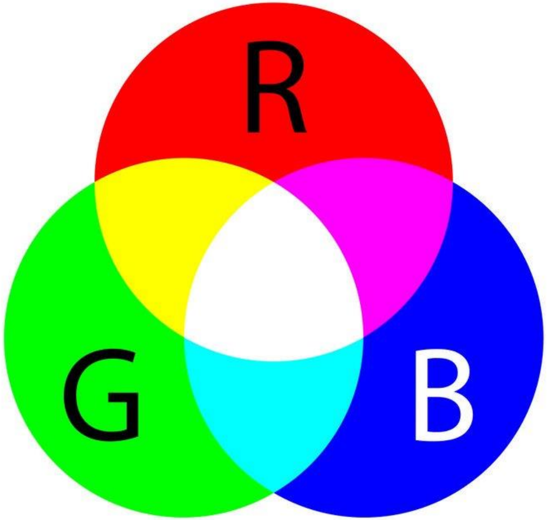
Deep learning par la pratique

Leçon 1 : Computer vision

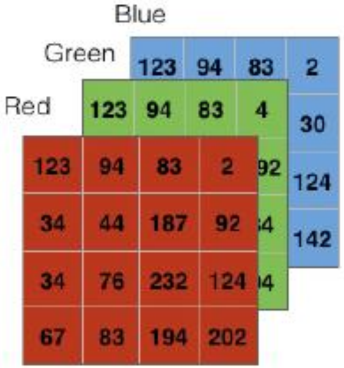
Présenté par **Morgan Gautherot**

Une image en informatique



(4x4x3)

=

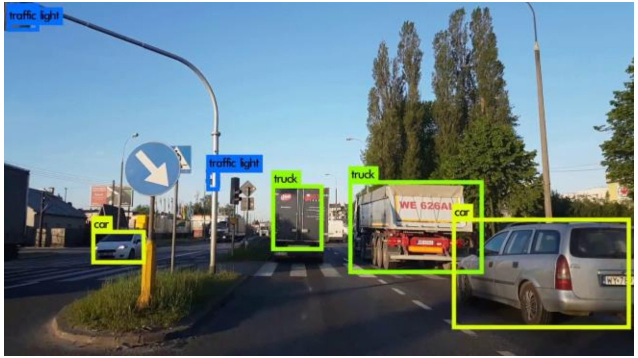


Computer vision

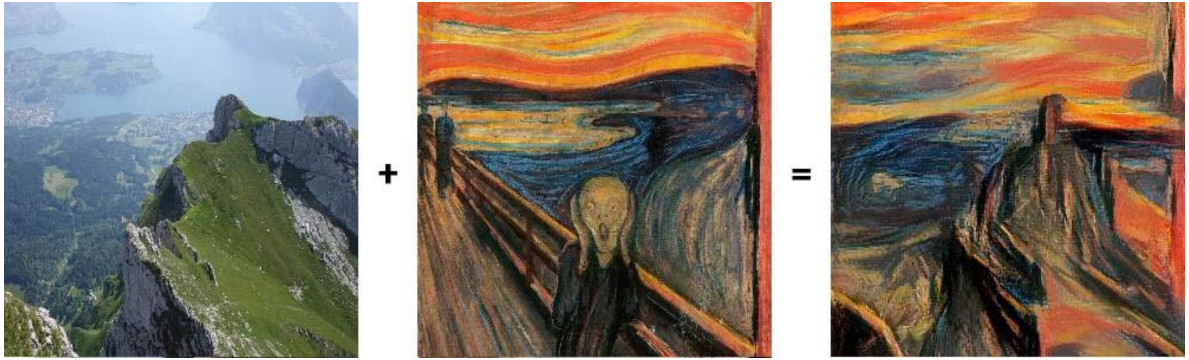
Classification d’images

→ Cat ? (0/1)

Détection d’objets



Transfert de style



Un grand nombre de caractéristiques

64 x 64

64 x 64 x 3

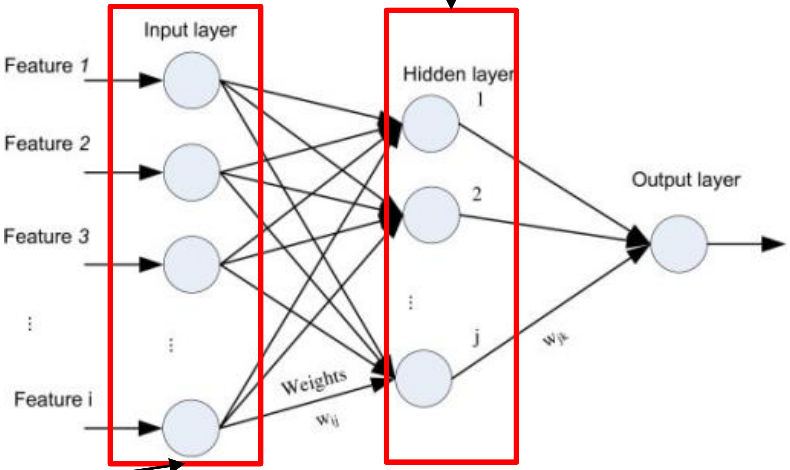
12288 caractéristiques

1000 x 1000

1000 x 1000 x 3

3 millions de caractéristiques

1 000 neurones



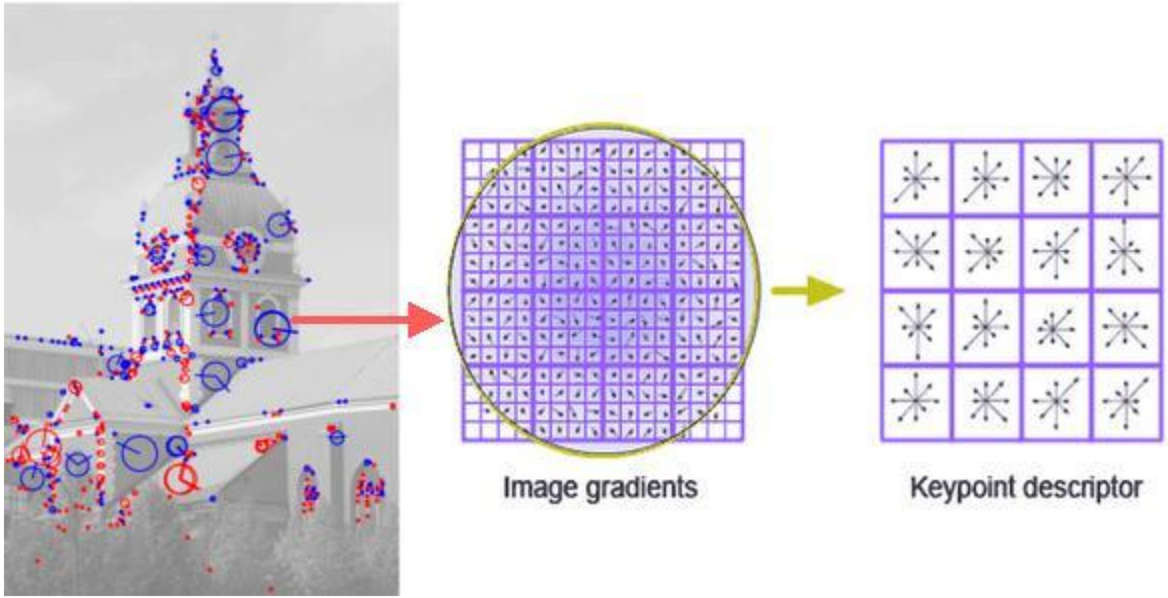
$W^{[1]}$ = 3 milliards de paramètres

Deep learning par la pratique

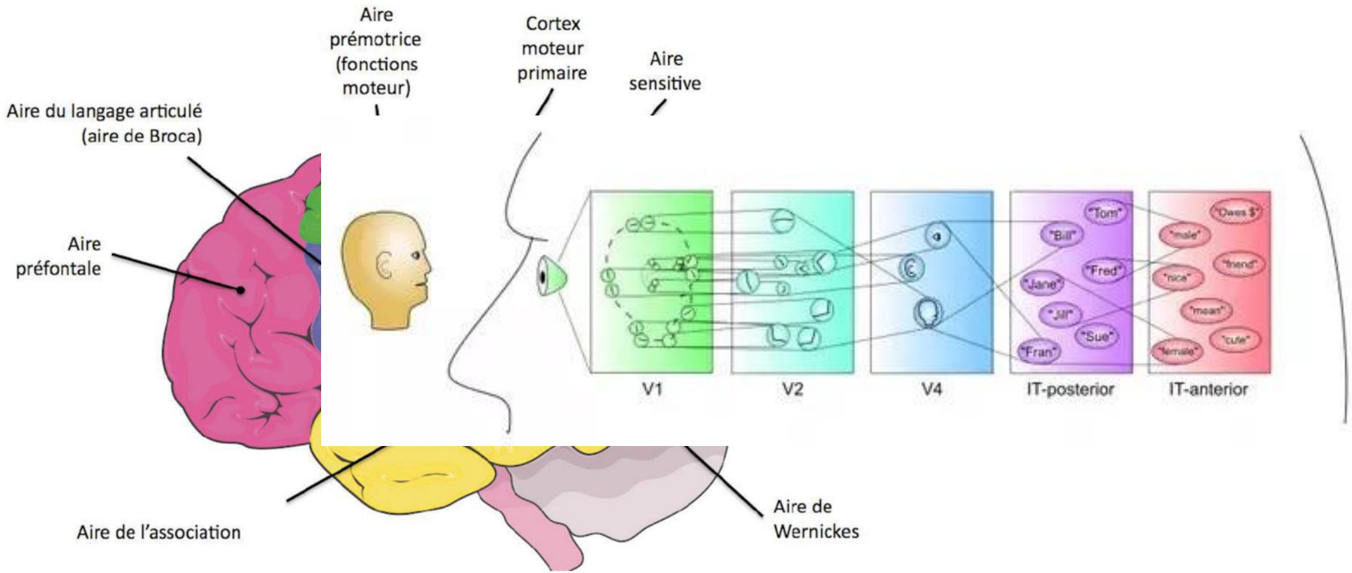
Leçon 2 : Extraction de caractéristiques

