

DS n°3

La calculatrice est interdite.



Dans ce devoir, on va s'intéresser à la manipulation d'images en se basant sur l'exemple ci-contre. On dispose d'une liste de listes B correspondant à la matrice de pixels de cette image en couleur. Le nombre de lignes est ici de 600 et le nombre de colonnes de 700.

I. Introduction

- Q 1** - Donner les commandes permettant de récupérer le nombre de lignes et le nombre de colonnes de cette image.
- Q 2** - Donner une estimation de la taille de l'image dans ce mode en ko (kilo-octet).
- Q 3** - Donner une instruction `Python` permettant de créer un pixel noir en mode nuances de gris puis de créer un pixel rouge et un pixel vert en mode RGB.
- Q 4** - Une image RGB est représentée par la liste suivante en Python :
 $L = [[[0,0,255], [0,255,0], [255,255,255]], [[255,0,0], [255,0,0], [0,0,0]], [[255,255,255], [0,0,0], [0,255,0]]]$
 Représenter l'image associée sur votre copie.
- Q 5** - Donnez une suite d'instructions `Python` permettant de construire un drapeau de la France avec 300 lignes et 600 colonnes. On rappelle que le drapeau de la France est un drapeau tricolore à bandes verticales bleu, blanc, rouge.
- Q 6** - Ecrire une fonction `bonnet_vert(image)` transformant le bonnet rouge en bonnet vert.

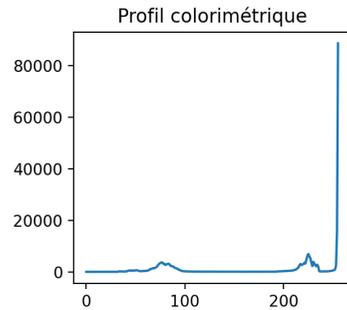
II. Manipulation d'image

On désire remplacer les valeurs du tableau image ci-dessous à gauche noté P par les valeurs du tableau B comportant l'image du bonnet. Les fonctions suivantes ont pour but d'effectuer le montage ci-dessous.



Il faut tout d'abord réduire l'image du bonnet d'un facteur $\alpha = 0,2$.

- Q 7** - Proposer une fonction `reduction(B,alpha)` renvoyant un tableau $B2$ tel que les dimensions du tableau soient réduites d'un coefficient `alpha`.
 On notera $B2$ le tableau comprenant l'image du bonnet réduit. On cherche maintenant à isoler le bonnet du fond blanc de l'image.
- Q 8** - Ecrire une fonction `couleurtogris(img)` permettant de passer d'une image en mode RGB à une image en mode nuance de gris. **La valeur du niveau de gris sera égale à $0,2 \times R + 0,7 \times G + 0,1 \times B$.**
- Q 9** - Quelle est la complexité de cette fonction ?
- Q 10** - Ecrire une fonction `niveau_gris(img)` qui prend en argument une image en niveau de gris et retourne son profil de gris. Cette fonction doit renvoyer une liste `profil` de taille 256 : en première position (indice 0), le nombre de pixels noirs (gris 0), en deuxième position (indice 1), le nombre de pixels gris 1, ..., en dernière position (indice 255), le nombre de pixels blancs (gris 255).



□ **Q 11** - la figure ci-contre correspond à l'application de la fonction précédente sur le tableau B du bonnet. Identifier les pixels correspondant au bonnet, ceux de la fourrure presque blanche et ceux du fond. Que choisir comme valeur seuil de niveau de gris pour isoler le bonnet du fond blanc ?

□ **Q 12** - On souhaite transformer l'image du bonnet en nuances de gris à une image en noir et blanc afin d'isoler les pixels du bonnet. Pour cela on utilise un seuil tel que :

- Si la valeur de gris du pixel est supérieure ou égale au seuil, le pixel est considéré blanc.
- Le pixel est considéré noir sinon.

Ecrire une fonction `gris2NB(img, seuil)` utilisant ce procédé.

□ **Q 13** - Quelle suite d'instructions devez vous appeler pour transformer le tableau réduit B2 en une image en noir et blanc ?

□ **Q 14** - Écrire une fonction `decoupe(img, seuil)` renvoyant une liste des coordonnées (i,j) des pixels de l'image dont le niveau de gris est en dessous du seuil déterminé ci-dessus. On pourra utiliser la fonction précédente.

□ **Q 15** - Proposer une suite d'instructions permettant de remplacer les pixels du tableau P par les pixels du tableau B sans le fond. L'insertion se fera à partir du coin supérieur gauche de coordonnées $i = 150$, et $j = 0$

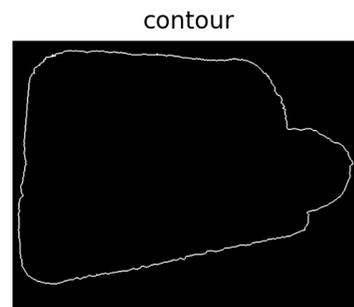
Pour améliorer le rendu de l'image P, il est possible de rajouter un liseret noir autour du bonnet pour souligner sa présence.

□ **Q 16** - On veut faire une détection de contour d'une image en noir et blanc. Pour cela, pour chaque pixel **qui n'est pas sur le bord de l'image**, on calcule le gradient via la relation :

$$\text{grad}[i][j] = \text{abs}(\text{img}[i-1][j] + \text{img}[i+1][j] + \text{img}[i][j-1] + \text{img}[i][j+1] - 4 * \text{img}[i][j])$$

Si le gradient est non-nul, on considère que le pixel est blanc, sinon on le considère noir.

Ecrire une fonction `contour(imgNB)` fonctionnant sur ce principe. Pour cette image, on obtient le résultat suivant :



□ **Q 17** - Proposer un code permettant d'obtenir un liseret noir sur le bonnet.

