

Notes de cours

Semaine 15

Cours Turing

Corrélation et convolution

Une convolution, dans le traitement d'images, est une opération qui a de nombreuses applications. Elle est notamment utilisée pour le floutage, la détection de contours, la détection de lignes, la détection de formes, etc.

Corrélation en deux dimensions

La corrélation est une opération mathématique qui permet de mesurer la similarité entre deux signaux. Elle est présentée en premier, car son fonctionnement est plus intuitif que celui de la convolution.

Soit deux tableaux de nombres A et B de même dimension $m \times n$.

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \quad B_{m,n} = \begin{pmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m,1} & b_{m,2} & \cdots & b_{m,n} \end{pmatrix}$$

La corrélation de A par B est donnée par la formule suivante :

$$A_{m,n} \otimes B_{m,n} = \begin{matrix} a_{1,1} \cdot b_{1,1} & + & a_{1,2} \cdot b_{1,2} & + & \cdots & + & a_{1,n} \cdot b_{1,n} & + \\ a_{2,1} \cdot b_{2,1} & + & a_{2,2} \cdot b_{2,2} & + & \cdots & + & a_{2,n} \cdot b_{2,n} & + \\ \vdots & + & \vdots & + & \ddots & + & \vdots & + \\ a_{m,1} \cdot b_{m,1} & + & a_{m,2} \cdot b_{m,2} & + & \cdots & + & a_{m,n} \cdot b_{m,n} & \end{matrix}$$

Voici un exemple numérique de corrélation entre deux tableaux A et B de dimension 3×3 :

$$A_{m,n} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad B_{m,n} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

$$A * B = \begin{matrix} 1 \cdot 1 & + & 2 \cdot 2 & + & 3 \cdot 3 & + \\ 4 \cdot 4 & + & 5 \cdot 5 & + & 6 \cdot 6 & + \\ 7 \cdot 7 & + & 8 \cdot 8 & + & 9 \cdot 9 & \end{matrix} = 285$$

Convolution

La convolution est équivalente à la corrélation, mais avec une matrice B tournée de 180 degrés.

Soit deux tableaux de nombres A et B de même dimension $m \times n$:

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \quad B_{m,n} = \begin{pmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m,1} & b_{m,2} & \cdots & b_{m,n} \end{pmatrix}$$

La convolution de A par B est donnée par la formule suivante :

$$A_{m,n} * B_{m,n} = \begin{matrix} a_{1,1} \cdot b_{m,n} & + & a_{1,2} \cdot b_{m,n-1} & + & \cdots & + & a_{1,n} \cdot b_{m,1} & + \\ a_{2,1} \cdot b_{m-1,n} & + & a_{2,2} \cdot b_{m-1,n-1} & + & \cdots & + & a_{2,n} \cdot b_{m-1,1} & + \\ \vdots & + & \vdots & + & \ddots & + & \vdots & + \\ a_{m,1} \cdot b_{1,n} & + & a_{m,2} \cdot b_{1,n-1} & + & \cdots & + & a_{m,n} \cdot b_{1,1} & \end{matrix}$$

Voici un exemple numérique de convolution entre deux tableaux A et B de dimension 3×3 :

$$A_{m,n} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad B_{m,n} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

$$A * B = \begin{matrix} 1 \cdot 9 & + & 2 \cdot 8 & + & 3 \cdot 7 & + \\ 4 \cdot 6 & + & 5 \cdot 5 & + & 6 \cdot 4 & + \\ 7 \cdot 3 & + & 8 \cdot 2 & + & 9 \cdot 1 & \end{matrix} = 165$$

Convolution d'une image par un filtre

La convolution est une opération très utilisée pour analyser des images à l'aide de filtres. Un filtre est généralement une matrice carrée et plus petite que l'image.