Présenté par **Morgan Gautherot**

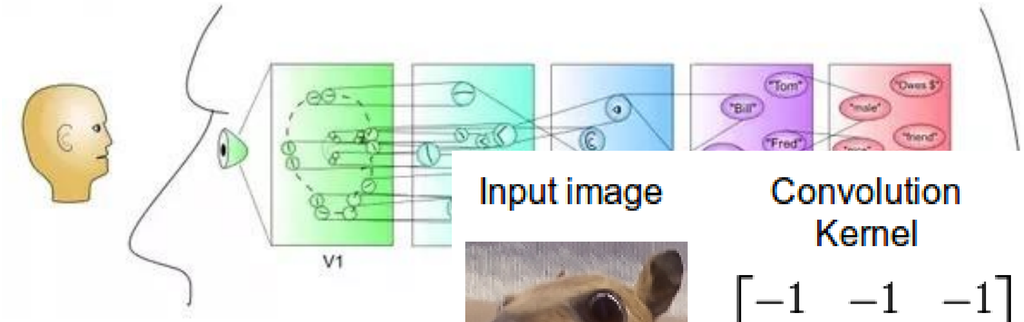
Extraction de caractéristiques

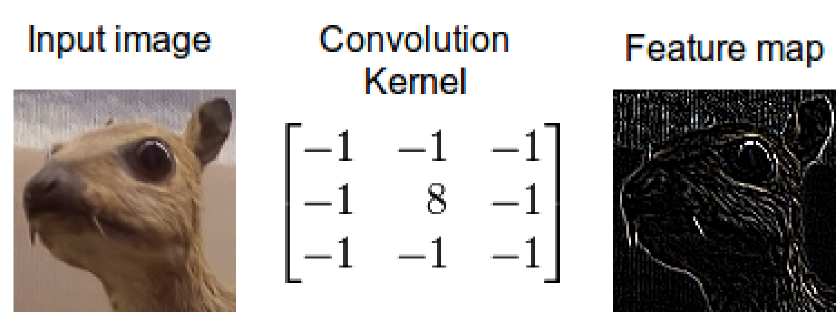


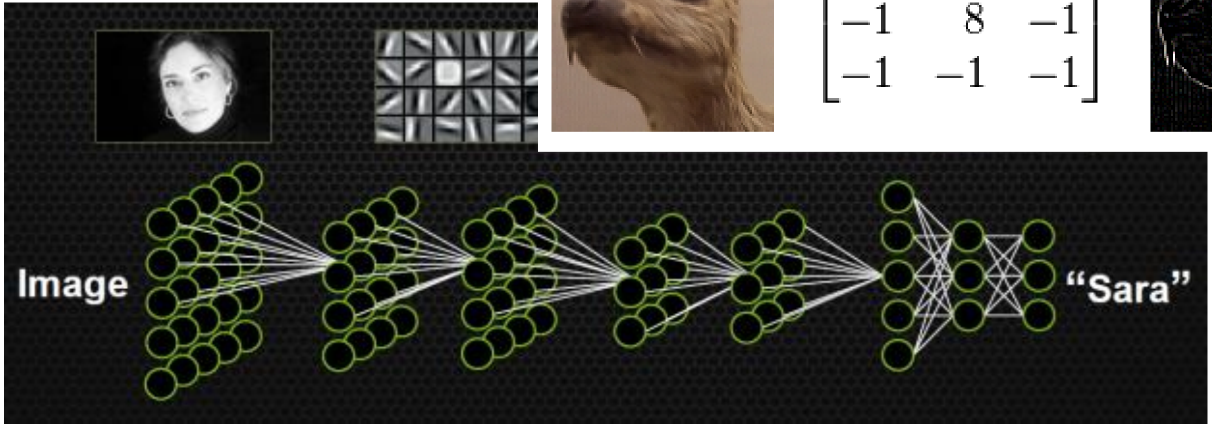
Comment voyons nous ?



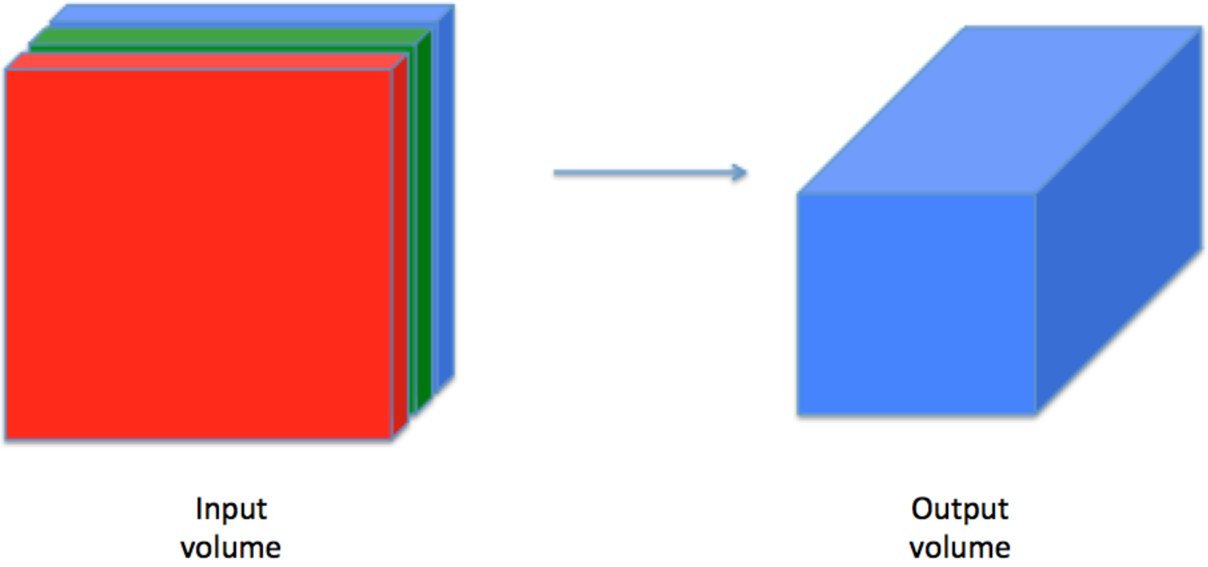
De la biologie à l'informatique







Représentation vectorielle de l’image d’entrée



Deep learning par la pratique

Leçon 3 : La convolution

Présenté par **Morgan Gautherot**

Les filtres

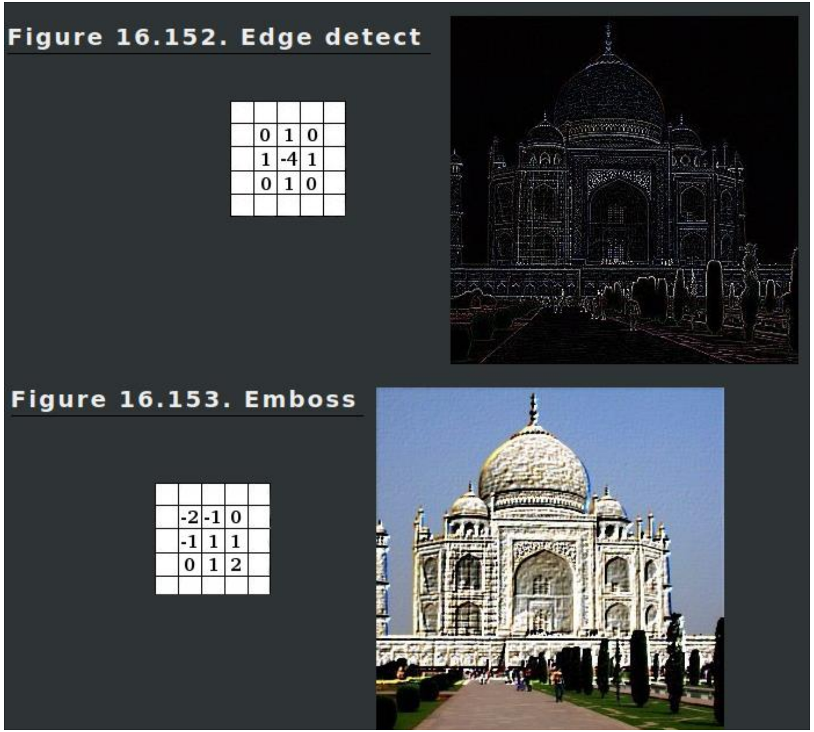


Figure 16.152. Edge detect
0 1 0
1 -4 1
0 1 0
Figure 16.153. Emboss
-2 -1 0
-1 1 1
0 1 2

Et mathématiquement ?

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

(6, 6)

Convolution

*

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

1	0	-1
1	0	-1
1	0	-1

(3, 3)

