USTHB FI 2024-2025.
M2 SII. TP N°1
Module : Vision par ordinateur.

Exercice 1 :

Sous Visual Studio Code, en utilisant OpenCV et NumPy :

- Créez un nouveau projet nommé **TP1**.
- Chargez une image en niveaux de gris dans une variable **img** à l'aide de la fonction **imread**.
- Vérifiez si le chargement s'est effectué correctement avec l'instruction : **« if img is None : »**
- Créez une nouvelle image **imgResultat** de la même taille que **img**.
- Modifiez les valeurs des pixels de **imgResultat** en leur attribuant l'inverse des valeurs de **img** (c'est-à-dire **255 - valeur_pixel**).
- Affichez **img** et **imgResultat**.
- Changez le type de l'image **img** en **uint32**, **float64**, etc., et affichez l'image à chaque fois.
- Utilisez l'outil **Debug** de **VSCode** pour exécuter le programme pas à pas et afficher les valeurs de quelques pixels.
- Testez les fonctions **imwrite**, **copy** ainsi que le **slicing** (:).

Exercice 2 :

- Refaire l'Exercice 1, mais cette fois avec une image en couleur.
- Effectuer la conversion de l'image couleur en une image en niveaux de gris.

Exercice 3 (Examen 2021-2022) :

Compléter le programme Python ci-dessous en répondant aux questions suivantes :

1. Ecrire une fonction qui retourne la position du pixel noir dans l'image.
2. Ajouter les instructions nécessaires afin de déplacer le pixel noir avec un pas (en nombre de pixels) dans les quatre directions en tapant sur les touches 2, 4, 6 et 8. ('4' : Gauche, '6' Droite, '8' : Haut, '2' : Bas) et '0' pour quitter le programme.
3. Afficher le pixel noir dans l'image avec un carré (5x5 pixels) sans l'utilisation des boucles
4. Sauvegarder l'image après le dernier déplacement avec un type de compression qui garde l'image sans aucun changement.
5. Nous voulons afficher une forme (cercle, ellipse, croix ….) à la place du carré. Quel est le filtre qui permet de réaliser cette opération et comment ? (donner des explications)

USTHB FI 2024-2025.
M2 SII. TP N°1
Module : Vision par ordinateur.

```python
import cv2
import numpy as np
from random import randrange

def createImgWithPointRand(h,w):
    img = np.ones((heigthImg,widthImg),np.float32)
    #randrange(x) return une valeur aléatoire entre 0 et x
    randPointY,RandPointX = randrange(heigthImg),randrange(widthImg)
    img[randPointY,RandPointX] = 0
    return img


heigthImg=200
widthImg =400

img = createImgWithPointRand(heigthImg,widthImg)

cv2.imshow('image gray',img)

'''la fonction waitKey return le code ASCII d'un caractère dans
la variable q. Le code ASCII de '0'=48, '1'=49, '2'=50....etc. '''
q = cv2.waitKey(0) & 0xFF
cv2.destroyAllWindows()
```

Exercice 4 (Normalisation d'image):

L'histogramme normalisé : aussi appelé étirement d'histogramme ça consiste à normaliser le niveau de gris entre 0 et 255, ceci est intéressant notamment pour des images sombres mais l'inconvénient est le bruit qui entachera d'avantage le résultat.

- Créez un programme qui permet de normaliser une image en utilisant la fonction suivante :

$$\text{PixelNormalisé} = \frac{(Pixel - NiveauGris_{min}) \times 255}{NiveauGris_{max} - NiveauGris_{min}}$$