Comme le filtre est plus petit que l'image, il ne sera possible d'analyser qu'une sous-partie de l'image avec une convolution. Cependant, la convolution sera effectuée entre chaque sous-partie de l'image et le filtre. C'est-à-dire que le filtre sera déplacé sur toute la surface de l'image afin de créer une nouvelle image.

Si une image a une dimension $H \times L$ et le filtre a une dimension $K \times K$, l'image résultante aura les dimensions $(H - (K - 1), L - (K - 1))$.

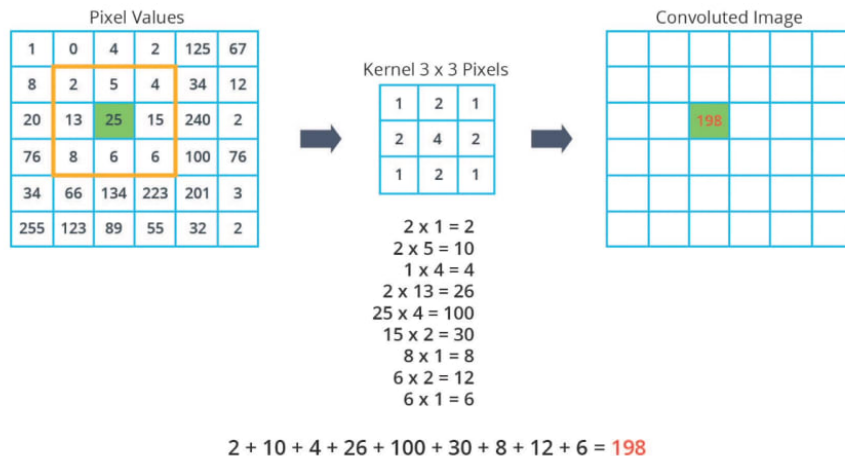


Figure 1: Convolution d'une image par un filtre

Il est possible d'obtenir une image résultante avec les mêmes dimensions que l'image originale en utilisant certaines techniques comme le "zero-padding" : le tableau est entouré de 0 pour que le filtre puisse être appliqué sur les bords de l'image.

! Lorsque l'on effectue une convolution d'une image par un filtre, le résultat peut être interprété comme une nouvelle image. Dans cette nouvelle image, chaque pixel est le résultat d'une opération entre le filtre et une sous-partie de l'image, c'est-à-dire que chaque pixel de l'image résultante dépend de plusieurs pixels voisins de l'image originale.

Création de filtre

Nous allons maintenant voir comment créer des filtres pour réaliser des opérations sur des images. Les filtres sont généralement des tableaux carrés de taille impaire.

Identité

Voilà 3 exemples de filtres d'identité :

$$(1) \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

En utilisant un seul 1 et que des 0, il est possible de créer un filtre d'identité de n'importe quelle taille. Chaque pixel du résultat final dépend uniquement d'un pixel original et pas du tout de ses voisins.

Voilà comment créer un filtre d'identité de taille 3 x 3 en Python:

```
filtre = np.array([
[0, 0, 0],
[0, 1, 0],
[0, 0, 0],
])
```

Flou

Contrairement au filtre d'identité, pour flouter, il faut prendre en compte les pixels voisins.

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Le résultat de ce filtre est visible dans la figure 2.

Le problème ici est que le filtre est trop fort et qu'il y a eu débordement de capacité.

En effet, en supposant l'intensité maximale de 255 pour le pixel analysé et ses voisins, le résultat de la convolution donnera 2295 ce qui est supérieur à 255.

Il faut donc le diviser par la somme de ses éléments pour que la somme des éléments du filtre soit égale à 1 et que le résultat de la convolution ne dépasse pas 255 : $\frac{9 \cdot 255}{9} = 255$

$$\frac{1}{9} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Le résultat de ce filtre est visible dans la figure 3.