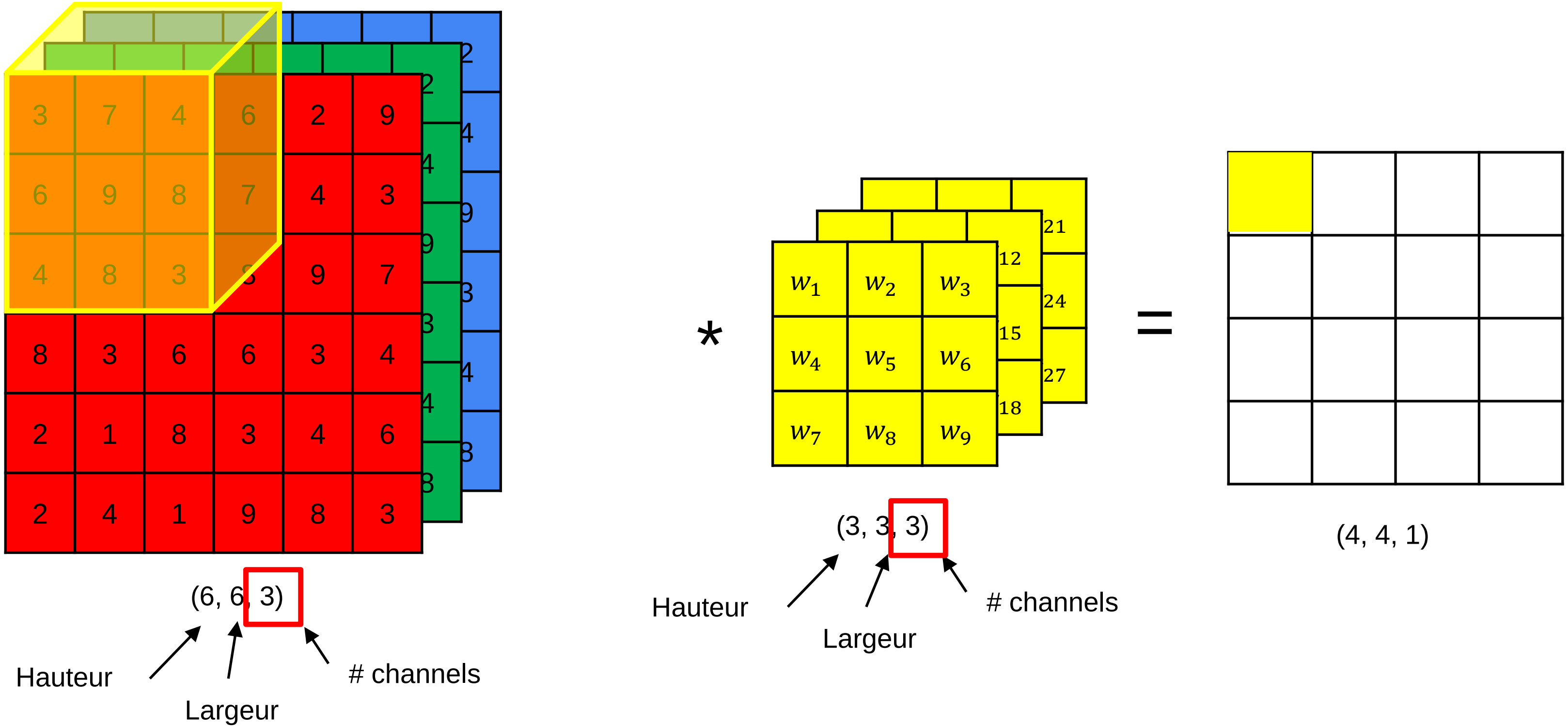
Filtre

=

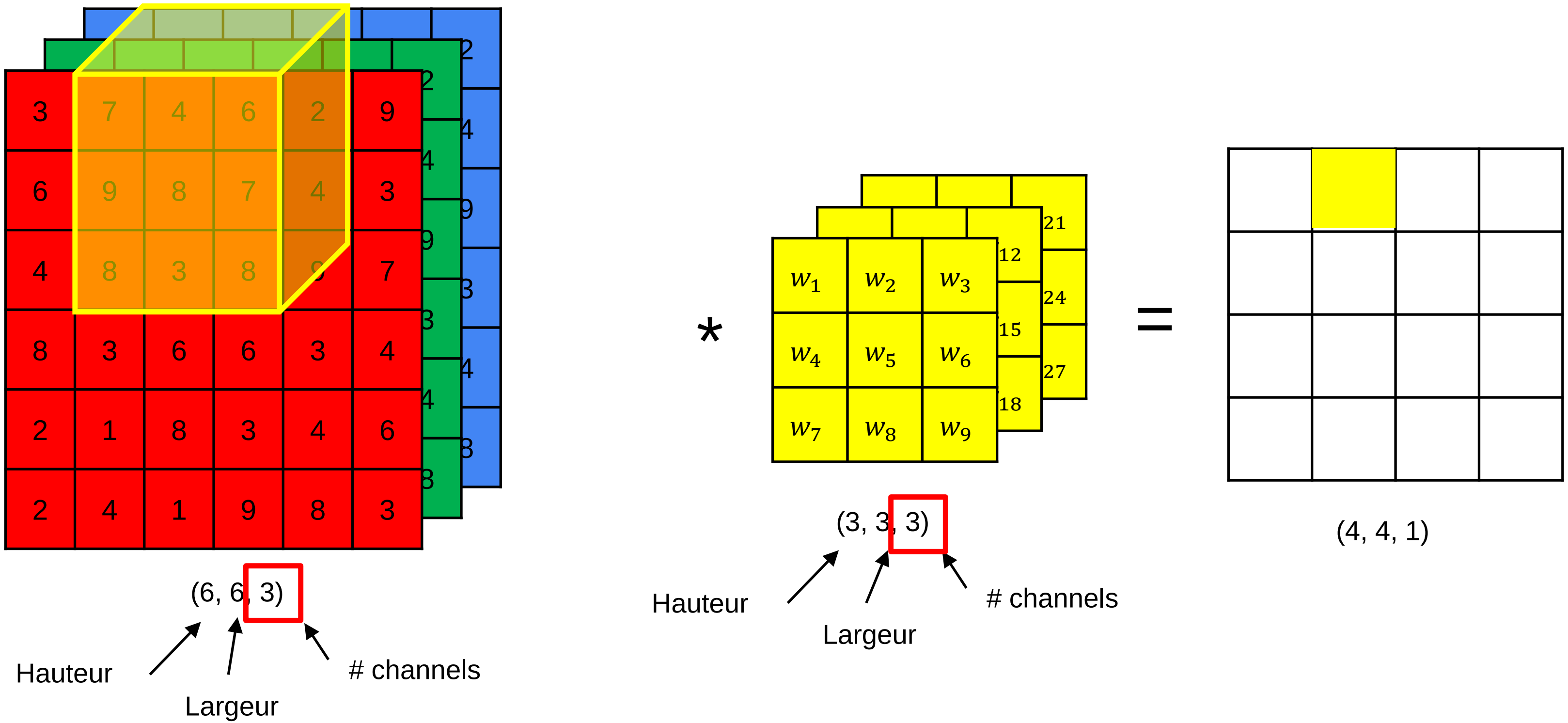
-5	-4	0	8
-10	-2	2	3
0	-2	-4	-7
-3	-2	-3	-16

(4, 4)

