

Traitement d'images

TP 3 : morphologie mathématique-

Application

Dans ce TP, nous allons pratiquer des transformations morphologiques sur des images. Fondamentalement, il existe deux transformations morphologiques de base : la dilatation et l'érosion. Elles sont présentes dans le traitement des images dans différentes applications. Elles sont utilisées pour supprimer le bruit ou pour trouver les bosses ou les trous dans les images. En outre, ces opérations peuvent également être utilisées pour calculer les gradients des images. En outre, une fois que nous avons appris les deux opérations morphologiques de base, nous pouvons les combiner pour créer des opérations supplémentaires telles que l'ouverture et la fermeture.

Préparation des images et prétraitement

1. Ouvrir ou acquérir une image couleur. Transformer cette image en image niveaux de gris.
2. Binariser l'image précédente de telle sorte à obtenir des objets en blanc et un fond en noir.

Opération d'érosion

Principes de l'érosion :

- L'érosion fait disparaître les limites de l'objet au premier plan
- Utilisé pour diminuer les caractéristiques d'une image.

Fonctionnement de l'érosion :

- Un noyau (une matrice de taille impaire (3,5,7)) est convolué avec l'image.
 - Un pixel de l'image originale (1 ou 0) ne sera considéré comme 1 que si tous les pixels sous le noyau sont 1, sinon il est érodé (ramené à zéro).
 - Ainsi, tous les pixels proches de la limite seront éliminés en fonction de la taille du noyau.
 - Ainsi, l'épaisseur ou la taille de l'objet de premier plan diminue ou simplement la région blanche diminue dans l'image.
3. Appliquer une érosion sur l'image binaire en utilisant comme élément structurant un carré de taille 3×3 pixels. Commenter les résultats.
 4. Appliquer une érosion sur l'image binaire en utilisant comme élément structurant un carré de taille 5×5 pixels. Commenter et comparer les résultats.

5. Appliquer deux érosions consécutivement sur l'image binaire en utilisant comme élément structurant un carré de taille 3×3 pixels. Commenter et comparer les résultats.
6. Appliquer une érosion sur l'image binaire en utilisant comme élément structurant un disque de taille 5×5 pixels. Commenter et comparer les résultats.

Dilatation

Principes de la dilatation :

- Augmente la surface de l'objet
- Utilisée pour accentuer les caractéristiques

Fonctionnement de la dilatation :

- Un noyau (une matrice de taille impaire (3,5,7)) est convolué avec l'image.
- Un élément de pixel de l'image originale est "1" si au moins un pixel sous le noyau est "1".
- Il augmente la zone blanche de l'image ou la taille de l'objet au premier plan.

7. Appliquer une dilatation sur l'image binaire en utilisant comme élément structurant un carré de taille 3×3 pixels. Commenter les résultats.
8. Appliquer une dilatation sur l'image binaire en utilisant comme élément structurant un carré de taille 5×5 pixels. Commenter et comparer les résultats.
9. Appliquer deux dilations consécutivement sur l'image binaire en utilisant comme élément structurant un carré de taille 3×3 pixels. Commenter et comparer les résultats.
10. Appliquer une dilatation sur l'image binaire en utilisant comme élément structurant un disque de taille 5×5 pixels. Commenter et comparer les résultats.

Exemple de code Python montrant l'érosion et la dilatation sur une image

```
# creation du noyau (ici noyau carré de taille 5*5)
kernel = np.ones((5, 5), np.uint8)

# Le premier argument est l'image originale
# le second argument est le
# le troisième est le nombre d'itérations qui détermine
combien l'érosion # ou la dilatation est appliquée sur une
image donnée
img_erosion = cv2.erode(img, kernel, iterations=1)
img_dilation = cv2.dilate(img, kernel, iterations=1)

cv2.imshow('Input', img)
cv2.imshow('Erosion', img_erosion)
cv2.imshow('Dilation', img_dilation)
```

Ouverture et fermeture

Une ouverture est une érosion suivie d'une dilatation et une fermeture est une dilatation suivie

d'une érosion.

Ouverture

En utilisant les fonctions définies précédemment, réaliser une ouverture sur l'image binarisée. Comparer et commenter les résultats.

Réaliser la même opération avec la fonction `cv2.morphologyEx()`

```
# noyau (élément structurant)
kernel = np.ones((5, 5), np.uint8)

# definition de l'ouverture sur l'image
opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)

# The mask and opening operation
# is shown in the window
cv2.imshow('original image', img)
cv2.imshow('Opening', opening)
```

Que se passe-t-il lorsque cette opération est répétée ou lorsque la taille ou la forme de l'élément structurant change ?