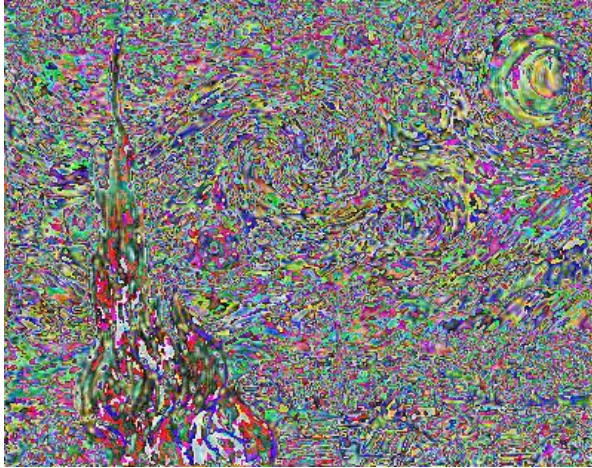


Figure 2: Débordement de capacité



Figure 3: Flou avec un filtre dont la somme des valeurs est égale à 1

Un filtre de flou peut avoir n'importe quelle taille tant que la somme de ses éléments est égale à 1. Plus le filtre sera grand, plus le flou sera important comme montré dans la figure 4
À noter qu'avec cette définition, un filtre de flou de taille 1x1 correspond à un filtre d'identité.



Figure 4: Flou avec des filtres de tailles différentes, de gauche à droite : 1x1, 5x5, 9x9, 13x13, 17x17

Accentuation des contours

Pour accentuer les contours, il faut utiliser un filtre qui met en évidence les différences de luminosité entre les pixels voisins.

Pour cela, il faut utiliser un filtre qui a des valeurs négatives et positives.

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Si le pixel analysé est plus lumineux que ses voisins, le résultat de la convolution sera positif.

Si le pixel analysé est moins lumineux que ses voisins, le résultat de la convolution sera négatif.

Si le pixel analysé est de la même luminosité que ses voisins, le résultat de la convolution sera nul.

Le résultat de ce filtre est visible dans la figure 5 où la moitié droite de l'image est l'image accentuée.



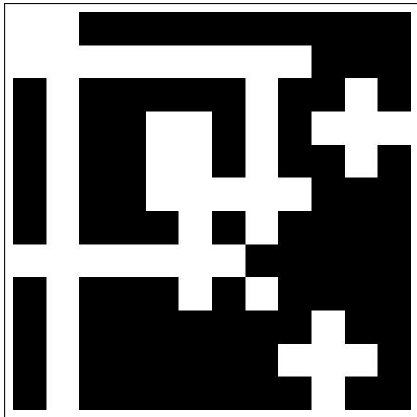
Figure 5: Dans la moitié gauche, l'image originale, dans la moitié droite, l'image accentuée

! Il est important de noter que ce filtre peut produire des valeurs négatives et des valeurs plus grandes que 255. Pour régler ce problème, les valeurs inférieures à 0 sont mises à 0 et les valeurs supérieures à 255 sont mises à 255 en utilisant la fonction `numpy.clip`.

```
img_np = img_np.clip(0,255)
```

Recherche de motifs

Un filtre peut également être utilisé pour représenter et rechercher un motif dans une image.



$$\begin{pmatrix} -1 & 1 & -1 \\ 1 & 1 & 1 \\ -1 & 1 & -1 \end{pmatrix}$$

Figure 7: Filtre de recherche

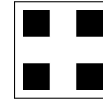


Figure 8: Visualisation du filtre recherché

Figure 6: Original

En supposant que l'on cherche le motif de la figure 8 dans l'image de la figure 6, nous allons utiliser le filtre de la figure 7 pour rechercher le motif.

