[Histogramme](https://fr.statisticseasily.com/histogramme-vs-graphique-%C3%A0-barres/" \o "En savoir plus sur l'histogramme) L'égalisation est une technique de traitement d'image sophistiquée utilisée pour améliorer le contraste des images. En redistribuant les niveaux d'intensité des pixels d'une image, cette méthode vise à obtenir un histogramme uniforme. L'objectif principal est d'améliorer la visibilité des caractéristiques d'une image, ce qui la rend particulièrement utile dans les scénarios où l'image d'origine peut être peu contrastée ou masquée par les conditions d'éclairage.  
  
L’égalisation d’histogramme consiste à **répartir les niveaux de gris (ou les couleurs)** d’une image de manière plus uniforme, pour que **tous les niveaux de luminosité soient mieux représentés**.

**Un histogramme :**

Un histogramme est une représentation graphique de la distribution des intensités de pixels dans une image

Un **histogramme d’image** montre combien de pixels ont une certaine intensité lumineuse.  
Par exemple :

* Une image sombre aura un histogramme concentré à gauche (niveaux de gris faibles).
* Une image très claire aura un histogramme concentré à droite.

La convolution est courante en traitement d’images. Elle consiste en une opération de multiplication de deux matrices de tailles différentes (généralement une petite et une grande), mais de même dimensionnalité semblable (p.ex. 1D, 2D), produisant une nouvelle matrice (également de même dimensionnalité). La convolution est donc le traitement d’une matrice (p.ex. une image) par une autre petite matrice appelée matrice de convolution ou noyau (kernel). Le filtre parcourt toute la matrice principale (p.ex. l’image) de manière incrémentale et génère une nouvelle matrice constituée des résultats de la multiplication. Notez qu’il y a une marge dans la matrice finale pour laquelle nous ne pouvons pas calculer de valeur. Dans le traitement d’image, ceci est utilisé par exemple pour effectuer un flou gaussien, ou détourer les éléments d’une photo (exemples sur http://stephanieluu.com/image-convolution/ ).

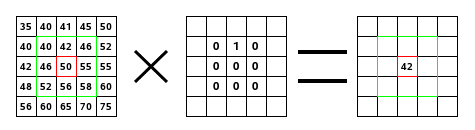
C’est un **mécanisme de filtrage**. On applique un **noyau (masque)** sur chaque pixel pour calculer une nouvelle valeur en fonction de ses voisins.

🔍 **But** : modifier l’apparence de l’image (flouter, lisser, détecter les contours, etc.).

La **convolution** est une technique utilisée en **traitement d’images**, qui applique une **petite matrice appelée noyau (ou filtre)** sur une **grande matrice** (souvent une image). Le noyau parcourt l’image **de façon incrémentale**, en effectuant des **multiplications** à chaque position pour générer une **nouvelle matrice**. Cette méthode permet, entre autres, de réaliser des effets comme le **flou gaussien** ou la **détection de contours**.

La **convolution** est une technique utilisée en **traitement d’images**. Elle consiste en une opération de **multiplication de deux matrices** de tailles différentes (généralement une petite et une grande), mais de même dimensionnalité semblable (par exemple 1D, 2D), produisant une **nouvelle matrice** (également de même dimensionnalité).

La convolution est donc le **traitement d’une matrice** (une image) par une autre petite matrice appelée **matrice de convolution**, **le filtre** ou noyau (kernel). Le **filtre parcourt toute la matrice de l’image** de manière incrémentale et génère une nouvelle matrice constituée des résultats de la multiplication

. Le **filtre étudie successivement chacun des pixels** de l'image. Pour chaque pixel, que nous appellerons « **pixel initial** », il **multiplie la valeur de ce pixel et de chacun des 8 pixels qui l'entourent** par la valeur correspondante dans le noyau. Il **additionne l'ensemble des résultats** et le pixel initial prend alors la valeur du résultat final.  
Vite un exemple, simple :

À gauche se trouve la **matrice de l'image** : chaque pixel est indiqué par sa valeur. Le **pixel initial** est encadré de rouge. La **zone d'action du noyau** est encadrée de vert. Au centre, se trouve le **noyau** et, à droite, le **résultat de la convolution**.  
Voici ce qui s'est passé : le filtre a **lu successivement**, de gauche à droite et de haut en bas, les pixels de la zone d'action du noyau et il a **multiplié chacun d'eux par la valeur correspondante du noyau**, puis **additionné les résultats**. Le pixel initial a pris la valeur **42** :  
(40\*0) + (42\*1) + (46\*0) + (46\*0) + (50\*0) + (55\*0) + (52\*0) + (56\*0) + (58\*0) = **42**.

L’image suivante montre le code de la classe Convolution qui hérite de la classe Processing1. Elle contient un constructeur prenant en paramètre une image d’entrée, un noyau de convolution ainsi qu’un booléen optionnel inPlace pour déterminer si le traitement modifie directement l’image originale, elle contient aussi une méthode Process() qui permet d’appliquer l’opération de convolution. La classe propose également trois méthodes statiques : creerMoyenneur, creerGaussien et creerExponentiel, qui permettent respectivement de générer des noyaux de convolution de type moyenneur, gaussien et exponentiel