

TD Traitement de l'image

Exercice 1 (extrait du concours juin 2021) : Une image en niveaux de gris est une collection de pixels organisés sous forme matricielle, où chaque pixel est défini par :

- ses coordonnées (y, x) où **y** est l'indice de ligne et **x** est l'indice de colonne ;
- sa valeur, parmi 256 valeurs possibles entre le noir et le blanc (0 pour le noir et 255 pour le blanc), représentant le niveau de gris du pixel.

Les 8-voisins d'un pixel p de coordonnées (y, x) sont ses voisins immédiats, comme l'illustre la figure ci-dessous.

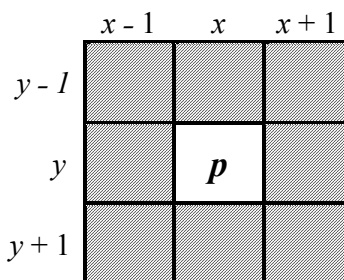


Figure : Les 8-voisins immédiats (cases hachurées) du pixel p .

1. Écrire une fonction, nommée **voisins**, qui prend en entrée un tuple p contenant les coordonnées d'un pixel et retourne un ensemble formé par les coordonnées des 8 pixels voisins immédiats du pixel de coordonnées p . Aucun traitement particulier n'est à prévoir pour les pixels associés aux bords de l'image
2. Écrire une fonction, nommée **sont_voisins**, qui prend deux tuples p et q représentant les coordonnées de deux pixels, retourne **True** si p et q sont voisins et **False** sinon.
3. Une **région** est un ensemble de pixels représentant une partie d'une image. La figure ci-dessous illustre une image à 3 lignes et 5 colonnes où les cases hachurées forment une région r :

	0	1	2	3	4
0	128	64	128	32	32
1	64	255	0	64	128
2	0	0	255	64	54

pixel de la région de coordonnées (1, 3) avec un ton de gris = 64

L'ensemble des tuples de la région $r = \{ (1,1), (1,2), (1,3), (2,2), (2,3) \}$

Un **masque** associé à une image en niveaux de gris, est une image en noir et blanc où chaque pixel a une valeur booléenne, **False** pour le noir et **True** pour le blanc.

Écrire une fonction, nommée **masque**, qui prend en entrée :

- un ensemble de tuples, noté r , représentant une région ;
 - un tuple de 2 entiers, noté **taille**, contenant respectivement la hauteur et la largeur du masque à retourner.
 - Cette fonction retourne un masque où seuls les pixels de la région r sont blancs.
4. La **frontière** d'une région r est formée par les pixels qui possèdent un voisin extérieur à la région r . Écrire une fonction, nommée **frontiere**, qui prend en entrée un ensemble de tuples représentant une région r , et retourne un ensemble contenant les coordonnées des pixels appartenant à la frontière de r .

Exercice 2 (Filtre moyennneur) : Le filtre moyennneur est une opération de traitement d'images utilisée pour réduire le bruit dans une image. Le filtre moyennneur fait partie de la catégorie des filtres d'images locaux car pour calculer la nouvelle valeur d'un pixel, il regarde la valeur des pixels proches. Concrètement, la valeur filtrée d'un pixel p est égale à la moyenne des valeurs des pixels proches de p . En général, on définit les « pixels proches de p » comme l'ensemble de pixels contenus dans un carré de largeur k centré sur p .

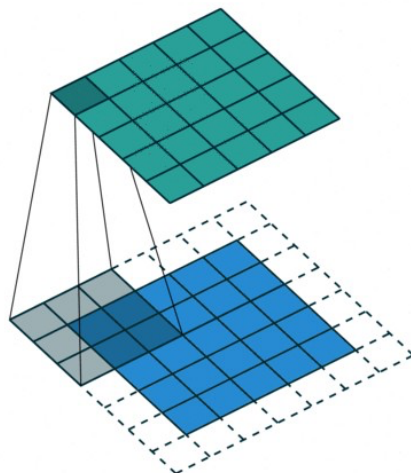
24	32	128	240	255
12	42	111	154	222
4	23	123	176	243
15	63	145	134	172
27	12	98	75	143