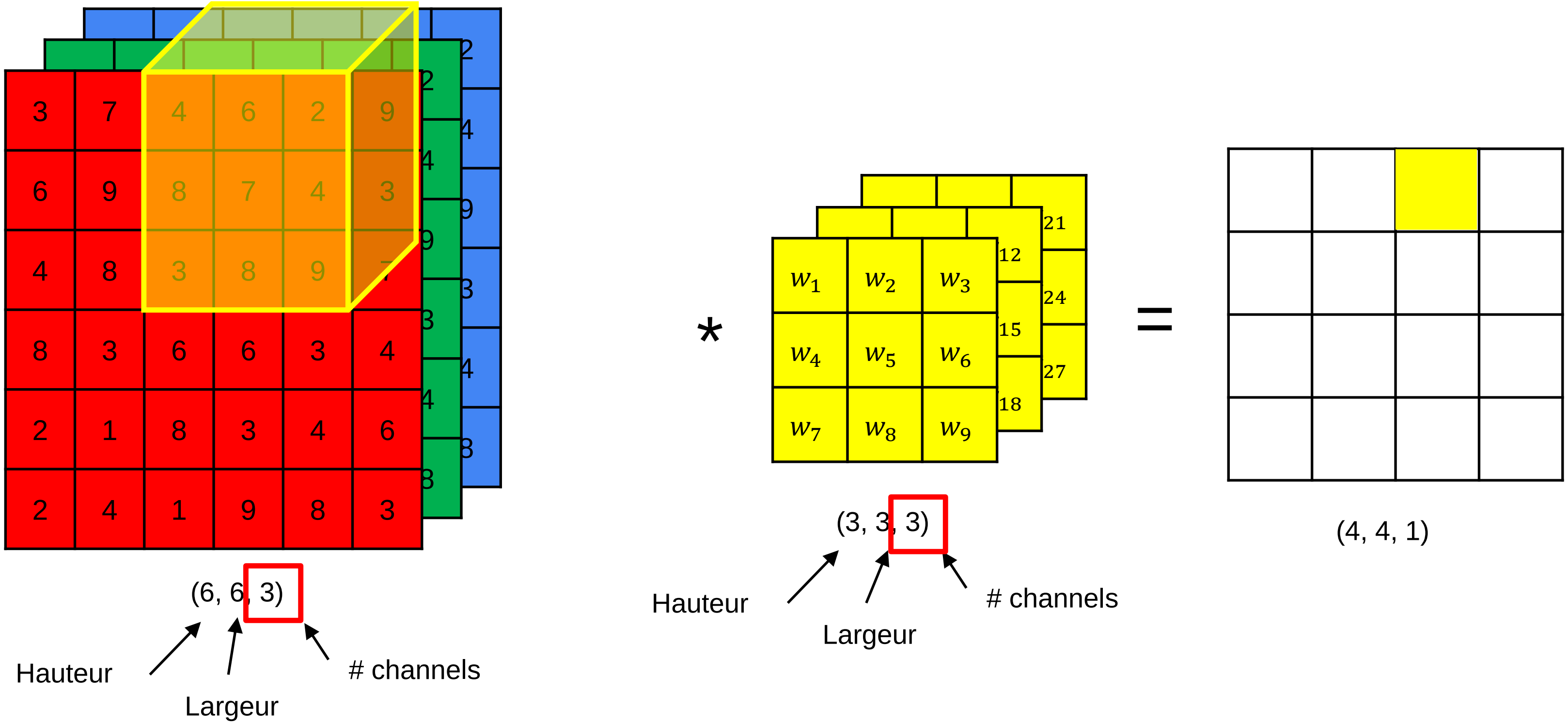
Convolution pour les images RGB



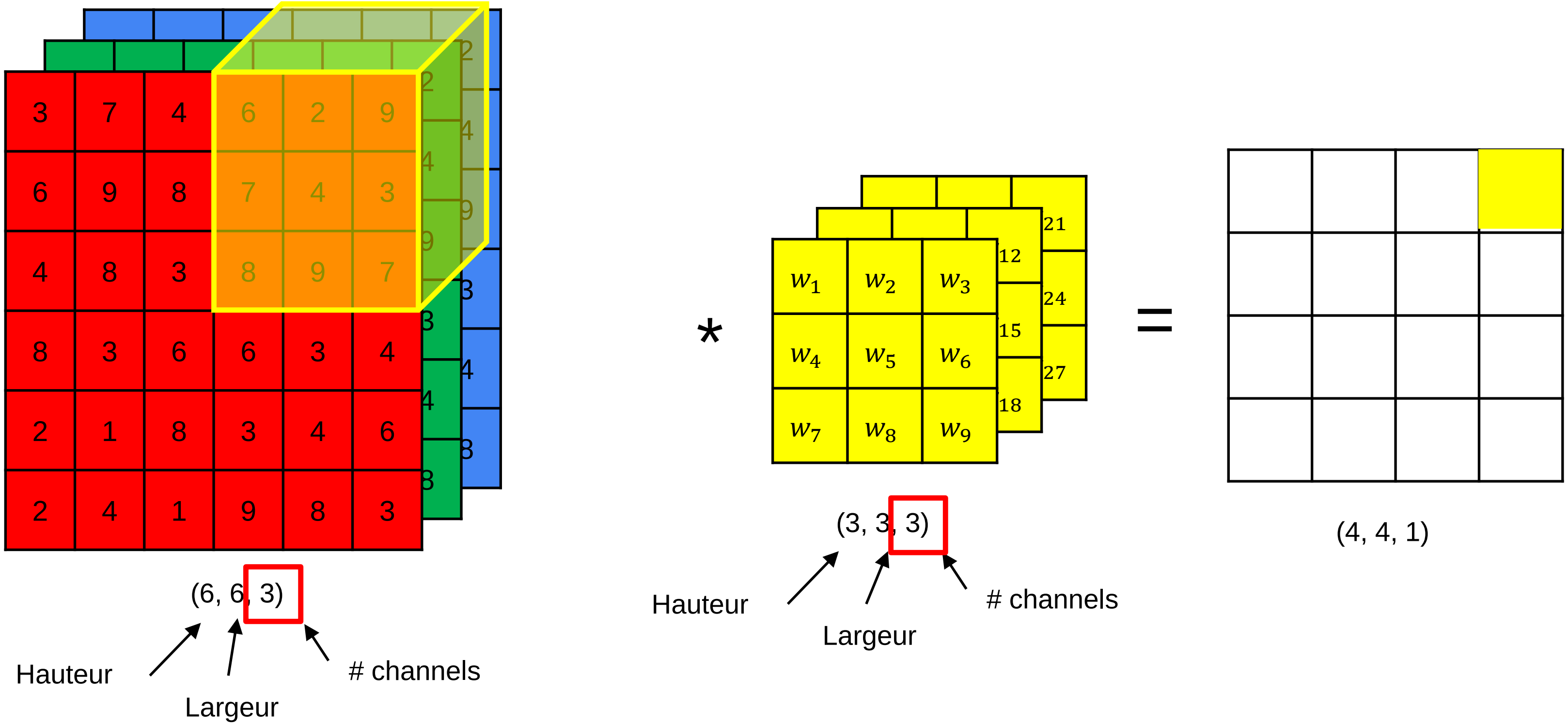
Convolution pour les images RGB



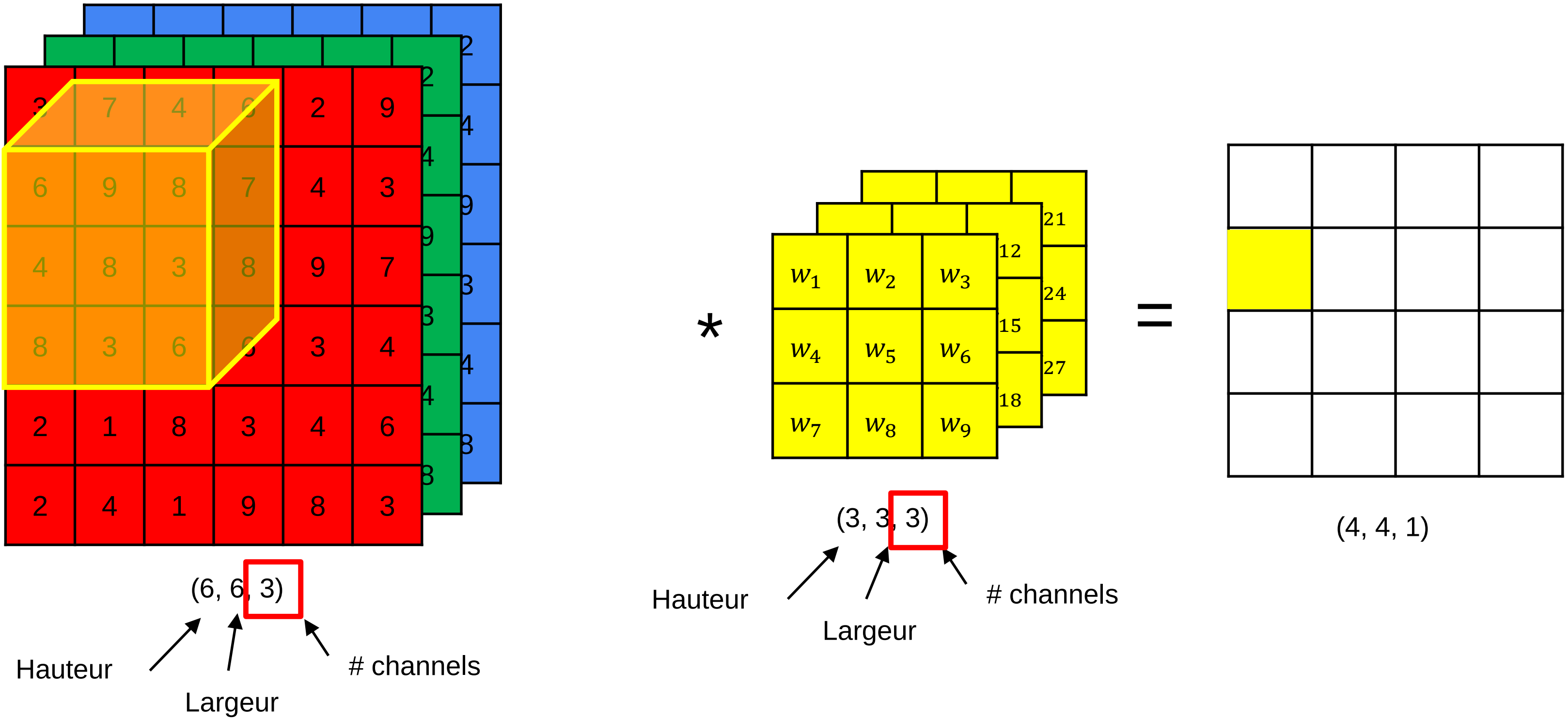
Convolution pour les images RGB



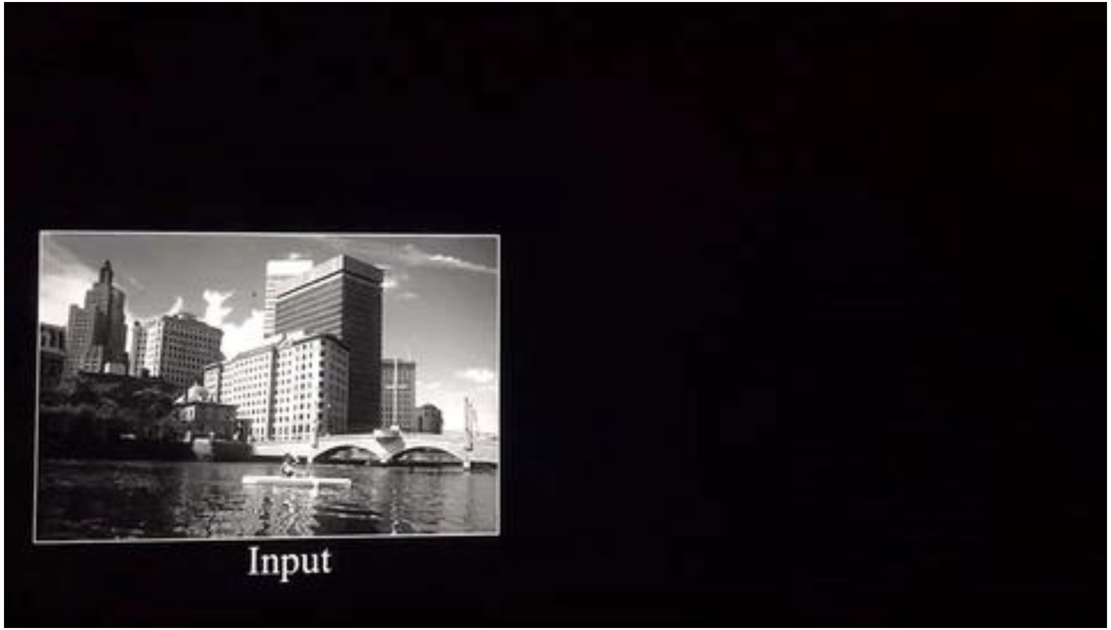
Convolution pour les images RGB



