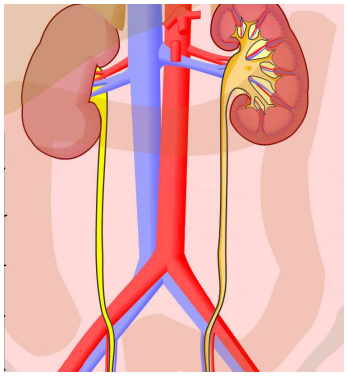
Comment améliorer le système Endo-Cobot ?

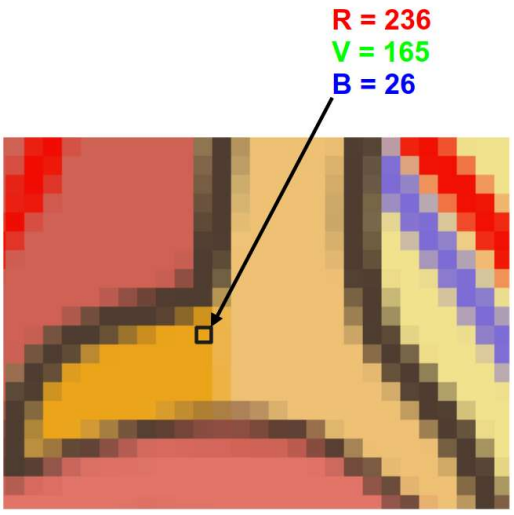
Connaissance associée : codage et traitement de l'information



Nous avons vu dans la première activité que l'image endoscopique des artères iliaques devait être annotée sur écran tactile. Pour tenter d'automatiser l'annotation, la détection de contours peut être envisagée.

Une des premières étapes est alors de convertir l'image en niveau de gris.

1) Comment passer d'une image couleur à une image en niveau de gris



Une image couleur est codée sur 3 octets (0 à 255) au format RVB.

Lancer le logiciel Gimp et ouvrir l'image **test.bmp.**

Zoomer (**Affichage** → **Zoom**) sur l'image au voisinage des zones rouge et bleu pour voir apparaître les pixels.

Aller sur **Fenêtres** → **Fenêtres ancrables** → **Pointeur** et passer sur l'image avec le pointeur de la souris, les valeurs des pixels apparaissent.

Faire **Image** → **Mode** → **Niveaux de gris** et repasser sur l'image.

1. Que constate-t-on ?
2. Sauvegarder l'image en niveau de gris (**Fichier** → **Exporter sous**) en nommant le fichier **testNG.bmp**. Calculer le rapport de la taille des fichiers **test.bmp** et **testNG.bmp**. Le résultat est-il prévisible ? Pourquoi ? Pourquoi la valeur n'est-elle pas entière ?
3. Comment passer de trois valeurs **RVB** à une seule en niveau de gris ?

Pour réaliser cela, nous allons utiliser la bibliothèque **Pillow** du langage Python.

Ouvrir l'IDE Thonny et aller à **Outils** → **Ouvrir la console du système** et taper

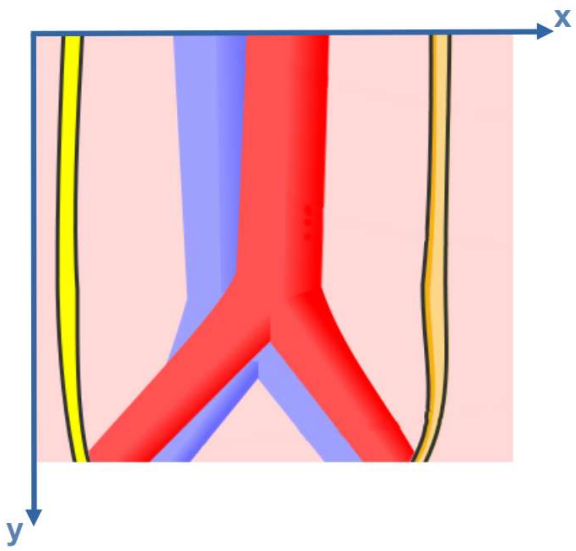
pip install --proxy=http://10.129.254.254:3128 pillow

4. Écrire la ligne de code qui permet de stocker, dans une variable **imgPil,** l'image **test.bmp**.