Fermeture

En utilisant les fonctions définies précédemment, réaliser une fermeture sur l'image binarisée. Comparer et commenter les résultats.

Réaliser la même opération avec la fonction `cv2.morphologyEx()` en mettant le bon flag.

Que se passe-t-il lorsque cette opération est répétée ou lorsque la taille ou la forme de l'élément structurant change ?

Exemples d'éléments structurants avec Python

```
# Creation de noyaux 3x3 de différentes formes
kernel_1 = np.array ( [ [0,1,0], [1,1,1], [0,1,0] ] )
kernel_2 = np.array ( [ [0,0,1], [0,0,1], [0,0,1] ] )
kernel_3 = np.array ( [ [1,1,1], [0,0,0], [0,0,0] ] )

kernel_1 = kernel_1.astype('uint8')
kernel_2 = kernel_2.astype('uint8')
kernel_3 = kernel_3.astype('uint8')
```

Application

Analyse qualitative de globules rouges à partir de l'observation microscopique d'échantillons sanguins. Cette analyse pourrait se faire à l'aide d'un traitement morphologique des images issues d'un microscope :

- Prendre l'image globules.tif
- En utilisant des opérateurs morphologiques, imaginer et appliquer un algorithme permettant de segmenter les globules rouges de façon précise : l'algorithme devra éliminer les plaquettes (petites cellules foncées), les globules rouges qui sont coupés (qui touchent le bord), les globules rouges qui sont superposés.

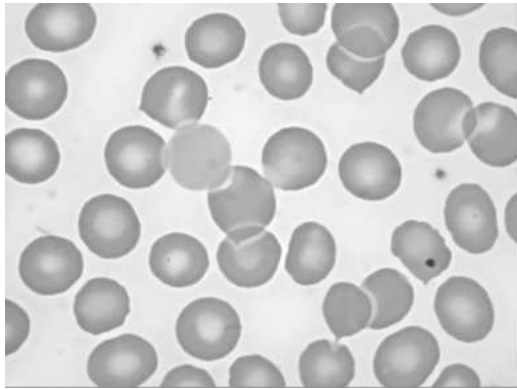


Image d'origine

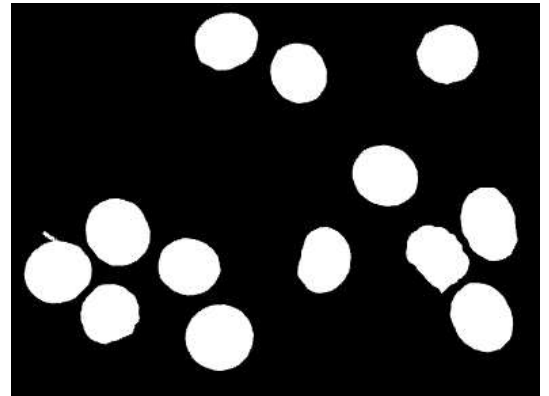


Image souhaitée

- De même imaginer un algorithme pour déterminer les inclusions intra-globules.

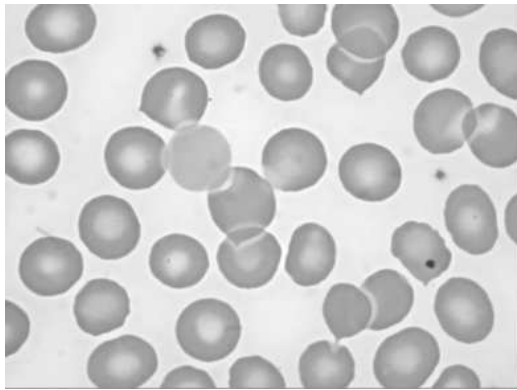


Image d'origine

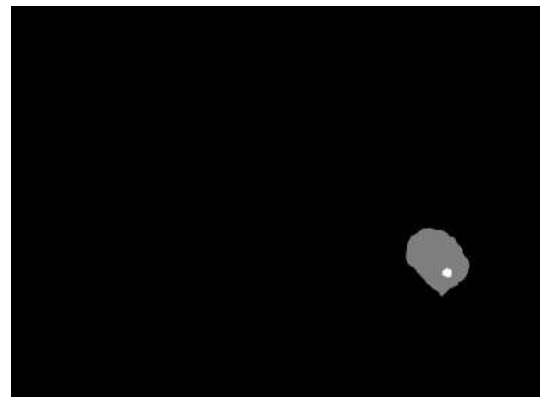


Image souhaitée