

Augmentation du contraste d'une image

January 31, 2025

1 Position du problème

```
[1]: import numpy as np
import PIL.Image as Image
import matplotlib.pyplot as plt
```

```
[2]: im = Image.open("ecrevisse.png")
M = np.asarray(im)
```

1.1 Écrire une fonction `hist_img` qui renvoie l'histogramme de l'image à partir de sa matrice.

```
[3]: def hist_img(M : np.ndarray) -> np.ndarray:
    """Extrait l'histogramme (r,v,b) de la matrice image M.

    Args:
        M (np.ndarray): Matrice image contenant trois canaux de couleurs.

    Returns:
        np.ndarray: Matrice histogramme r,v,b ((3,256), int).
    """
    M_hist = np.zeros((3,256), dtype=int)
    n,m,k = M.shape
    for i in range(n):
        for j in range(m):
            r,v,b = M[i,j][0:3]
            M_hist[0][r] += 1
            M_hist[1][v] += 1
            M_hist[2][b] += 1
    return M_hist
```

```
[ ]: print(hist_img(M))
```

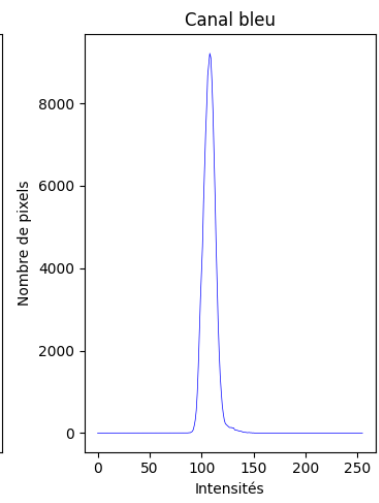
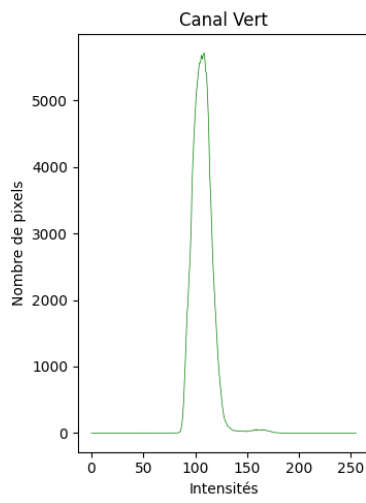
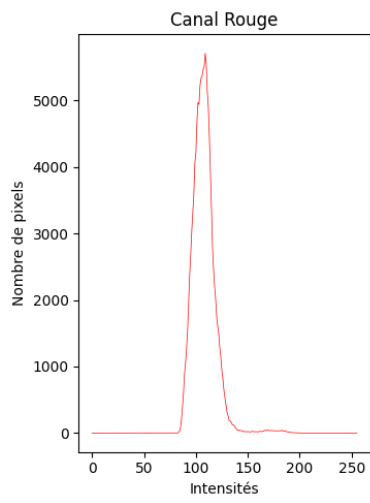
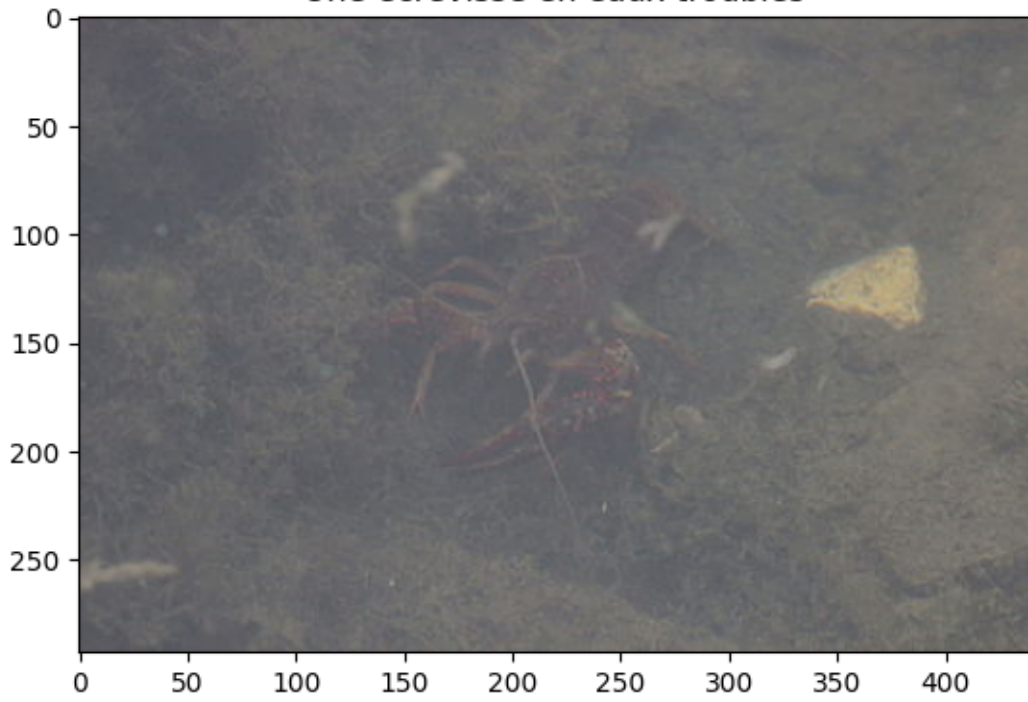
1.2 Écrire une fonction `plot_hist` qui à partir d'un histogramme, affiche les trois histogrammes d'une image (pour chaque couleur on représentera l'histogramme par une courbe continue).

```
[5]: def plot_hist(M_hist : np.ndarray) -> None:
      """Affiche la courbe des trois histogrammes de couleurs

      Args:
          M_hist (np.ndarray): Matrice histogramme r,v,b ((3,256), int).
      """
      graph = {
          0 : ("Red", "Canal Rouge"),
          1 : ("Green", "Canal Vert"),
          2 : ("Blue", "Canal bleu")
      }
      fig, ax = plt.subplots(1, 3, figsize=(11, 5))
      for i in range(3):
          ax[i].plot(range(256), M_hist[i], color=graph[i][0], linewidth=0.5)
          ax[i].set_title(graph[i][1])
          ax[i].set_xlabel("Intensités")
          ax[i].set_ylabel("Nombre de pixels")
      plt.tight_layout()
      plt.show()
```

```
[6]: fig, ax = plt.subplots(1, 1)
      ax.imshow(im)
      plt.title("Une écrevisse en eaux troubles")
      plt.show()
      plot_hist(hist_img(M))
```

