



Descriptif:

Classe Convolution_layer

Encadrant: **Hugo Bolloré**

Par: **Aicha Maaoui**

Date: 31/12/2021

Problématique

Le but de cette partie est la description de la classe *Convolution_layer* implémentée dans le code. Généralement, la convolution utilise un ensemble de filtres (formés par des Kernels) pour extraire les caractéristiques (features) d'une image donnée à l'entrée.

Classe "Convolution_layer"Generalités:

Les images en couleur sont représentées par une matrice de pixels. Un pixel se dispose de 3 canaux RGB (Rouge-Vert-Bleu), comme illustré dans la figure (1). Chaque filtre doit de même avoir 3 canaux, égal à l'image d'entrée, comme illustré dans la figure (2).

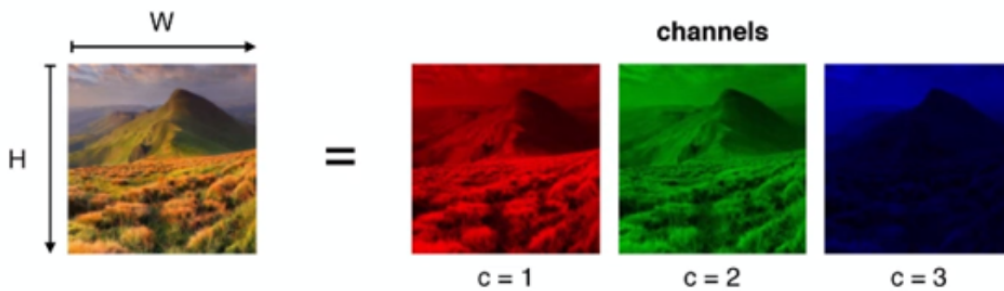
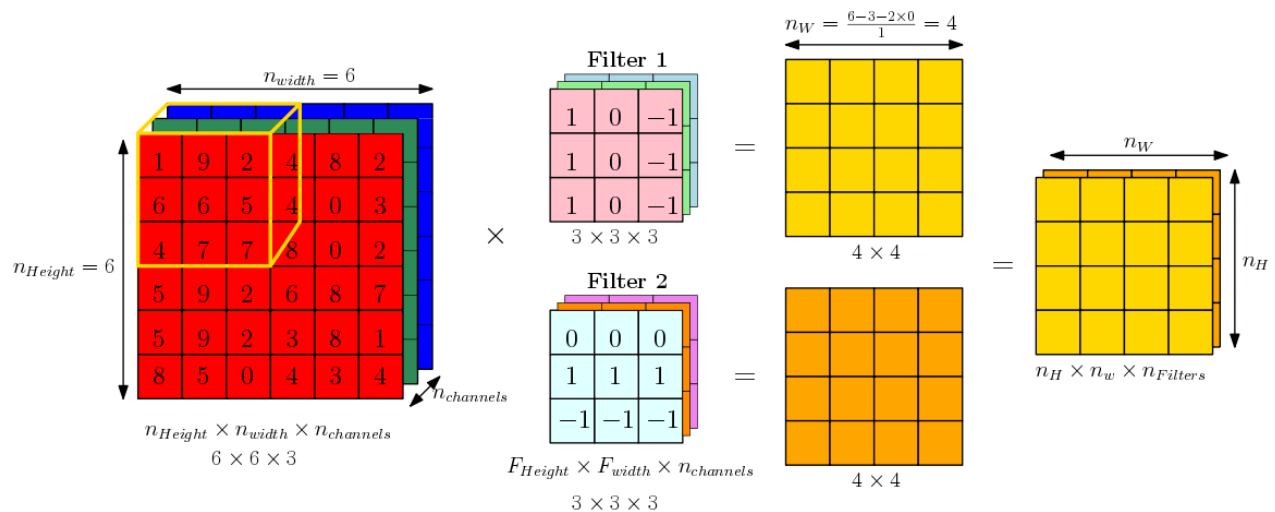


Figure 1: Représentation d'une image par 3 canaux RGB.

Figure 2: Convolution avec deux filtres, stride $S = 1$, padding $p = 0$, nombre de canaux = 3.

Initialement, l'image d'entrée est considérée comme un volume de dimension ($input_number \times Matrix_height \times Matrix_width \times Channels_number$). Après la réduction du nombre de canaux de 3 à 1, chaque image d'entrée sera considérée comme une matrice 2D de dimensions ($input_number \times Matrix_height \times Matrix_width$).