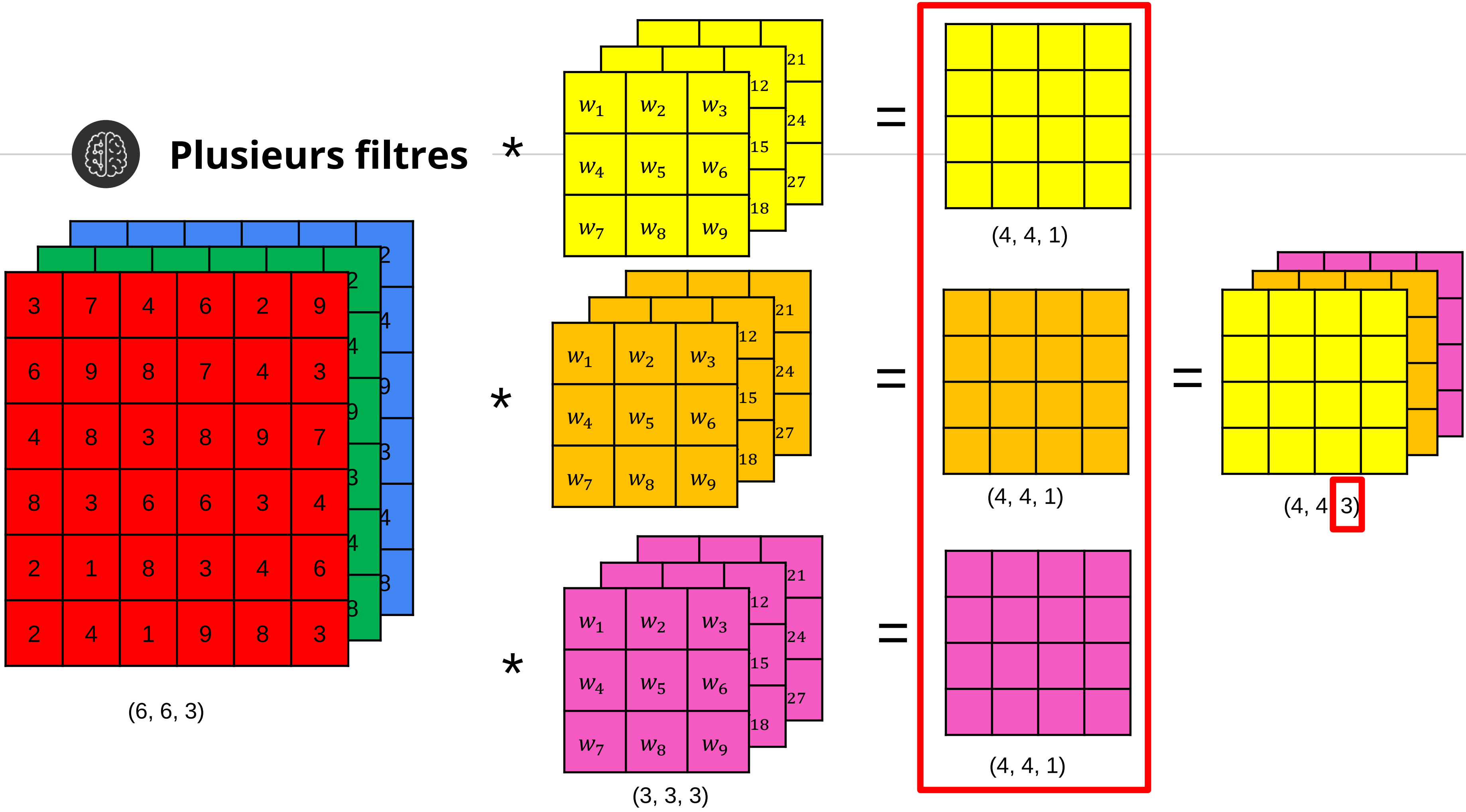
Convolution pour les images RGB



Plusieurs filtres



Plusieurs filtres

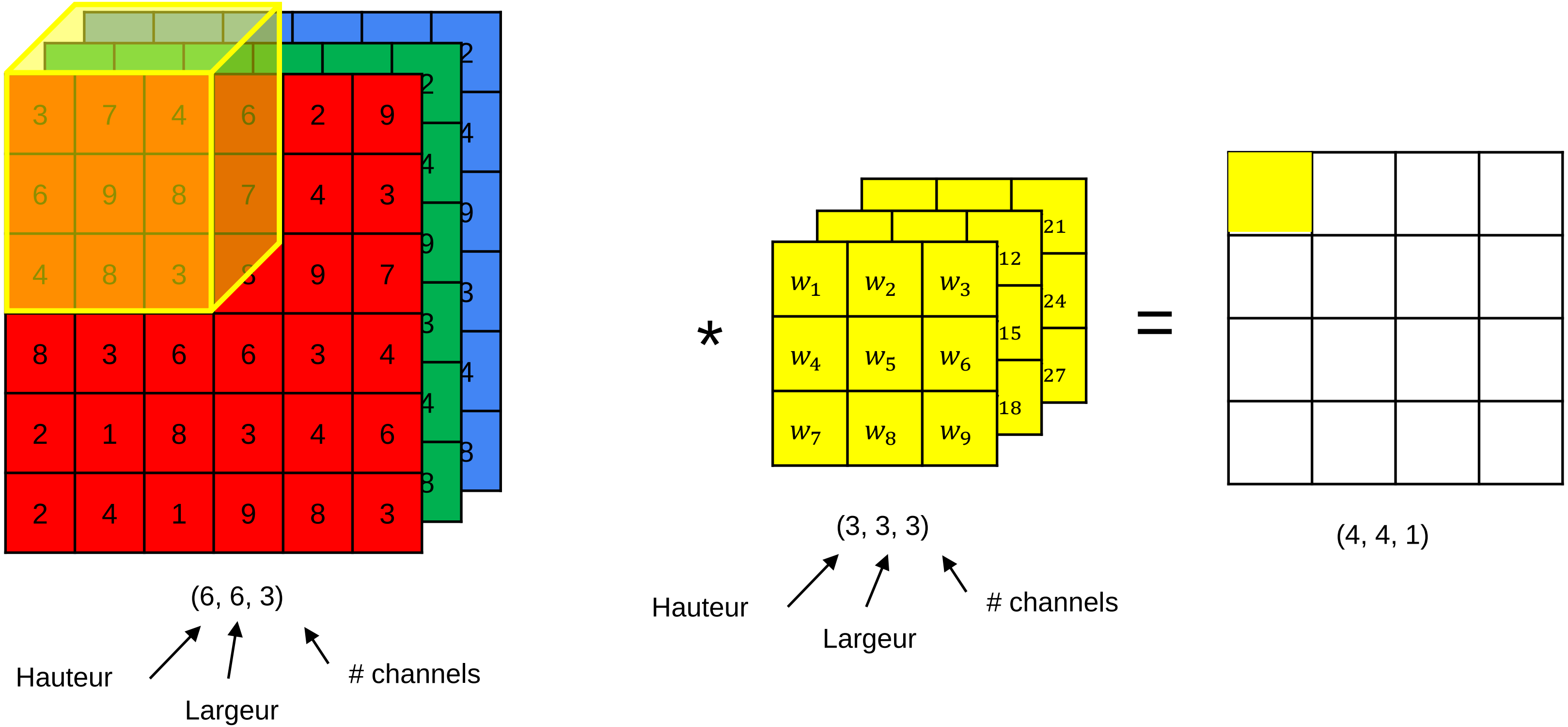


Deep learning par la pratique

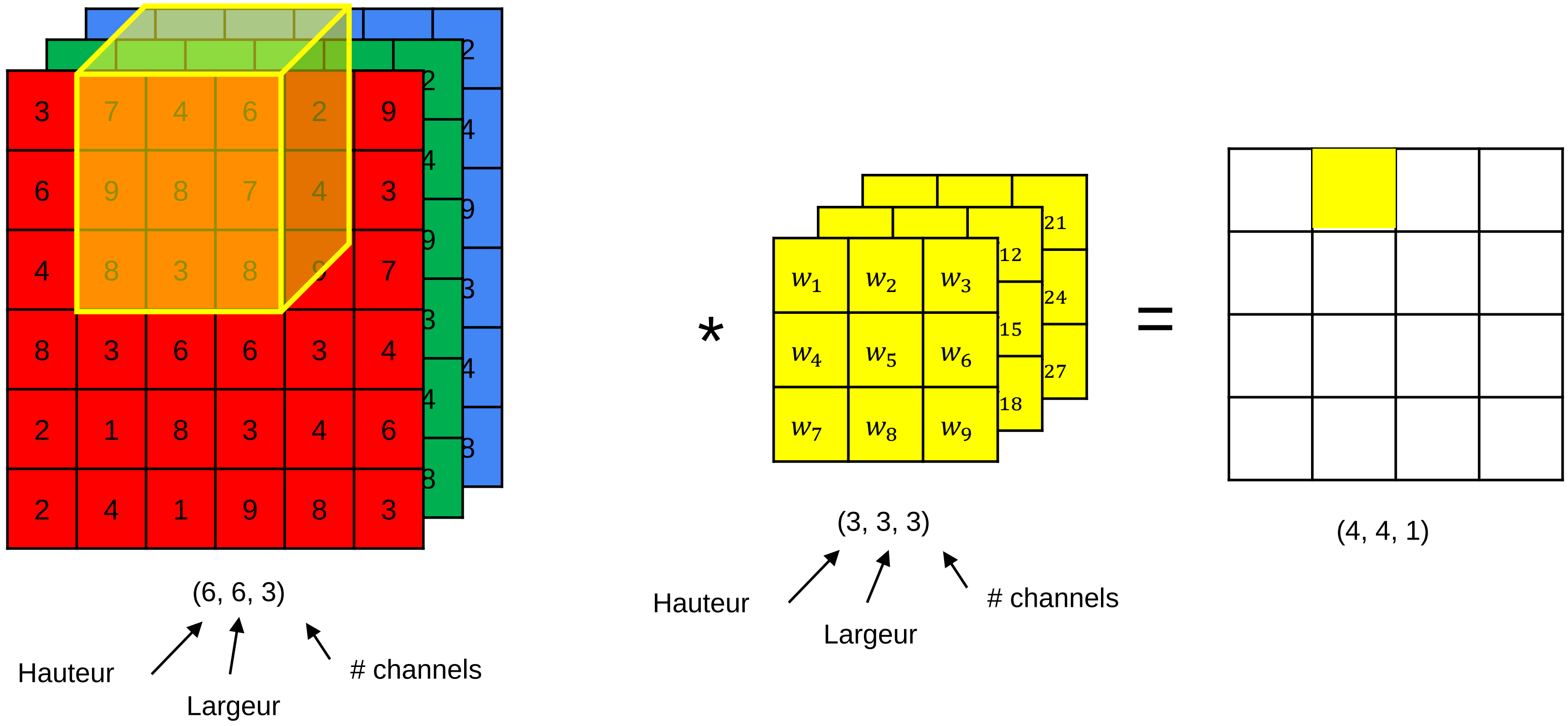
Leçon 4 : La convolution informatiquement

