

Chapitre 8 : Traitement d'images

I- Qu'est-ce qu'une image ?

Une photo est associée à tableau à 2 dimensions (image en niveau noir et blanc) ou 3 dimensions (image en couleurs avec une composant pour le vert, le bleu et le rouge) dont les valeurs positives sont souvent converties sur 8 bits : *uint8*.

Les valeurs rencontrées sont donc comprises entre 0 (noir) et 255 (intensité maximale). La de manipulation des tableaux nécessite une réflexion afin d'éviter les problèmes d'overflow suivant car les valeurs sont 256 périodiques.

-256	-1	0	1...	255	256	511		
0			255	0				255	0			255		

II- Listes et tableaux numpy

`import numpy as np` # Préfixe np à utiliser pour importer toutes les fonctions

a) Comparaisons

Un tableau `numpy` ne suit pas les mêmes règles de calcul qu'une liste :

	Liste	Tableau numpy
Création	#creation d'une liste <code>L1=[0,1,2,3]</code>	#creation d'un tableau d'une ligne (équivalent à un vecteur) <code>T1=np.array([0,1,2,3])</code> L'exemple ci-dessus peut s'interpréter comme la conversion d'une liste en tableau : <code>T4=np.array(L1)</code> On peut utiliser un 2 ^e argument optionnel pour fixer ou convertir le type de données (par défaut <code>int64</code>) <code>T4=np.array(T1, dtype="uint8")</code>
	#creation d'une liste de liste <code>L2=[[0,1,2,3],[4,5,6,7]]</code>	#creation d'un tableau 2D : <code>T2=np.array([[0,1,2,3],[4,5,6,7]])</code>
Multiplier tous les éléments de la séquence par un entier	<code>L3=2*L1</code> #L3 est une concaténation de L1 avec L1	<code>T3=2*T1</code> #Chaque élément de T1 est multiplié par 2 (on parle de vectorisation)

b) Opérations « de bases » sur les tableaux numpy

Lignes de commandes	Interpréteur
"""deux fonctions intéressantes pour créer des tableaux 1D en ligne""" <code>T5=np.linspace(0,4,5)</code> # (départ, fin incluse, nbre de points) <code>T6=np.arange(0,5,1)</code> # (départ, fin exclue, incrément)	<code>[0. 1. 2. 3. 4.]</code> <code>[0 1 2 3 4]</code>
"""initialisation : création d'un tableau de zéros""" <code>T7=np.zeros((2,3))</code> # ((nbre de lignes, nbre de colonnes))	<code>[[0. 0. 0.]</code> <code>[0. 0. 0.]]</code>

"""somme de tous les éléments d'un tableau""" <code>s=np.sum(T1)</code>	6
"""obtenir les dimensions d'un tableau""" <code>lignes,colonne=np.shape((T7))</code> #renvoie un tuple (nbre lignes, nbre colonnes) <code>lignes= np.shape((T7)) [0]</code>	(2, 3) 2
"""accès à une valeur ou création d'un sous tableau""" <code>T8=np.array([[0,1,2,3],[4,5,6,7],[8,9,10,11]])</code> <code>print(T8[1])</code> #renvoie la ligne d'index 1 (2 ^e ligne) <code>print(T8[-1])</code> #renvoie la dernière ligne <code>print(T8[1,2])</code> #renvoie le motif en 2e ligne et 3e colonne <code>print(T8[1,:])</code> #renvoie la ligne d'index 1 <code>print(T8[1,-1])</code> #renvoie la ligne d'index 1 sans le dernier élément <code>print(T8[:,1])</code> #renvoi tout le colonne d'index 1 <code>print(T8[1:3,1:3])</code> # [index le ligne:index de la dernière ligne exclue, index le colonne:index de la dernière colonne exclue] <code>print(T8[-1::-1,-1::-1])</code> #début, fin, pas =-1 pour fixer le sens inverse <code>print(T8[[True,False,False],:])</code> #que le ligne	<code>[4 5 6 7]</code> <code>[8 9 10 11]</code> 6 <code>[4 5 6 7]</code> <code>[4 5 6]</code> <code>[1 5 9]</code> <code>[[5 6]</code> <code>[9 10]]</code> <code>[[11 10 9 8]</code> <code>[7 6 5 4]</code> <code>[3 2 1 0]]</code> <code>[[0 1 2 3]]</code>

c) Obtenir le négatif d'une photo avec une liste et avec un tableau

```
from PIL import Image
import numpy as np
import time
im=Image.open("tours.JPG") #ouverture du tableau
tab=np.array(im, dtype="uint8") #conversion
l,c,p=np.shape(tab) #dimension 1 : lignes, colonnes, p=1 ou 3
```

liste	Tableau
<code>liste=tab.tolist()</code> #conversion du tableau en liste <code>t0=time.time()</code> #initialisation de la mesure du temps <code>for i in range(l):</code> <code>for j in range(c):</code> <code>for k in range(p):</code> <code>liste[i][j][k]=255-liste[i][j][k]</code> #calcul du négatif <code>print(time.time()-t0)</code> #temps d'exécution <code>tab_new=np.array(liste, dtype="uint8")</code> <code>photo_new=Image.fromarray(tab_new)</code> <code>photo_new.show()</code>	<code>tab_new=np.copy(tab)</code> <code>t0=time.time()</code> <code>tab_new=255-tab</code> <code>print(time.time()-t0)</code> <code>photo_new=Image.fromarray(tab_new)</code> <code>photo_new.show()</code>

La vectorisation des tableaux numpy permet un gain de temps négligeable dans la manipulation de photos

Commenté [AM5]: Autrement appelé découpage ou slicing

Commenté [AM6]: On pourra noter une différence avec les listes de listes pour obtenir des tranches :
Liste[index_ligne][index_colonne]
Tab[index_ligne,index_colonne] ou
Tab[index_lignes][index_colonne]

Commenté [AM1]: Avec les tableaux numpy, la conversion est fixée par

Commenté [AM2]: Exemples de dépassement :
255 + 1 = 0
0 - 1 = 255

Commenté [AM3]: On peut aussi écrire :
`from numpy import *`
`from numpy import nom de fonction`

Commenté [AM4]: Les séquences ou structures de données rencontrées depuis le 1^{er} semestre sont :
-Les chaînes et les tuples (qui sont immuables : les valeurs stockées dans ces séquences ne peuvent être modifiées par affectation après création- pour les chaînes une modification par la méthode `replace` est possible mais conduit à la création d'une nouvelle chaîne avec une autre adresse mémoire)
-Les listes (muables, on dit aussi variables)