

# TP N° : 03

## Histogrammes

### Objectifs du TP

- Afficher la luminosité et le contraste d'une image.
- Calculer et afficher manuellement les histogrammes d'une image en niveau de gris et des canaux de couleur (rouge, vert, bleu) d'une image en couleur.
- Appliquer les différentes méthodes d'amélioration de la luminosité et de contraste

### Étapes du TP

#### 1/ Image en niveau de gris

Commencez par charger une image en niveau de gris. Vous pouvez utiliser l'image `rose.tif`

1. **Chargement de l'image :**
  - Charger l'image spécifiée (par exemple, `rose_1024.tif`) et la convertir en niveaux de gris
2. **Calcul de la luminosité :**
  - Calculer la luminosité de l'image en utilisant la formule vue en classe
3. **Calcul du contraste :**
  - Évaluer le contraste de l'image en utilisant la formule vue en classe
4. **Calcul de l'histogramme :**
  - Initialiser un tableau de zéros pour stocker les valeurs de l'histogramme (256 niveaux de gris).
  - Parcourir chaque pixel de l'image et incrémenter le tableau correspondant à l'intensité du pixel.
5. **Affichage des résultats :**
  - Afficher la luminosité et le contraste calculés.
  - Tracer et afficher l'histogramme des intensités de pixel de l'image.

#### 2/ Image en couleur

Commencez par charger une image en couleur. Vous pouvez utiliser n'importe quelle image à votre disposition.

1. **Chargement de l'image :**
  - Vous pouvez utiliser n'importe quelle image à votre disposition.
2. **Séparation des canaux :**
  - Séparez l'image en trois canaux (bleu, vert, rouge) à l'aide de `cv2.split()`.
3. **Calcul des histogrammes :**
  - Créez trois tableaux de zéros pour les histogrammes (un pour chaque canal).

- Parcourez chaque pixel des canaux et incrémentez le tableau correspondant à l'intensité de pixel.
4. **Affichage des histogrammes :**
- Tracer les histogrammes de chaque canal dans la couleur correspondante.

### 3/ Appliquer des méthodes d'amélioration de la luminosité et de contraste

- Choisir une image fournie pour ce TP
- Calculer la luminosité son histogramme normalisé et cumulé.
- Appliquer des méthodes de contraste telles que la translation de l'histogramme (par exemple + 50) ; l'extension de l'histogramme ; l'expansion dynamique et l'égalisation d'histogramme.
- Comparer les histogrammes obtenus pour chaque méthode et les résultats d'OpenCV avec les implémentations manuelles.

#### Remarque :

- **Luminosité** : Moyenne des valeurs des pixels.
- **Contraste** : Écart type des valeurs des pixels.
- **Translation de l'histogramme** :
  - La fonction `translation_histogramme` augmente la valeur des pixels de l'image d'un certain offset (ici, +50).
- **Expansion de dynamique** :
  - La fonction `expansion_dynamique` normalise l'image pour qu'elle utilise toute la plage de valeurs (0 à 255).
- **Egalisation d'histogramme** :
  - La fonction `egalisation_histogramme` améliore le contraste de l'image en redistribuant les intensités de pixels.

#### Fonctions utiles

Après avoir calculé les histogrammes manuellement, vous pouvez utiliser les fonctions OpenCV :

```
1. cv2.calcHist([channel], [0], None, [256], [0, 256]) :
```

- **channel** : le canal de l'image d'entrée (Bleu, Vert ou Rouge)
- **[0]** : l'indice du canal à analyser (puisque chaque canal est traité comme une image en niveaux de gris).
- **None** : aucune masque n'est utilisé, donc l'ensemble de l'image est pris en compte.
- **[256]** : le nombre de bins (256 valeurs correspondant aux intensités de pixels de 0 à 255).
- **[0, 256]** : la plage des valeurs d'intensité (de 0 à 255).

```
2. cv2.normalize(hist, hist, 0, 255, cv2.NORM_MINMAX)
```

- **hist** : L'histogramme à normaliser.
- La plage 0, 255 spécifie que les valeurs de l'histogramme doivent être normalisées pour être entre 0 et 255.

`cv2.NORM_MINMAX` : Spécifie que la normalisation doit s'effectuer sur la plage minimale et maximale de l'histogramme.

3. `cv2.equalizeHist(img)` pour l'histogramme égalisé