de l'image soit en blanc et que les pixels des objets soient en noir. Comparer l'image seuillée avec au moins 3 valeurs différentes et en déduire le seuil le plus pertinent.
- 2) A partir de l'image seuillée la plus intéressante, écrire la fonction `erosion()`, qui va permettre de supprimer les points objets isolées. Exécuter cette fonction en utilisant l'image binaire obtenue précédemment avec le seuil le plus intéressant.
- 3) A partir de l'image seuillée la plus intéressante, écrire la fonction `dilatation()`, qui va permettre de boucher des petits trous isolés dans les objets de l'image. Exécuter cette fonction en utilisant l'image binaire obtenue précédemment avec le seuil le plus intéressant.

Utilisez les images [banane.jpg](#), [tel.jpg](#)

- 4) Utilisez des fonctions `erode()` et `dilate()` d'OpenCV pour calculer le gradient morphologique. Testez-le sur différentes images.  
Le but de cet exercice est de développer une approche permettant de visualiser les contours d'une image. A partir de l'image érodée et de l'image dilatée, visualiser les contours des objets contenus dans l'image :
  - Si les deux pixels (de l'image érodée et de l'image dilatée) appartiennent au fond, alors le pixel correspondant de l'image de sortie appartiendra au fond (255).

- Si les deux pixels (de l'image érodée et de l'image dilatée) appartiennent à l'objet, alors le pixel correspondant de l'image de sortie appartiendra au fond (255).
  - Si le pixel de l'image dilatée appartient à l'objet et que le pixel de l'image érodée appartient au fond, alors le pixel correspondant de l'image de sortie appartiendra au contour (0).
- 5) Comment corriger des erreurs de binarisation dans l'image ci-dessous et obtenir le contour d'objet « propre » ?



Pour plus d'information sur les opérations morphologique, regardez la documentation de OpenCV :

[https://opencv24-python-tutorials.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_morphological\\_ops/py\\_morphological\\_ops.html](https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html)

## Exercice 2. – Où est Charlie ?

Utilisez la technique de détection du modèle dans l'image pour retrouver Charlie ([template.jpg](#)) dans l'image ([source.jpg](#)). Implémentez la fonction de la corrélation croisée normalisée.

Testez des autres modèles ([template2.jpeg](#) et [template3.jpg](#)) et des autres images de sources ([source2.jpeg](#) et [source3.jpg](#)). Que constatez-vous ?

### **Indice exercice 2 :**

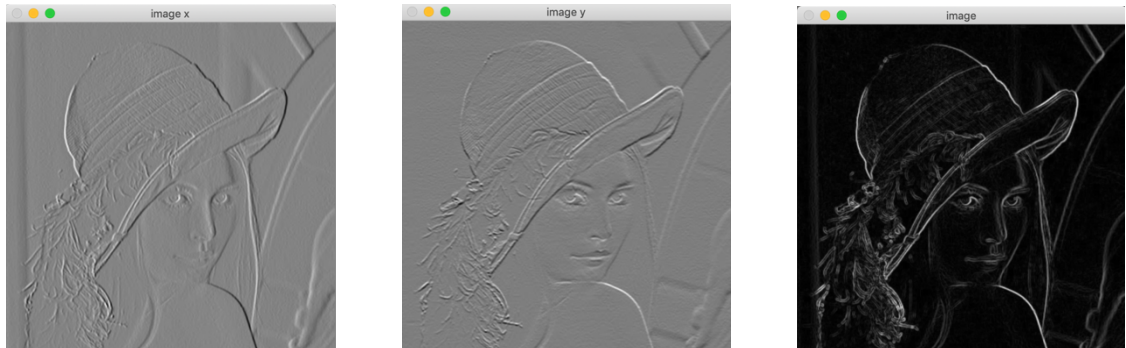
Regardez la fonction `matchTemplate()` dans le documentation de OpenCV [https://docs.opencv.org/4.x/d4/dc6/tutorial\\_py\\_template\\_matching.html](https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html)



### Exercice 3. – Gradient d'image

Calculez le gradient d'une image en direction x et y et détectez des contours en utilisant le détecteur de Sobel.

**Indice** : vous pouvez utiliser la fonction `Sobel()` du OpenCV.



### Exercice 4. – Détection des coins Harris

Utilisez l'image `chessboard00.png`.

- 1) Détectez des coins dans l'image `chessboard00.png` en utilisant la fonction implémentée dans OpneCV `cornerHarris()`.

Vous pouvez utiliser le tutoriel du OpenCV :

[https://docs.opencv.org/3.4/d4/d7d/tutorial\\_harris\\_detector.html](https://docs.opencv.org/3.4/d4/d7d/tutorial_harris_detector.html)

A quoi correspond la variable `thresh` ? Changez le code donné dans le tutoriel pour qu'on puisse définir par nous-même la valeur de la variable `thresh`. Quelle valeur devons-nous donner à cette variable pour retrouver tous les coins ?

### Exercice 5. – Transformation du photomaton

On part d'un tableau  $n \times n$ , avec  $n$  pair, chaque élément du tableau représente un pixel. À partir de cette image on calcule une nouvelle image en déplaçant chaque pixel selon une transformation, appelée transformation du photomaton. On découpe l'image de départ selon des petits carrés de taille  $2 \times 2$ . Chaque petit carré est donc composé de quatre pixels. On envoie chacun de ces pixels à quatre endroits différents de la nouvelle image : le pixel en haut à gauche reste dans une zone en haut à gauche, le pixel en haut à droite du petit carré, est envoyé dans une zone en haut à droite de la nouvelle image, etc. (voir illustration).

**Algorithme de transformation du photomaton :**

Pour chaque couple  $(i, j)$ , on calcule son image  $(i', j')$  par la transformation du photomaton selon les formules suivantes :

★ Si l'indice  $k$  est impair alors  $k' = \frac{k+1}{2}$

★ Si l'indice  $k$  est pair alors  $k' = \frac{k+n}{2}$ ,

Où  $k$  est soit  $i$ , soit  $j$ .

A	B	<u>a</u>	<u>b</u>				
C	D	<u>c</u>	<u>d</u>				

A	a			B	b		
C	c			D	d		

- 1) Programmez la transformation du photomaton d'une image de taille 256x256.
- 2) Faites-en sorte que votre programme fonctionne avec n'importe quelle image qui a le nombre pair de lignes et de colonnes.

**Référence :** <https://images.math.cnrs.fr/Mona-Lisa-au-photomaton.html>