5. En s'aidant des activités faites sur les boucles, expliquer comment passer par tous les pixels de la première ligne par exemple

6. De même, expliquer comment passer par toutes les lignes de pixels

7. Sans mettre d'instructions, écrire les deux boucles avec les variables y, x et hauteur, largeur pour les dimensions de l'image



L'image sera manipulée sous forme de tableau **numpy**. Chaque valeur **RVB** d'un **pixel de coordonnées (x,y)** d'une image **img** est représentée par un élément [R, V, B] d'un tableau numpy :

npImg [y] [x] = [R, V, B] soit **npImg [y] [x] [0] = R**

npImg [y] [x] [1] = V

npImg [y] [x] [2] = B

Nous avons donc un tableau à 3 dimensions y, x et RVB

8. Comment récupère-t-on les dimensions du tableau numpy **npImg** ?

9. Les valeurs obtenues pour l'image test.bmp sont (322, 363, 3). À quoi correspond chacune de ces valeurs ?

10. Comment ne récupérer que la valeur 363 par exemple ?

11. Compléter le code ci-dessous qui permet de donner les dimensions de l'image :

```
print("L'image est composée de", npImg.          , " lignes et ",npImg.          ,"
colonnes.")
```

12. Écrire le code qui permet de calculer la variable **moyenne** qui doit être un entier.

Moyenne = int( )

Les 3 valeurs RVB d'un pixel de coordonnées (x,y) seront replacées par une seule valeur. Le tableau **npImgNG** utilisé pour l'image en niveau de gris n'aura donc que 2 dimensions.

npImg [y] [x] = [R, V, B] → **npImgNG [y] [x] = valeurNiveauGris**

13. Télécharger le fichier **niveauGris.py** sur l'espace Moodle STI2D-SIN, séquence Endo-Cobot dans le dossier « Fichiers », compléter le code des lignes 13 à 17 et exécuter le programme.

2) Comment détecter des contours ?

14. Dans GIMP, passer sur l'image avec le pointeur aux alentours des contours. Que constate-t-on au passage sur un contour ?

15. Comment peut-on alors détecter ces variations en parcourant les pixels ?

16. Nous allons utiliser les valeurs de niveaux de gris des pixels voisins d'un **pixel de coordonnées (x,y) du tableau NG selon deux directions (horizontale et verticale)**. Compléter la figure ci-dessus en fonction de x et y.

	NG[ ][ ] = nv1	
NG[ ][ ] = nv3	**NG[y][x] = nv**	**NG[ ][ ] = nv4**
	NG[ ][ ] = nv2	

17. Un calcul de norme va permette de connaître l'écart de niveau de gris entre les voisins selon les 2 directions :

$$norme = \sqrt{(nv2 - nv1)^2 + (nv4 - nv3)^2}$$

Écrire le code qui permet de calculer cette norme en utilisant la notation **NG[ ][ ].** La bibliothèque **math** sera utilisée.

Plus la norme calculée est grande et plus il y a de différences de niveaux de gris entre pixels voisins. Les contours correspondent aux zones où la norme est importante, au dessus d'un certain seuil.

Pour dessiner les contours, il est possible de créer une image de pixels noirs de même dimensions que l'image en niveau de gris et de remplacer les pixels noirs par des pixels blancs si

18. Télécharger le fichier **contours.py** sur l'espace Moodle STI2D-SIN, séquence Endo-Cobot dans le dossier « Fichiers », compléter le code des lignes 30 et 31. Exécuter le programme.

19. La valeur du seuil choisi est de 8 (*npImgContours = contours (npImgNG, 8)*). Tester d'autres valeurs et discuter du choix du seuil.

20. Tester le programme avec une image au choix.