

TP N° : 02

Opérations de base sur les images

Objectifs

- Apprendre à manipuler des images en appliquant des opérations simples comme l'inversion (création d'un négatif) et l'addition d'images.

Instructions

1. Inversion d'une image

- Chargez l'image « rose_1024.tif »
- Utilisez une opération de négatif pour inverser les couleurs de l'image.
- Comparez l'image originale et l'image inversée en les affichant côte à côte.

2. Addition de deux images

- Chargez deux images (toADD et rose_1024) de même taille ou assurez-vous que les images sont redimensionnées si elles ne sont pas de la même taille (par exemple, avec cv2.resize).
- Effectuez l'addition pixel par pixel des deux images.
- Vérifiez si la valeur de l'addition est dans la plage acceptable de 0 à 255.
- Affichez l'image résultante pour observer les effets de l'addition.

3. Rotation d'une image

- Chargez l'image « rose_1024.tif »
- Effectuez une rotation de 45° de l'image
- Affiché l'image résultante

Fonctions utiles

Addition de deux images

Après avoir essayer la méthode manuelle pour obtenir la somme des pixels utilisez la fonction `cv2.add(image1, image2)` pour additionner

Rotation d'une image

Utiliser la fonction **`cv2.getRotationMatrix2D(center, angle, scale)`** pour effectuer une rotation autour d'un point donné (généralement le centre de l'image).

- **center** : (x,y) qui indique le point autour duquel la rotation va être effectuée
- **angle** : L'angle de rotation en degrés. Un angle positif fait tourner l'image dans le sens inverse des aiguilles d'une montre, et un angle négatif la fait tourner dans le sens des aiguilles d'une montre.
- **scale** : Un facteur d'échelle (1.0 garde la taille originale). Par exemple, si `scale=0.5`, l'image sera réduite à 50 % de sa taille d'origine après la rotation.

Une fois la matrice de rotation calculée, applique cette matrice pour transformer l'image.

`cv2.warpAffine(src, M, dsize)`

- **src** : L'image source, que l'on veut transformer.
- **M** : La matrice de transformation (ici, la matrice de rotation obtenue via `cv2.getRotationMatrix2D()`).
- **dsize** : (largeur,hauteur) qui représente la taille de l'image résultante.