Dans la figure , nous pouvons voir la matrice résultante de la convolution entre l'image et le filtre ainsi qu'une visualisation dans la figure .

$$\begin{pmatrix} 4 & 1 & 3 & 3 & 3 & 2 & 4 & 1 & 0 & 1 \\ 1 & -1 & -2 & -1 & -1 & -2 & 1 & -1 & 0 & 0 \\ 3 & -1 & 0 & 2 & 2 & -1 & 3 & 0 & 0 & 5 \\ 3 & -1 & -1 & 2 & 1 & -1 & 1 & -1 & 0 & 0 \\ 3 & -1 & 0 & 1 & 4 & -1 & 5 & 0 & 0 & 1 \\ 1 & -1 & -2 & 0 & -1 & 1 & -1 & 1 & -1 & 0 \\ 5 & 1 & 3 & 1 & 5 & -2 & 3 & -2 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 2 & 0 & 0 & 1 & -1 \\ 3 & -1 & 0 & -1 & 1 & -2 & 0 & 0 & 0 & 1 \\ 3 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 5 & 0 \end{pmatrix}$$

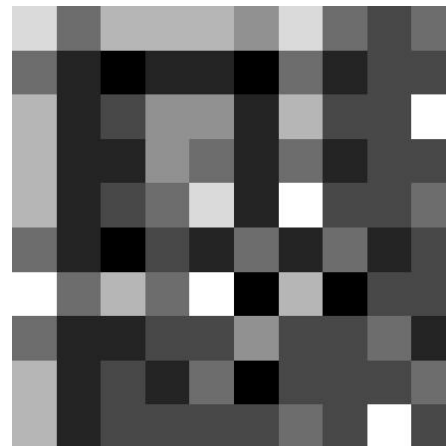


Figure 9: Résultat numérique et visuel de la convolution entre l'image (Figure 6) et le filtre (Figure 8)

En supprimant les valeurs non maximales, nous obtenons les résultats ci-dessous.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 \end{pmatrix}$$

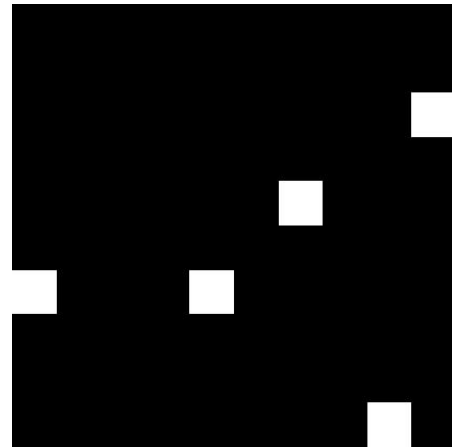


Figure 10: Résultat numérique et visuel de la convolution entre l'image (Figure 6) et le filtre (Figure 8) après suppression des valeurs non-maximales

Dans la figure 11, nous pouvons visualiser les endroits de l'image qui ont produit des réponses d'intensité maximale.

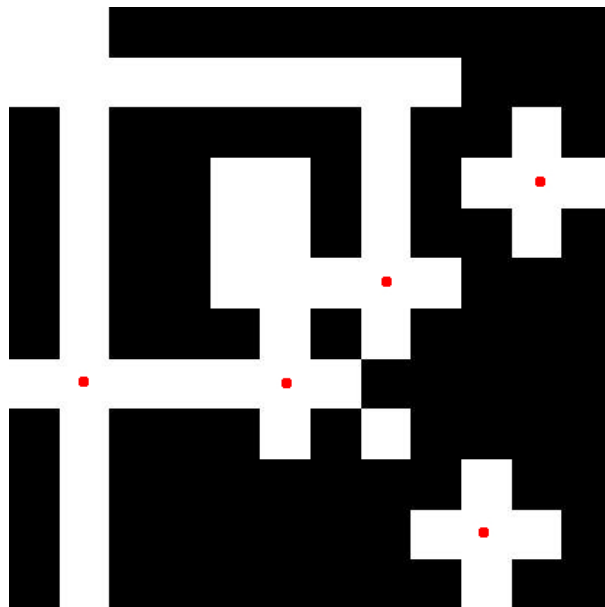


Figure 11: Original