TP2 : Opérations arithmétiques sur les images avec OpenCV

Objectifs

- Additionner les pixels de deux images
- Soustraire les pixels de deux images
- Multiplier les pixels de deux images
- Diviser chaque pixel de d'une image par celui d'une autre image
- Combiner deux images avec des poids personnalisés
- Augmenter ou diminuer la luminosité d'une image
- Calculer la valeur absolue de la différence entre deux images

1. Chargement et redimensionnement des images

```
import cv2
import numpy as np
# Charger les images
img1_url = r'I:\Mon Drive\ENSEIGNEMENT SUPERIEUR ODG\Informatique\Python\ESI Traitement Image\paysage.jpg'
img2_url = r'I:\Mon Drive\ENSEIGNEMENT SUPERIEUR ODG\Informatique\Python\ESI Traitement Image\lumiere.jpg'

# Charger deux images depuis les chemins
img1 = cv2.imread(img1_url)
img2 = cv2.imread(img2_url)

# Redimensionner img2 pour qu'elle ait la même taille que img1
img2 = cv2.resize(img2, (img1.shape[1], img1.shape[0]))

# Affichage des deux images originales
cv2.imshow("Image 1", img1)
cv2.imshow("Image 2", img2)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- cv2.imread(...) : charge une image à partir du disque.
- img1.shape[1], img1.shape[0] : donne les dimensions (largeur, hauteur) de l'image.
- cv2.resize(...) : ajuste img2 pour avoir la même taille que img1, nécessaire pour les opérations pixel à pixel.
- cv2.imshow(...) et cv2.waitKey(0) : affichent les images jusqu'à ce qu'une touche soit pressée.

2. Addition

```
add = cv2.add(img1, img2)
cv2.imshow("Addition", add)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- cv2.add() ajoute **chaque pixel** de img1 et img2.
- Les valeurs sont **saturées à 255**, c'est-à-dire que si R+G > 255, la valeur devient 255 (évite le débordement).
- Utile pour **fusionner** des images ou **augmenter la luminosité**.

3. Soustraction

```
sub = cv2.subtract(img1, img2)
cv2.imshow("Soustraction", sub)
cv2.waitKey(0)
cv2.destroyAllWindows
```

- cv2.subtract() effectue la soustraction pixel à pixel.
- Négatif est **ramené à zéro** (0 est le minimum).
- Pratique pour **détecter les différences entre deux images**.

4. Multiplication

```
mult = cv2.multiply(img1, img2)
cv2.imshow("Multiplication", mult)
cv2.waitKey(0)
```

- cv2.multiply() multiplie pixel par pixel.
- Si les deux images sont sombres, le résultat est encore plus sombre.
- Utile pour appliquer **des masques** ou **pondérer** une image par une autre.

5. Division

```
div = cv2.divide(img1, img2)
cv2.imshow("Division", div)
cv2.waitKey(0)
```

- cv2.divide() divise chaque pixel de img1 par celui de img2.
- OpenCV gère les divisions par zéro.
- Utile pour **corriger une variation d'intensité** (normalisation).

6. Fusion pondérée (addWeighted)

```
# Mélange pondéré : 70% de img1 + 30% de img2
alpha = 0.7
beta = 0.3
blended = cv2.addWeighted(img1, alpha, img2, beta, 0)

cv2.imshow("Fusion pondérée", blended)
cv2.waitKey(0)
```

- cv2.addWeighted() : combine deux images avec des **poids personnalisés**.

- Formule : result = img1 * alpha + img2 * beta + gamma
- Très utilisé pour faire des **fondues ou des superpositions**.

7. Luminosité – Ajouter ou soustraire de la lumière

```
# Crée une image blanche de même taille que img1
lumiere = np.ones(img1.shape, dtype='uint8') * 50

# Plus lumineux
brighter = cv2.add(img1, lumiere)

# Plus sombre
darker = cv2.subtract(img1, lumiere)

cv2.imshow("Plus lumineux", brighter)
cv2.imshow("Plus sombre", darker)
cv2.waitKey(0)
```

- np.ones(...)*50 crée une image blanche (valeurs 50).
- Ajouter cette image revient à **augmenter la luminosité globale**.
- Soustraire la même image **diminue la luminosité**.

8. Différence absolue (détection de mouvement)

```
diff = cv2.absdiff(img1, img2)
cv2.imshow("Différence absolue", diff)
cv2.waitKey(0)
```

- cv2.absdiff() calcule la **valeur absolue de la différence** entre deux images.
- Très utile pour détecter ce qui a **changé entre deux images** (ex : vidéosurveillance).

Résumé des opérations

Action	Fonction OpenCV
Redimensionner une image	cv2.resize()
Additionner	cv2.add()
Soustraire	cv2.Subtract()
Multiplier	cv2. Multiply()
Diviser	cv2. divide ()
Fusion pondérée	cv2. addWeighted ()
Différence absolue	cv2. absdiff ()