Présenté par **Morgan Gautherot**

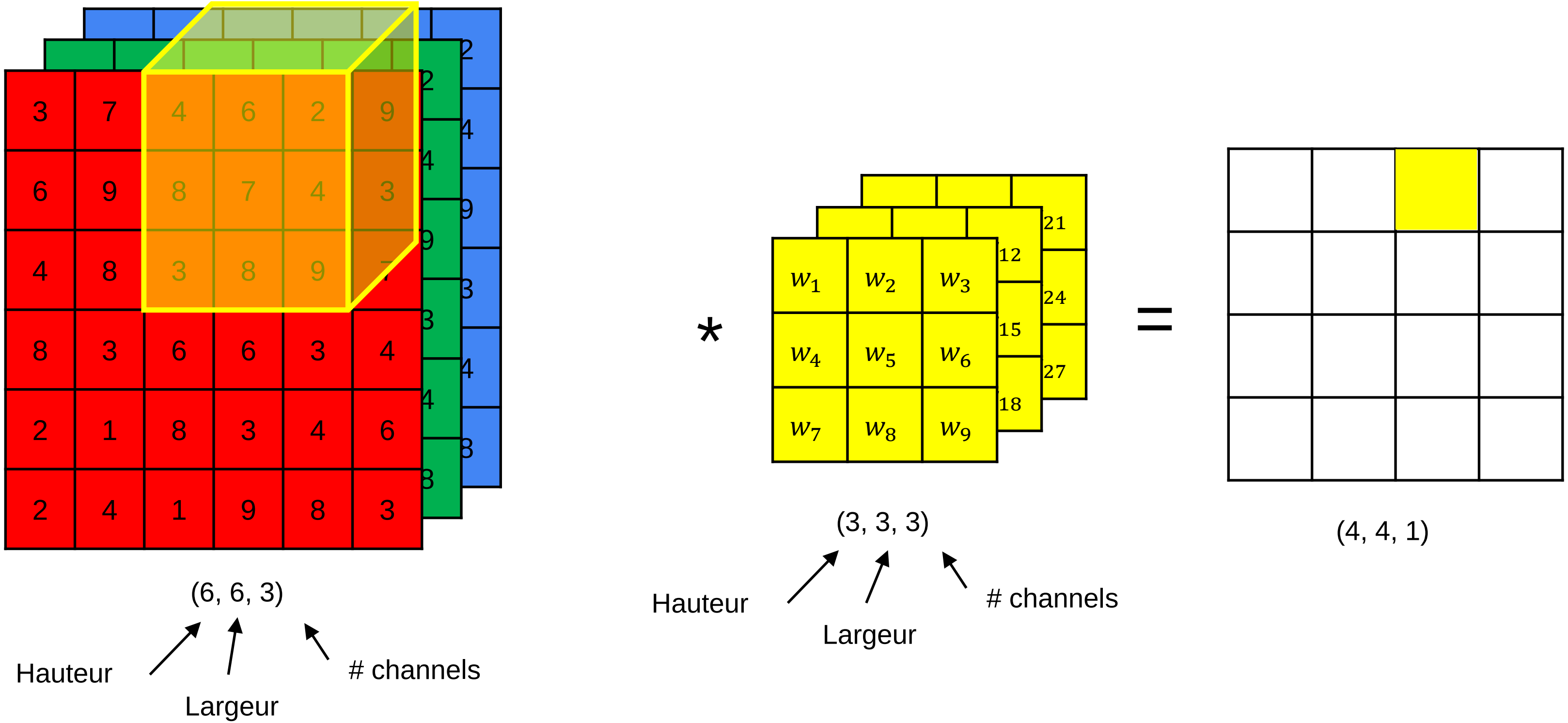
Convolution pour les images RGB



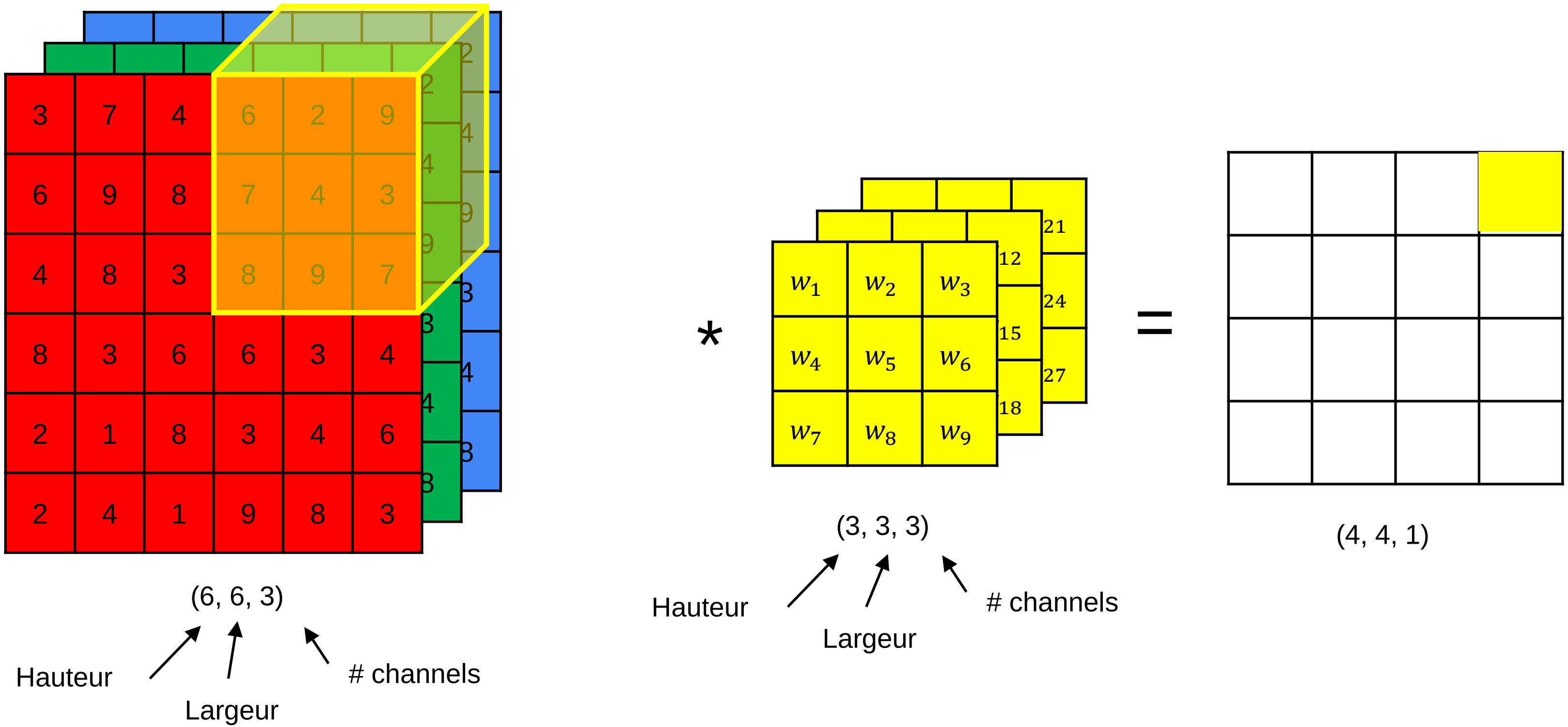
Convolution pour les images RGB



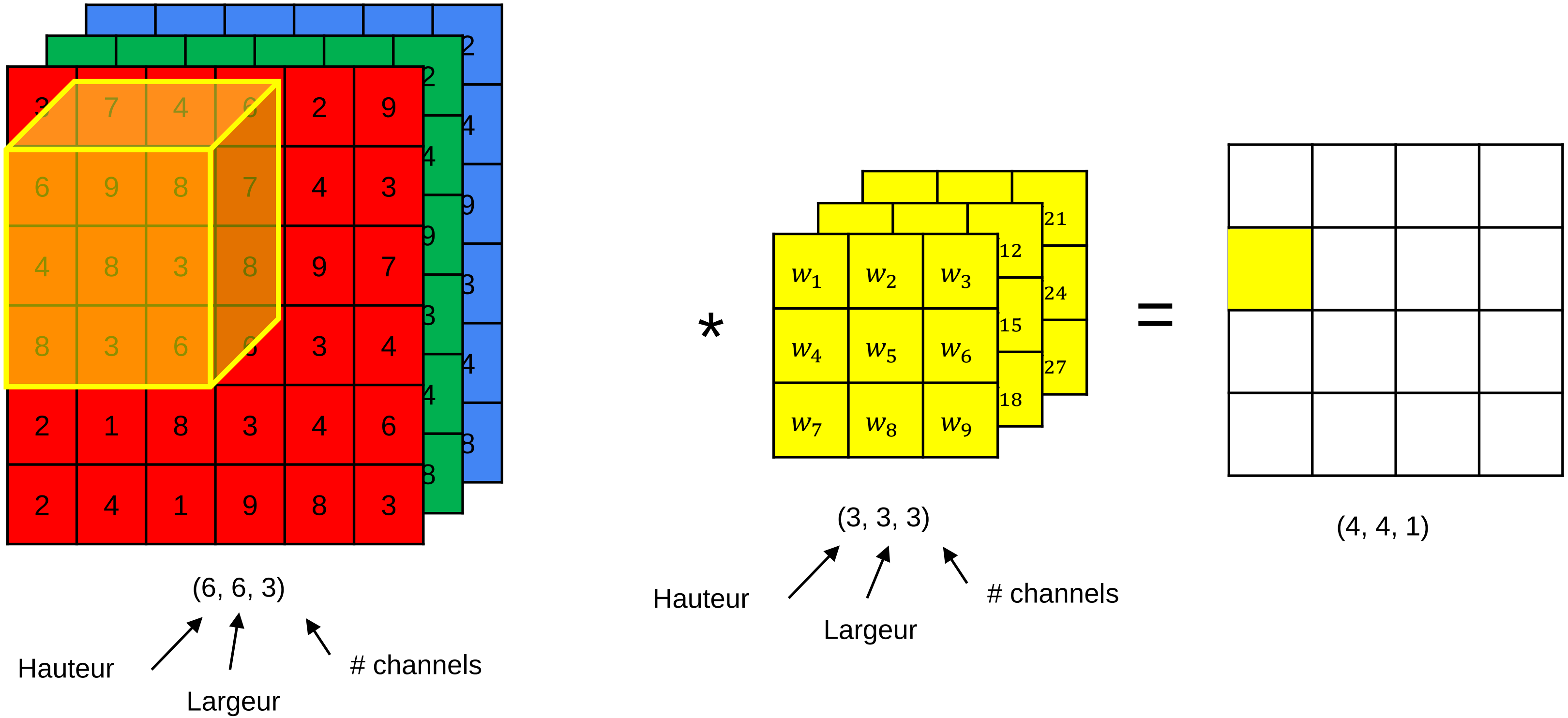
Convolution pour les images RGB



Convolution pour les images RGB



Convolution pour les images RGB



Convolution

$$\underset{4 \times 4}{\begin{bmatrix} 4 & 5 & 8 & 7 \\ 1 & 8 & 8 & 8 \\ 3 & 6 & 6 & 4 \\ 6 & 5 & 7 & 8 \end{bmatrix}} * \underset{3 \times 3}{\begin{bmatrix} 1 & 4 & 1 \\ 1 & 4 & 3 \\ 3 & 3 & 1 \end{bmatrix}} = \underset{2 \times 2}{\begin{bmatrix} 122 & 148 \\ 126 & 134 \end{bmatrix}}$$

Convolutional Matrix

1	4	1
1	4	3
3	3	1

3 x 3

=

1	4	1	0	1	4	3	0	3	3	1	0	0	0	0	0
0	1	4	1	0	1	4	3	0	3	3	1	0	0	0	0
0	0	0	0	1	4	1	0	1	4	3	0	3	3	1	0
0	0	0	0	0	1	4	1	0	1	4	3	0	3	3	1

4 x 16

4	5	8	7
1	8	8	8
3	6	6	4
6	5	7	8

4 x 4

=

4
5
8
7
1
8
8
8
3
6
6
4
6
5
7
8

16 x 1

Convolutional Matrix

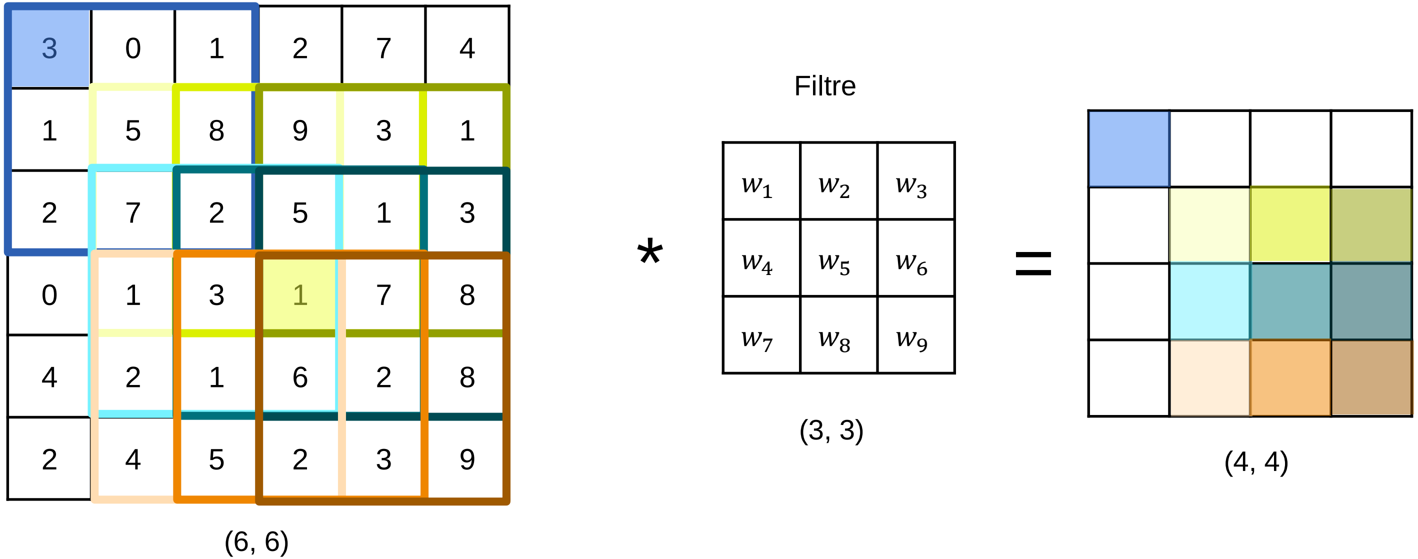
$$\underset{4 \times 16}{\begin{bmatrix} 1 & 4 & 1 & 0 & 1 & 4 & 3 & 0 & 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 1 & 4 & 3 & 0 & 3 & 3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 1 & 4 & 3 & 0 & 3 & 3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 1 & 4 & 3 & 0 & 3 & 3 & 1 \end{bmatrix}} \cdot \underset{16 \times 1}{\begin{bmatrix} 4 \\ 5 \\ 8 \\ 7 \\ 1 \\ 8 \\ 8 \\ 8 \\ 3 \\ 6 \\ 6 \\ 4 \\ 6 \\ 5 \\ 7 \\ 8 \end{bmatrix}} = \underset{4 \times 1}{\begin{bmatrix} 122 \\ 148 \\ 126 \\ 134 \end{bmatrix}} = \underset{2 \times 2}{\begin{bmatrix} 122 & 148 \\ 126 & 134 \end{bmatrix}}$$

