

Mini-projet

→ Un logiciel (simplifié) de traitement d'image

L'objectif est de créer en Python un programme capable de réaliser un certain nombre de modification d'image à la demande de l'utilisateur.

1 Travail demandé

Fonctions du programme :

Le programme python de traitement d'image doit au moins pouvoir faire :

- **1 changement de couleur** (action de filtre)
L'utilisateur choisit un changement de couleur ou de teinte sur toute l'image et le programme affiche le résultat.
- **1 détournage**
L'utilisateur choisit la valeur du paramètre de seuil utilisé et le programme affiche le résultat.

En plus, on pourra imaginer :

- Possibilité de sauvegarder les nouvelles images créées.
- Action du filtre sur une zone précise de l'image.
- Plusieurs options de mode de détournage.
- 1 outil de déformation quelconque (cisaillement, zoom, « barbouillage », ...)
- 1 interface graphique (tkinter, ...)

Compétences visées : (/10 points)

	A	B	C	D
ANA	Décomposer un problème en sous-problèmes			
3	<i>Fonctions pertinentes. Corps de programme fonctionnel.</i>	<i>Présence d'au moins une fonction. Corps de programme comportant des idées d'algorithmes.</i>	<i>Pas de fonctions, mais les idées d'algorithmes en place dans le corps du programme.</i>	<i>Pas de fonctions. Pas de corps de programme réel.</i>
REA	Concevoir des solutions algorithmiques			
3	<i>Algorithmes précis et rigoureux qui répondent à la demande. Commentaires.</i>	<i>Algorithmes commentés, mais qui répondent partiellement à la demande.</i>	<i>Algorithmes qui répondent partiellement à la demande. Absence de commentaires.</i>	<i>Pas de solution algorithmique.</i>
COM	Coopérer au sein d'une équipe dans le cadre d'un projet			
3	<i>Travail d'équipe efficace. Bonne répartition du travail entre les élèves.</i>	<i>Travail d'équipe, mais sans répartition claire des tâches.</i>	<i>Travail ensemble, mais sans concertation.</i>	<i>Pas de travail d'équipe.</i>

+1 pour un programme fonctionnel

2 En option : Utilisation de PIL

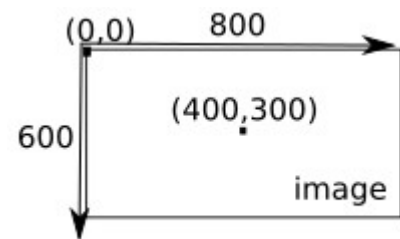
Pour tester quelque chose de différent par rapport au TP, nous utiliserons les fonctionnalités du module PIL directement, sans passer par une conversion *GrayMap*.

Doc de PIL : <https://he-arc.github.io/livre-python/pillow/index.html>

Voici quelques exemples permettant de manipuler des images numériques :

```
1 from PIL import Image
2
3
4 # Ouvrir et afficher l'image
5 img = Image.open("cn.jpg")
6 img.show()
7
8
9 # Dimensions
10 largeur, hauteur = img.size
11 print("largeur : ", largeur)
12 print("hauteur : ", hauteur)
13
14 # Obtenir la couleur (R, G, B) d'un pixel
15 R, G, B = img.getpixel((0, 0)) # pix de coord (0, 0)
16 print(R, G, B)
17
18
19 # Modifier la couleur (R, G, B) d'un pixel
20 img.putpixel((0, 0), (255, 0, 0))
21 # le pixel de coord (0, 0) passe en rouge
22
```

On notera que le pixel (0, 0) est situé en haut à gauche :



(exemple pour une image 300x400)

A l'aide de boucle **for**, on peut ainsi parcourir l'image pour modifier les valeurs du tuple (R, G, B).

On peut créer et sauvegarder une image :

```
1 from PIL import Image
2 largeur=600
3 hauteur=600
4 img =Image.new('RGB', (largeur,hauteur))
5 for y in range(hauteur):
6     for x in range(largeur):
7         img.putpixel((x,y),(255,0,0))
8
9 img.show()
10 img.save("CarreRouge.png", "PNG")
```