

Repertoire Git: <https://github.com/serineissaad/Image2-TP1-Notebook>

# Histogramme

## 1. Normalization Histogramme

Après avoir défini la fonction getHist retournant l'histogramme représentant une image, telle qu'on a le nombre de pixels ( $Y$ ) ayant un niveau de gris ( $X$ ), nous avons procédé à la normalisation de ces valeurs dans l'intervalle  $[0, 1]$ , en divisant les valeurs d'occurrences  $Y$  par le nombre total de pixels.

## 2. Etirement d'histogramme

Il s'agit de répartir le nombre d'occurrences des pixels sur une largeur de l'histogramme comprise entre un intervalle  $[0, 255]$  ( $a = 0, b = 255$ ).

- Les valeurs supérieures à 255 deviennent 255.
- Les valeurs inférieures à 0 deviennent 0.
- Les valeurs entre 0 et 255 subissent la transformation suivante

$$I_2(x, y) = (b - a) \cdot \frac{I_1(x, y) - N_{min}}{N_{max} - N_{min}} + a$$

$N_{min}$  et  $N_{max}$  sont les valeurs minimales et maximales de l'histogramme de l'image.

$I_1(x, y)$  est l'intensité du pixel original.

### a. Egalisation d'histogramme

Afin d'égaleriser l'image, nous utilisons la formule:

$$F(x, y) = 255 * \frac{Hc(l)}{Nb}$$

Telle que 255 correspond à  $2^D - 1$  avec  $D = 8$ .

Pour chaque niveau de gris  $l$  issu de l'histogramme cumulé, nous calculons une nouvelle valeur  $F(x, y)$ . Le graphe résultant représente, pour chaque niveau de gris de l'image originale (axe des X), le nouveau niveau de gris associé à notre image égalisée (axe des Y).

Ensuite, avec deux boucles imbriquées, nous parcourons chaque pixel de l'image originale pour ajuster sa valeur selon son nouveau niveau de gris déterminé par l'histogramme égalisé.

## Produit de Convolution et Filtrage

Nous avons choisi d'appliquer deux types différents de filtres pour remarquer les variations des résultats qu'on peut atteindre en utilisant la convolution. Le processus de filtrage passe par deux étapes:

### 1. Le traitement des bordures

Vu qu'on ne peut pas appliquer des filtres sur les pixels en bordures de l'image originale  $M * N$ , nous allons créer une nouvelle image de taille  $M' * N'$  telle que

$$M' = M + (2 * paddingM)$$

$$N' = N + (2 * paddingN)$$

A noter que padding est la marge à ajouter, c'est-à-dire le nombre de pixels à ajouter pour une seule bordure. Il est calculé par  $n/2$  et  $m/2$  ( $m * n$  est la matrice de filtrage).

Cette matrice est remplie de zéros initialement, puis, l'image originale est copiée dans le centre de la nouvelle matrice, en laissant une marge de zéros autour d'elle.

### 2. Application du filtre et calcul de la convolution

L'image de sortie par contre aura la même taille que l'image originale, vu qu'on ne va appliquer le filtre qu'aux pixels originaux. Ce, en parcourant chaque pixel de la nouvelle image (sans inclure le padding), en extrayant sa région qui est de la taille du filtre, et en multipliant cette dernière par le filtre. En sommant les produits obtenus, nous obtenons la nouvelle valeur du pixel en cours.

On multiplie chaque élément de la région par l'élément correspondant du filtre et on fait la somme de tous ces produits. Cela nous donne la valeur de convolution pour le pixel courant.

On retourne donc l'image après s'être assuré que chaque valeur (niveau de gris d'un pixel) est comprise entre 0 et 255.

**Note:** les filtres choisis sont: Flou de Gauss et le filtre de la Détection de contours.