Chaque filtre appliqué aux images d'entrée est un volume de dimension initiale ($Filter_height \times Filter_width \times Channels_number$). Par conséquent, les filtres disposent d'un volume total de ($Filter_height \times Filter_width \times$

$Channels_number \times Filters_number$). Après réduction de nombre de canaux de 3 à 1, les filtres totaux appliqués à l'image auront comme dimensions $(Filter_height \times Filter_width \times Filters_number)$.

La couche de convolution (ou *convolution_layer*) est la couche de base d'un réseau de neurones convolutif. Elle représente la matrice de valeurs obtenues en sommant les produits entre chaque pixel de l'image et le pixel du filtre. Soit l'exemple illustré dans la figure (2).

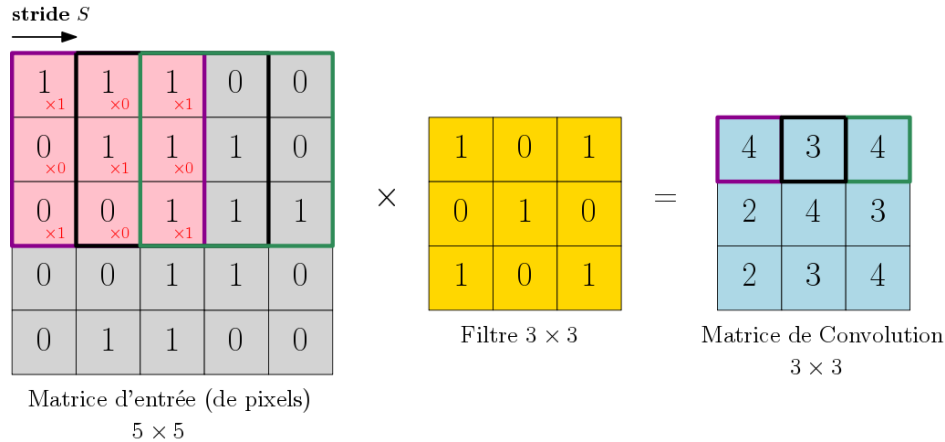


Figure 3: Matrice de convolution, stride $S = 1$, padding $p = 0$.

On explique dans ce qui suit les étapes d'obtention de la matrice de convolution [1] [2].

Paramètres de dimensionnement du volume de la couche de convolution (volume de sortie):

- **1. Filter (or Kernel) size (dimensions du filtre):** indique le nombre de pixels du filtre. A titre d'exemple, le filtre présenté dans la figure (3) est de taille 3×3 ,
- **2. Stride (or offset) S (pas):** indique le déplacement (en nombre de pixels) à chaque itération.
Dans la figure (3), on a un stride $S = 1$, ainsi chaque filtre est décalé de 1 pixel par rapport au bloc de la matrice d'entrée, auquel il est superposé. Cette procédure est représentée dans la figure (3) par les cadres en mauve, noir et vert dans la matrice d'entrée.
- **3. Padding p (Marge à zéro):** Parfois, dans le but de contrôler la dimension spatiale du volume de sortie, on met des zéros à la frontière du volume de l'image d'entrée.

En revenant à notre exemple illustré dans la figure (3), on a $n = 5$ pixels à l'entrée, le filtre a comme dimension $f = 3$. Alors, la matrice de convolution aura comme dimension $n_1 = n - f + 1 = 3$ (on risque de perdre quelques informations).

Maintenant, si on souhaite obtenir une dimension $n_1 = 5$ de la matrice de convolution obtenue comme sortie (égale à la dimension d'entrée), et on a une taille fixe du filtre $f = 3$, alors on doit chercher n tel que $n_1 = 5$. Dans ce cas, on trouvera $n = 7$.

Par conséquent, on ajoute un bloc enveloppant la matrice d'entrée formée par les pixels, comme illustrée dans la figure (4). Le but de cette procédure est d'éviter la perte des données en pixels de la matrice d'entrée.

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0

Matrice d'entrée (de pixels)
 7×7

Figure 4: Matrice d'entrée en pixels (avec bloc de 0: Zero padding).

En raison de simplicité de la couche de convolution, on ne considère pas de padding dans notre projet ($p = 0$). Il s'agit d'une convolution valide.