		108		

Prenons l'exemple d'un filtre moyennneur de paramètre $k = 3$. Afin de calculer la nouvelle valeur du pixel au centre de l'image de gauche, on calcule la valeur moyenne des pixels situés dans **un carré de dimension 3×3 centré sur ce pixel**. Cela donne la nouvelle valeur du pixel sur **l'image transformée (l'image de droite)**

$$\frac{(42 + 111 + 154) + (23 + 123 + 176) + (63 + 145 + 134)}{9} = 108$$

Cette opération est répétée pour tous les pixels de l'image. On parle de **fenêtre glissante** pour caractériser le **carré sur lequel est calculé la moyenne** des pixels et qui se déplace sur l'image. La fenêtre se déplace sur l'image du bas pour calculer les valeurs de la nouvelle image en haut :



QUESTION : Écrire une fonction **Filtre_moyennneur(img)** permettant de **calculer la valeur moyenne des pixels situés dans un carré de dimension 3×3 centré sur chaque pixel de l'image** passée en paramètre (**img**) .

Exercice 3 (extrait du concours 2018) : Binariser une image consiste à la partitionner en deux parties : une **partie sombre** et une **partie claire** selon un **seuil** de niveau de gris (voir la figure ci-dessous)



Une image peut être binarisé selon un seuil optimal déterminé par une méthode de seuillage. Dans la suite, nous allons décrire les étapes d'une **méthode de seuillage** qui consiste à déterminer un **couple de paramètres** intervenant dans le calcul du **seuil optimal**.

Description de la méthode :

Soit une image en niveaux de gris représentée par une matrice M à n lignes et m colonnes, où $M_{i,j} \in [0, 255]$ est le niveau de gris associé au pixel d'indices (i, j) ($0 \leq i \leq n-1$ et $0 \leq j \leq m-1$).

Les étapes de la méthode consistent à :

- Déterminer la matrice, notée V , à n lignes et m colonnes nommée matrice des moyennes des voisinages, où chaque élément $V_{i,j}$ ($0 \leq i \leq n-1$ et $0 \leq j \leq m-1$) a pour valeur :
 - $M_{i,j}$ pour $(i=0)$ ou $(j=0)$ ou $(i=n-1)$ ou $(j=m-1)$;
 - la partie entière de la moyenne des niveaux de gris de l'élément $M_{i,j}$ et de ses 8 voisins immédiats, pour $0 < i < n-1$ et $0 < j < m-1$.
- Déterminer une matrice carrée d'ordre 256, notée H et nommée **matrice histogramme**, où chaque élément $H_{i,j}$ ($0 \leq i \leq 255$ et $0 \leq j \leq 255$) représente le **nombre des pixels** ayant un niveau de gris égal à i et une moyenne des niveaux de gris du voisinage égale à j
- Déterminer la matrice colonne, notée Moy , définie par :

$$Moy = \begin{pmatrix} \sum_{i=0}^{255} \sum_{j=0}^{255} i \cdot H_{i,j} \\ \sum_{i=0}^{255} \sum_{j=0}^{255} j \cdot H_{i,j} \end{pmatrix}$$

- A partir de la matrice H , déterminer **parmi tous les couples (s, t) possibles**, le couple **optimal** dont la valeur de la trace de la matrice variance ci-dessous définie, est maximale.