Deep learning par la pratique

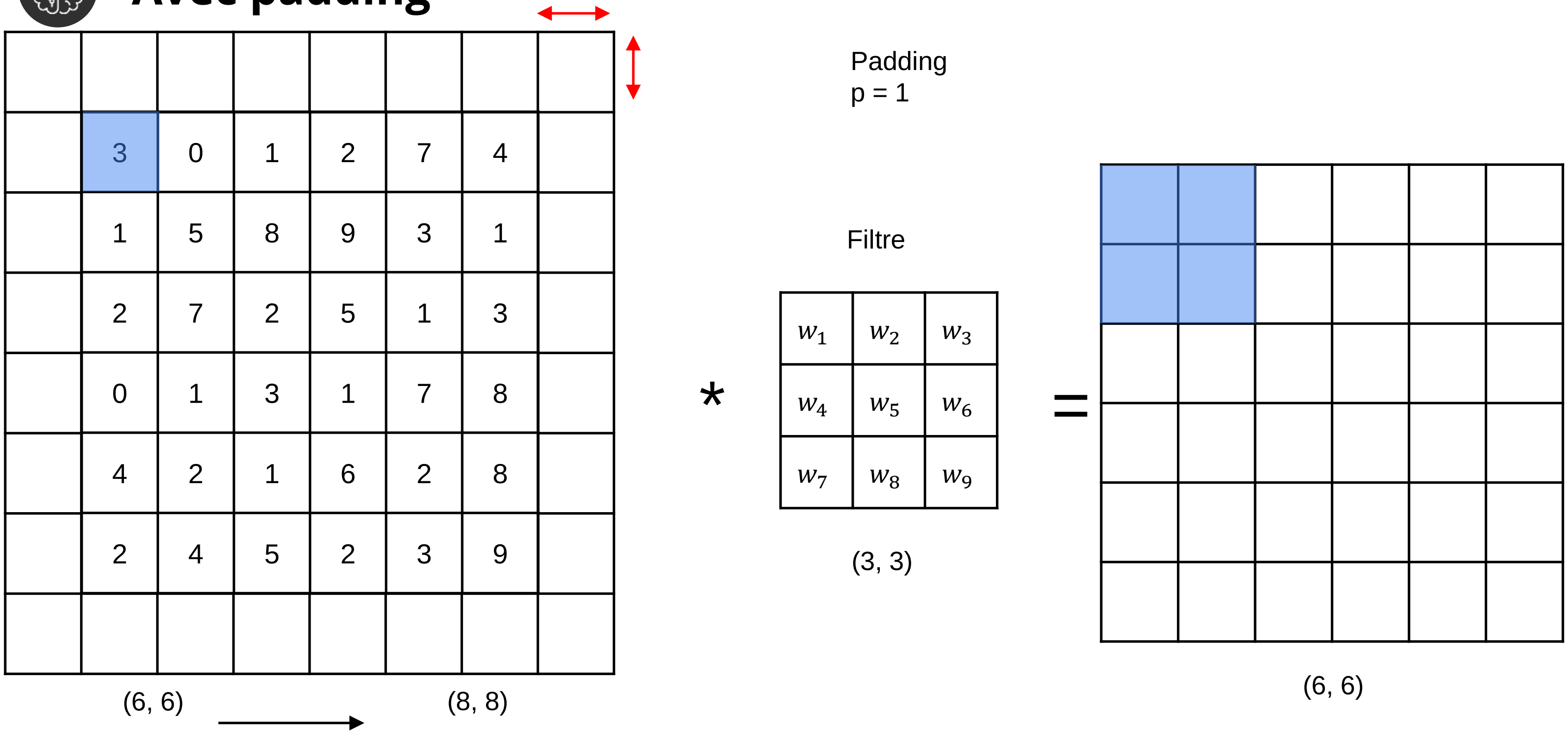
Leçon 5 : La convolution

Présenté par **Morgan Gautherot**

Sans padding



Avec padding



Stride

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

(7, 7)

Stride
s = 2

*

filter

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

(3, 3)

=

91	100	83
69	91	127
44	72	74

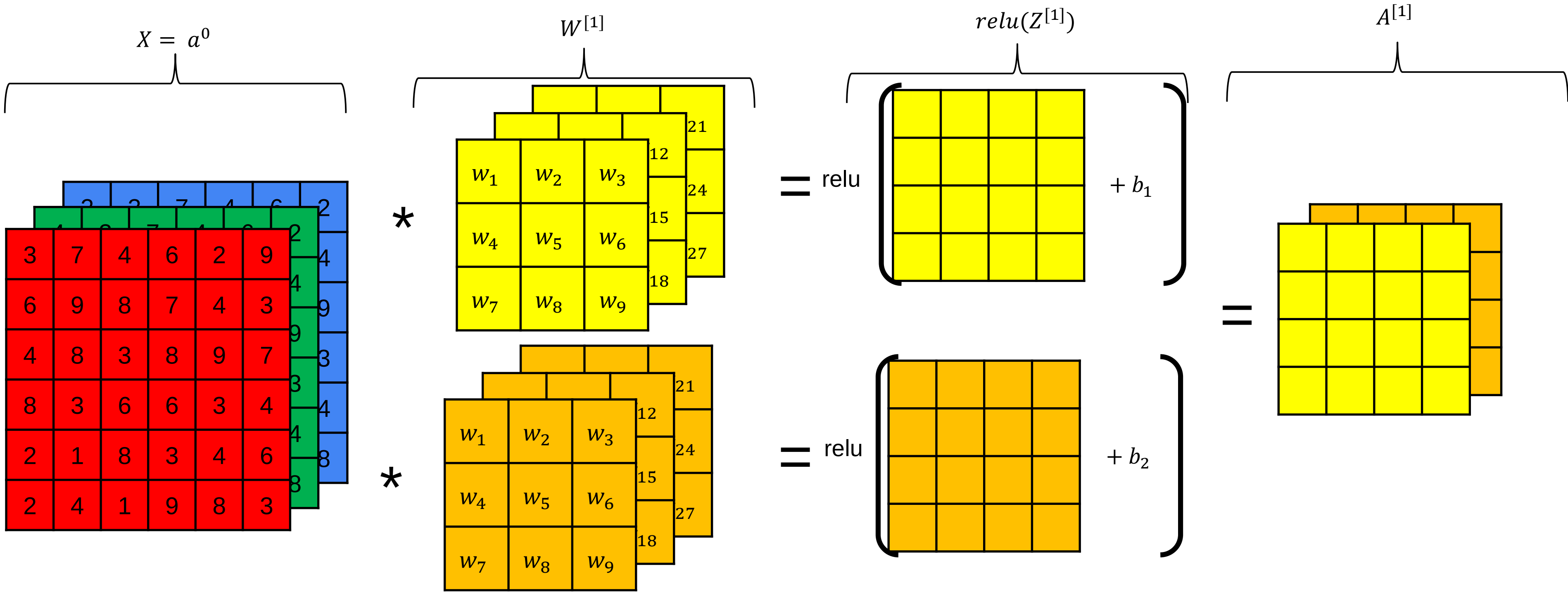
(3, 3)

Deep learning par la pratique

Leçon 6 : La convolution

Présenté par **Morgan Gautherot**

Une couche de convolution



Deep learning par la pratique

Leçon 7 : Le pooling

Présenté par **Morgan Gautherot**

Maintenir de l’invariance

Entrée Fenêtre Sortie

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Max = 2

↑

Ne change pas

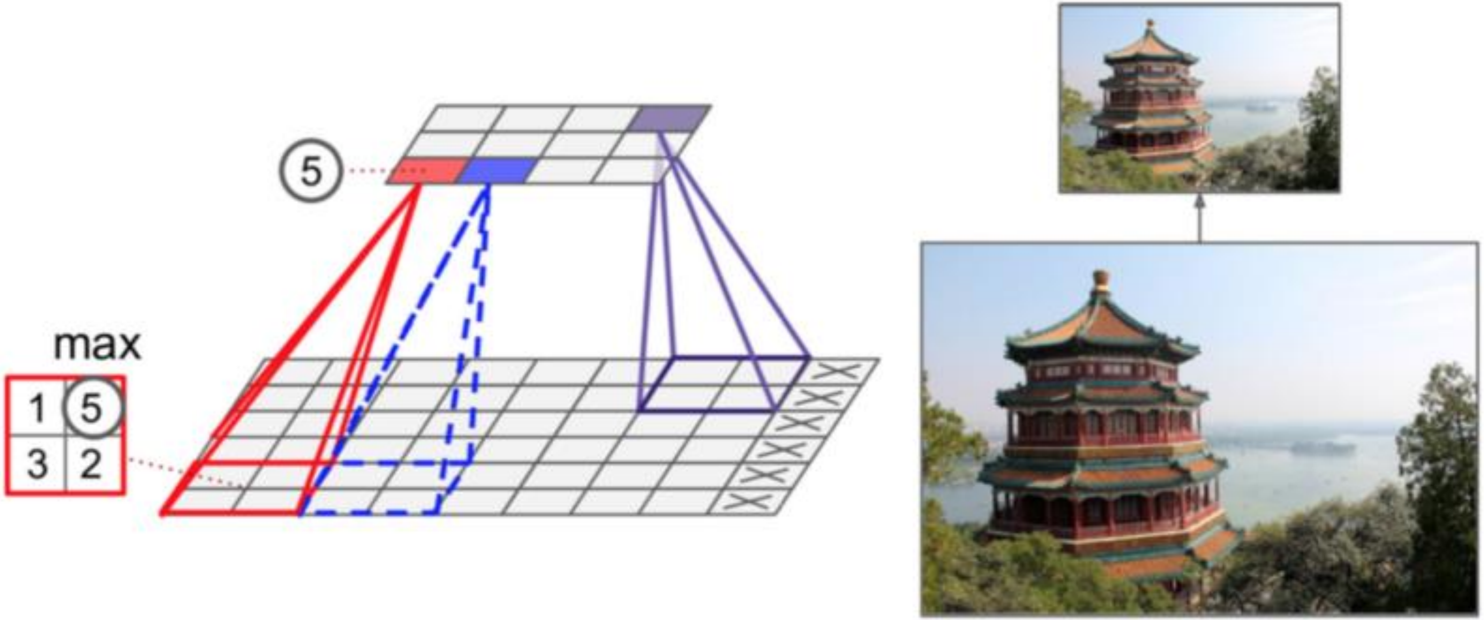
↓

Max = 2

Entrée Fenêtre Sortie

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Application sur une image



Pooling layer: Max pooling

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

(6, 6)