La dimension du volume de sortie est décrite dans l'équation (1).

$$Largeur_{matrice\ de\ sortie} = \frac{Largeur_{matrice\ d'entrée} - Largeur_{filtre} + 2 \times Padding}{Stride} + 1 \quad (1)$$

On applique l'équation (1) dans notre projet (où la taille de matrice d'entrée est 50×50 pixels et le filtre est 3×3). on aura comme taille de matrice de convolution à la sortie:

$$Largeur_{matrice\ de\ sortie} = \frac{50-3+2 \times 0}{1} + 1 = 50 - 2 = 48 = Largeur_{matrice\ d'entrée} - 2$$

Soit la figure (5), illustration du procédure de convolution de ce projet.

Étapes de Convolution):

Pour produire la matrice de convolution à la sortie de la couche, on utilise une image d'entrée et un filtre. Les étapes de convolution de ce filtre avec l'image d'entrée sont:

- Superposition du filte avec l'image d'entrée,
- Multiplication membre-à-membre de pixels dans le filtre et ceux correspondants dans l'image d'entrée,
- sommation de valeurs obtenus pour un emplacement de filtre donné,
- Répétition de la procédure de convolution pour tous les emplacements du filtre.

Les paramètres utilisés dans le projet de reconnaissance des images de plantes sont regroupés dans le tableau (1).

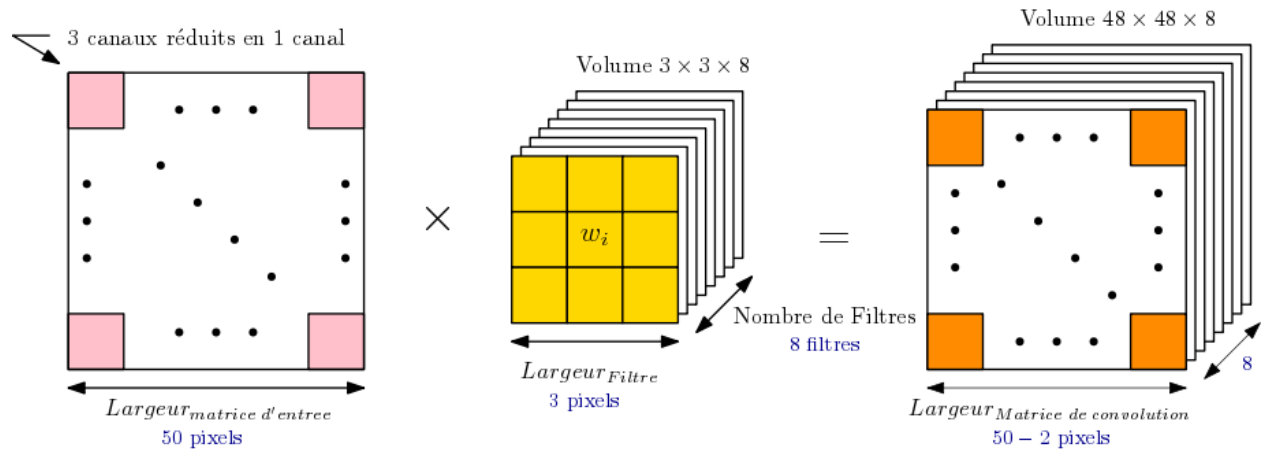


Figure 5: Couche de convolution du Projet.

Dimensions de l'image d'entrée	50 × 50
Nombre de canaux de l'image d'entrée	3 (RGB)
Nombre de canaux à l'entrée de <i>Convolution_layer</i>	1
Dimensions du filtre	3 × 3
Nombre de filtres utilisés	8
Stride S	1
Padding p	0

Table 1: Paramètres de la couche de convolution du projet.

Le choix du filtre dans le cadre du projet est justifié par le fait qu'un filtre de dimensions impaires et minimales est le plus utilisé [3].

Le nombre total des poids dans les 8 filtres est égal à: $3 \times 3 \times 8 = 72$ poids.

Le volume de sortie de la matrice de convolution est de dimension $(48 \times 48 \times 8)$.

References:

- [1] "Réseau neuronal convolutif", Wikipedia, 2021, "https://fr.wikipedia.org/wiki/R%C3%A9seau_neuronal_convolutif".
- [2] "Convolutional neural network", Wikipedia, 2021, "https://en.wikipedia.org/wiki/Convolutional_neural_network".
- [3] "Deciding optimal kernel size for CNN", Sabyasachi Sahoo, Towards Data Science, 2018, "<https://towardsdatascience.com/deciding-optimal-filter-size-for-cnns-d6f7b56f9363>".