Le **couple optimal** est déterminé comme suit :

- Pour chaque couple d'indices (s, t) de H tels que $0 \leq s \leq 255$ et $0 \leq t \leq 255$:
 - Calcul des Probabilités P_0 et P_1 définies par :

$$P_0 = \frac{1}{m \cdot n} \sum_{i=0}^s \sum_{j=0}^t H_{i,j} \quad P_1 = \frac{1}{m \cdot n} \sum_{i=s+1}^{255} \sum_{j=t+1}^{255} H_{i,j}$$

- Construction des matrices colonnes Moy_0 et Moy_1 définies par :

$$Moy_0 = \begin{pmatrix} \frac{1}{P_0} \sum_{i=0}^s \sum_{j=0}^t i \cdot H_{i,j} \\ \frac{1}{P_0} \sum_{i=0}^s \sum_{j=0}^t j \cdot H_{i,j} \end{pmatrix} \quad Moy_1 = \begin{pmatrix} \frac{1}{P_1} \sum_{i=s+1}^{255} \sum_{j=t+1}^{255} i \cdot H_{i,j} \\ \frac{1}{P_1} \sum_{i=s+1}^{255} \sum_{j=t+1}^{255} j \cdot H_{i,j} \end{pmatrix}$$

- Construction de la matrice variance S définie par :

$$S = P_0 A A^T + P_1 B B^T \quad \text{où } A = (Moy_0 - Moy) \text{ et } B = (Moy_1 - Moy)$$
- Calcul de la valeur de la trace de la matrice S et mise à jour de la valeur de la trace maximale. On rappelle que la trace d'une matrice correspond à la somme des valeurs des éléments de sa diagonale principale.

Travail demandé

N.B : Dans la suite, les fonctions demandées seront écrites en langage Python, en utilisant impérativement la nomenclature donnée par le tableau suivant :

Nom	Type	Description
M	<i>numpy.ndarray</i>	Matrice d'entiers (entre 0 et 255) représentant une image
V	<i>numpy.ndarray</i>	Matrice d'entiers contenant les moyennes des voisinages de M
n, m	<i>int</i>	Respectivement nombre de lignes et de colonnes de M et V
H	<i>numpy.ndarray</i>	Matrice histogramme d'ordre 256 correspondant à M et V

1. Écrire la fonction, nommée **GenererMat**, qui prend en paramètres **n** et **m**, et retourne une matrice **M** remplie par des valeurs entières aléatoires comprises entre 0 et 255.
2. Écrire la fonction, nommée **MoyVoisinage**, qui prend en paramètre **M** et retourne **V**
3. Écrire la fonction, nommée **NbPixels**, qui prend en paramètres **M**, **V** et deux entiers **i** et **j**. Elle retourne le **nombre** d'éléments de **M** ayant $M_{a,b} = i$ et $V_{a,b} = j$ (avec $0 \leq a \leq n-1$ et $0 \leq b \leq m-1$)
4. Écrire la fonction, nommée **Histogramme**, qui prend en paramètres **M** et **V** et retourne la matrice **H**.
5. Écrire la fonction, nommée **Sigma**, qui prend en paramètres **H**, **binf1**, **binf2**, **bsup1**, **bsup2** et **comp** (paramètre ayant la valeur par défaut 0), calcule et retourne :

$$\sum_{i=\text{binf}_1}^{\text{bsup}_1} \sum_{j=\text{binf}_2}^{\text{bsup}_2} i \cdot H_{i,j} \quad \text{si } \text{comp} \text{ égal à } 0$$

$$\sum_{i=\text{binf}_1}^{\text{bsup}_1} \sum_{j=\text{binf}_2}^{\text{bsup}_2} j \cdot H_{i,j} \quad \text{si } \text{comp} \text{ égal à } 1$$

6. Écrire la fonction, nommée **Seuillage**, qui à partir du paramètre **H**, retourne le couple optimal (**s**, **t**) correspondant au maximum de toutes les valeurs des traces des matrices **S**.

À retenir :

- Le pixel en haut à gauche correspond au pixel de coordonnées (0,0).
- Le pixel en haut à droite est le pixel de coordonnées (largeur-1 , 0).
- Le pixel en bas à gauche a pour coordonnées (0 , hauteur-1).
- Le pixel en bas à droite correspond au pixel (largeur-1, hauteur-1).
- Pour une image en niveaux de gris, le code *du pixel* est un nombre compris entre 0 et 255.
- Pour une image en couleurs, chaque *pixel p* sera codé par ses trois composantes p[0], p[1] et p[2], qui sont les composantes RVB, comprises entre 0 et 255.