Hyperparameters:

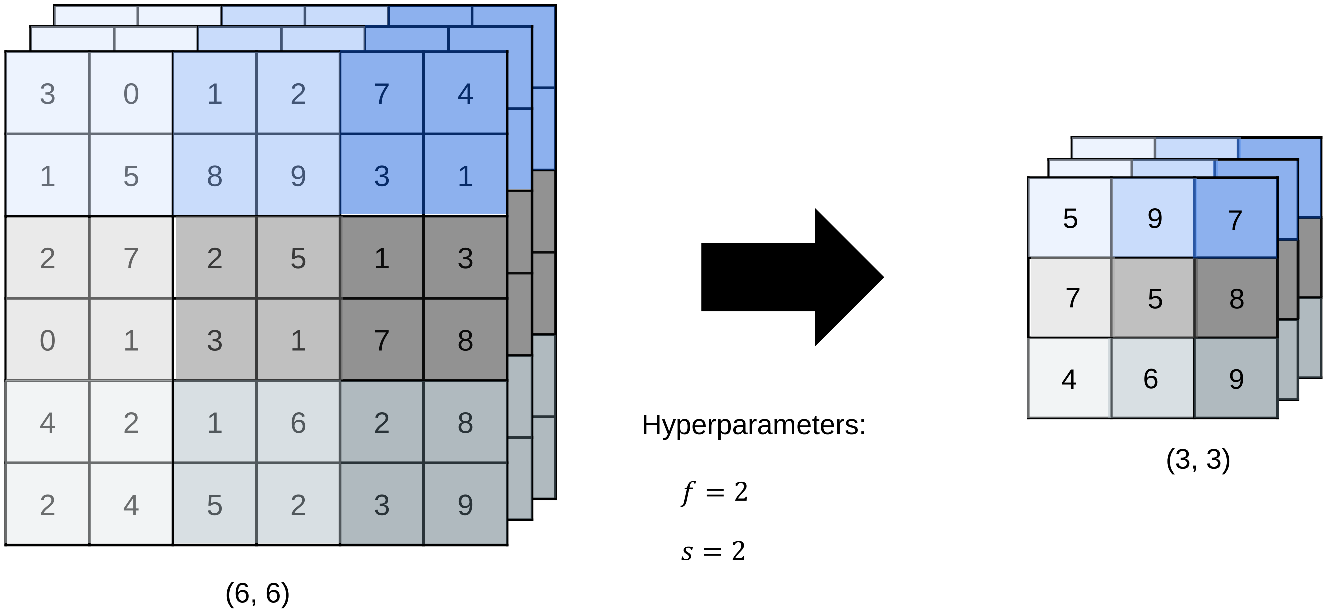
$f = 2$

$s = 2$

5	9	7
7	5	8
4	6	9

(3, 3)

Pooling layer: Max pooling



Pooling layer: Average pooling

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

(6, 6)

Hyperparameters:

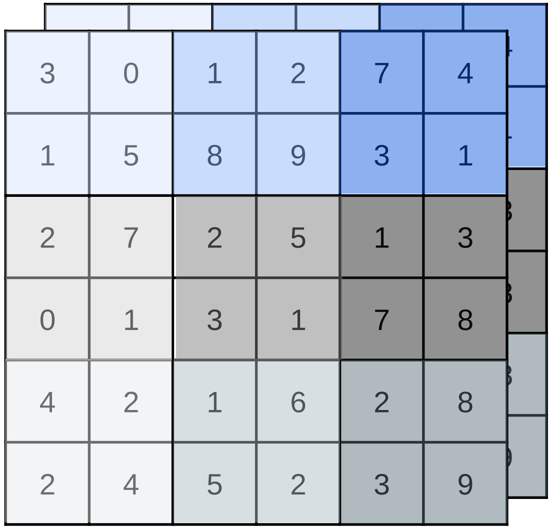
$f = 2$

$s = 2$

2.25	5	3.75
2.5	2.75	4.75
3	3.5	5.5

(3, 3)

Pooling layer: Average pooling

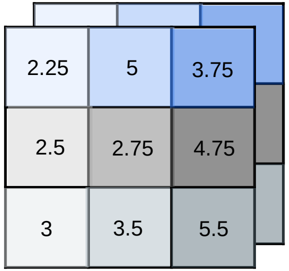


(6, 6, 2)

Hyperparameters:

$f = 2$

$s = 2$



(3, 3, 2)

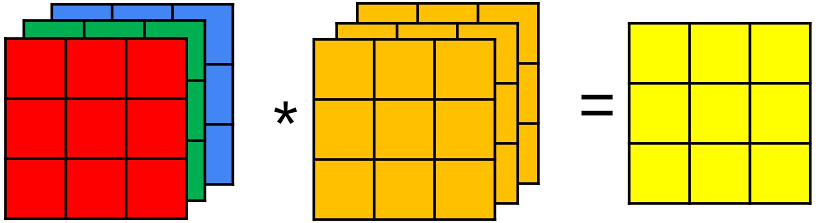
Deep learning par la pratique

Leçon 8 : Les trois blocs du CNN

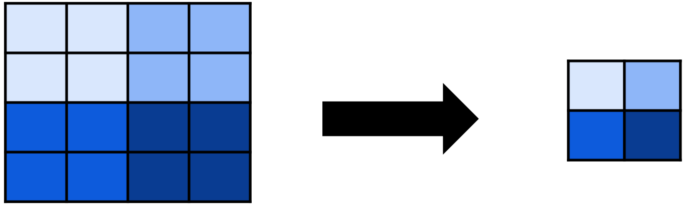
Présenté par **Morgan Gautherot**

Les trois blocs du CNN

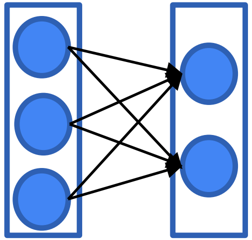
Convolutional bloc



Pooling bloc

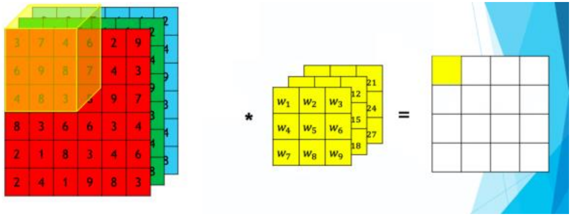


Fully connected bloc

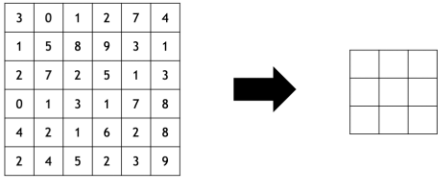


Les trois blocs du CNN

Convolution

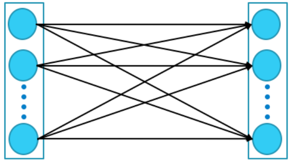


Pooling



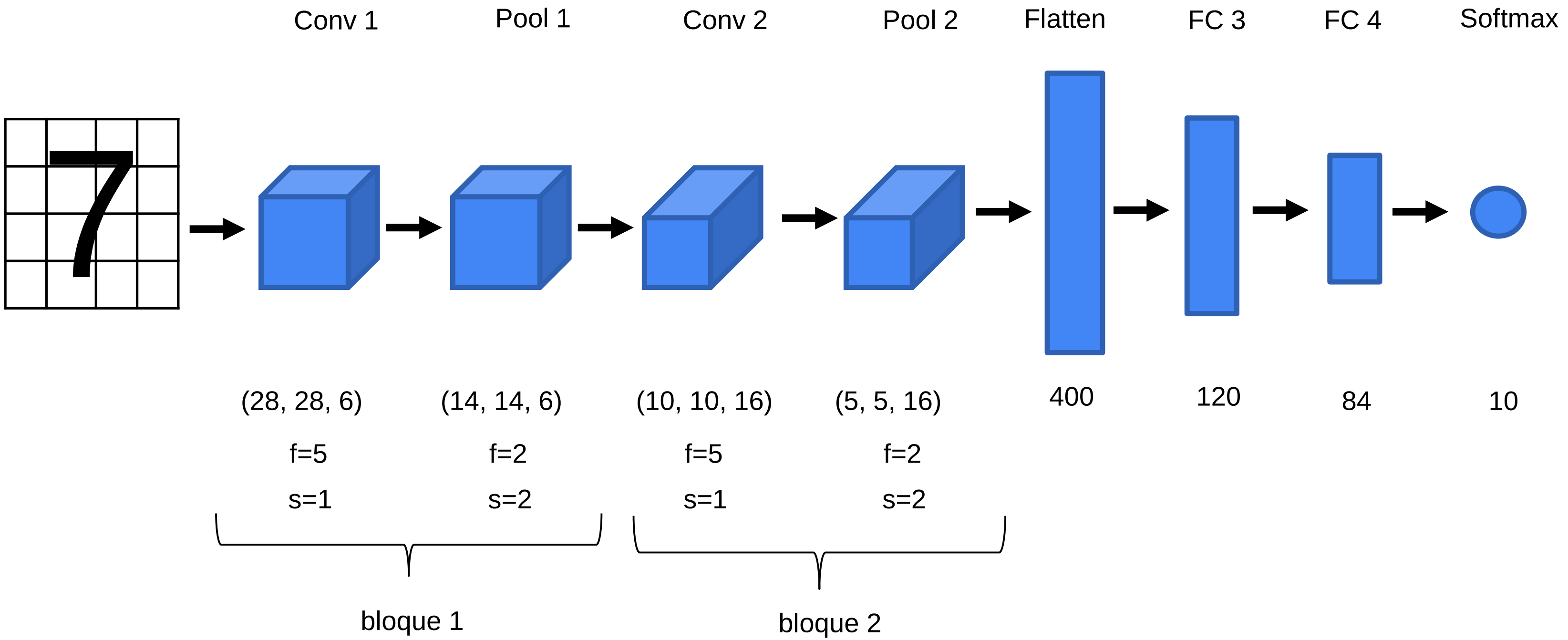
Flatten

Couche dense



LeNet-5

≈ 60k paramètres



LeNet-5

	Activation shape	Activation size	# parameters
Input:	(32, 32, 3)	3 072	0
CONV1 (f=5, s=1)	(28,28, 8)	6 272	208
POOL1	(14, 14, 8)	1 568	0
CONV2 (f=5, s=1)	(10, 10, 46)	1 600	416
POOL2	(5, 5, 16)	400	0
FC3	(120, 1)	120	48 001
FC4	(84, 1)	84	10 081
Softmax	(10, 1)	10	841