Une écrevisse en eaux troubles



2 Augmentation du contraste par étalement de l'histogramme

2.1 Si la fonction f est affine, exprimer n' en fonction de n , a , b , a' , et b' .

Si f est affine, alors f est de la forme $f(n) = \alpha n + \beta$ avec $f(a) = a'$ et $f(b) = b'$

Par taux d'accroissement, on a :

$$\alpha = \frac{f(b) - f(a)}{b - a} \Rightarrow \boxed{\alpha = \frac{b' - a'}{b - a}}$$

On peut ensuite réinjecter cette équation dans l'équation $f(a) = a'$:

$$\begin{aligned} f(a) &= \alpha a + \beta \\ a' &= \alpha a + \beta \\ \beta &= a' - \alpha a \Rightarrow \boxed{\beta = a' - \left(\frac{b' - a'}{b - a}\right) a} \end{aligned}$$

Donc l'expression générale de n' en fonction de n , a , b , a' , b' est :

$$\begin{aligned} f(n) &= \alpha n + \beta \\ n' &= \alpha n + \beta \\ n' &= \left(\frac{b' - a'}{b - a}\right) n + a' - \left(\frac{b' - a'}{b - a}\right) a \Rightarrow \boxed{n' = a' + \frac{(b' - a')(n - a)}{b - a}} \end{aligned}$$

2.2 Si on veut étaler l'histogramme au maximum, quelles valeurs choisir pour a' et b' .

Si l'on veut étaler l'histogramme au maximum, il faut prendre le couple (a', b') tel que n' soit maximum. On peut donc prendre $\boxed{a' = 0}$ et $\boxed{b' = 255}$ (Étalement sur 8 bits) ce qui nous donne :

$$\begin{aligned} n' &= a' + \frac{(b' - a')(n - a)}{b - a} \\ n' &= 0 + \frac{(255 - 0)(n - a)}{b - a} \Rightarrow \boxed{n' = \frac{255(n - a)}{b - a}} \end{aligned}$$

2.3 Pourquoi l'histogramme n' -a-t-il plus l'allure d'une courbe continue ? Expliquer.

L'histogramme n' n'a plus l'allure d'une courbe continue. En effet en passant par f qui est un modèle affine, les intensités n peuvent ne plus être des entiers naturels compris entre 0 et 255 et seront potentiellement envoyé sur les mêmes valeurs de n' . Ces valeurs d'intensité sont donc arrondis, et dillaté par groupe ce qui rompt l'effet de continuité et crée des sauts/pics fréquents.

2.4 Pourquoi les valeurs extrêmes de l'intensité augmentent-elles ? Proposer une explication raisonnable en analysant les cas extrêmes des valeurs de n et n' .

Soit $n \in [a, b]$ et $n' \in [a', b']$

Étudions les cas extrêmes des valeurs de n et n' :

Fixons $n = a$:

$$n' = a' + \frac{(a - a)(b' - a')}{b - a} \Rightarrow \boxed{n' = a'}$$

Fixons $n = b$:

$$n' = a' + \frac{(b - a)(b' - a')}{b - a}$$
$$n' = a' + (b' - a') \Rightarrow \boxed{n' = b'}$$

Ainsi, les anciennes bornes de la plage d'intensité sont envoyés vers les nouvelles bornes. Mais de plus, si un pixel est inférieur ou supérieur aux bornes de l'ancienne plage, alors elle-aussi seras envoyé aux nouvelles bornes du fait que le pixel ne peut pas être "supprimé" de l'image. Elle est donc saturé. On a donc une augmentation des valeurs extrêmes de l'intensités, due au fait qu'il y a saturation des pixels inférieur ou supérieur aux bornes. Cela est cohérent car on cherche à rendre le contraste de notre image plus important.

2.5 Écrire une fonction seuils qui renvoie a et b à partir d'une coordonnée "Couleur" d'un histogramme d'une image passé en argument.

```
[7]: def seuils(couleur : np.ndarray, percent : float) -> (int,int):
    """Retourne les seuils d'un histogramme de couleur.

    Args:
        couleur (np.ndarray): Une matrice histogramme simple. (1x256)
        percent (float): Pourcent qui permet le réglage du seuil (en %).

    Returns:
        (int, int) : Tuple contenant le seuil minimal et maximal.
    """
    nb_total_px = 0
    for value in couleur:
        nb_total_px += value
    seuil = percent/100 * nb_total_px

    nb_pixel = 0
    for i in range(256): # 0, 1, ... 255
        if nb_pixel >= seuil:
            a = i
            break
        nb_pixel += couleur[i]
    nb_pixel = 0
    for j in range(255,-1,-1): #255, 254, ..., 0
        if nb_pixel >= seuil:
            b = j
            break
        nb_pixel += couleur[j]
    if b < a:
        raise ValueError("Les seuils se recoupent ! Diminuer le pourcentage !")
    return (a,b)
```

```
[8]: M_rouge, M_vert, M_bleu = hist_img(M)
print(f"Seuil du canal rouge avec un seuil à 1%: {seuils(M_rouge, 1)}")
print(f"Seuil du canal vert avec un seuil à 2%: {seuils(M_vert, 2)}")
print(f"Seuil du canal bleu avec un seuil à 3%: {seuils(M_bleu, 3)}")
```

Seuil du canal rouge avec un seuil à 1%: (90, 144)

Seuil du canal vert avec un seuil à 2%: (92, 126)

Seuil du canal bleu avec un seuil à 3%: (99, 118)

2.6 Écrire une fonction `aug_contraste` qui prend en argument une matrice d'image `M` et un float `percent` et qui renvoie la matrice correspondant à l'image avec contraste augmenté.

```
[9]: def f(n : int, a : int, b : int, a_ : int = 0, b_ : int = 255) -> int:
    """Fonction qui a f associe n, et qui renvoie n' en fonction de a, b, a', b'

    Args:
        n (int): intensité de couleur compris entre 0 et 255.
        a (int): Seuil inférieur de la plage d'intensités.
        b (int): Seuil supérieur de la page d'intensités.
        a_ (int, optional): Nouveau Seuil inférieur (0 par défaut)
        b_ (int, optional): Nouveau Seuil supérieur (255 par défaut)

    Returns:
        float: La nouvelle intensité de couleur.
    """
    if n < a:
        n = a
    elif n > b:
        n = b
    return a_ + ((n - a) * (b_ - a_)) / (b - a) # Question 3.
```

```
[10]: def aug_contraste(M : np.ndarray, percent : float) -> np.ndarray:
    """Renvoie la matrice image M avec contraste augmentée.

    Args:
        M (np.ndarray): Matrice image
        percent (float): Pourcentage utilisée par le seuil.

    Returns:
        np.ndarray: Matrice image avec contraste augmentée.
    """
    M_hist = hist_img(M)
    a_r, b_r = seuils(M_hist[0], percent)
    a_v, b_v = seuils(M_hist[1], percent)
    a_b, b_b = seuils(M_hist[2], percent)
    M2 = np.zeros(M.shape, dtype=np.uint8)
    n, m, k = M.shape
    for i in range(n):
        for j in range(m):
            r, v, b = M[i, j][0:3]
            r_, v_, b_ = f(r, a_r, b_r), f(v, a_v, b_v), f(b, a_b, b_b)
            M2[i, j][0:3] = r_, v_, b_
            M2[i, j][3:] = M[i, j][3:]
    return M2
```

```
[11]: M2 = aug_contraste(M, 1)
im2 = Image.fromarray(M2)
fig, ax = plt.subplots(1, 1)
ax.imshow(im2)
plt.title("Une écrevisse en eaux troubles (Contrasté)")
plt.show()
plot_hist(hist_img(M2))
```

