

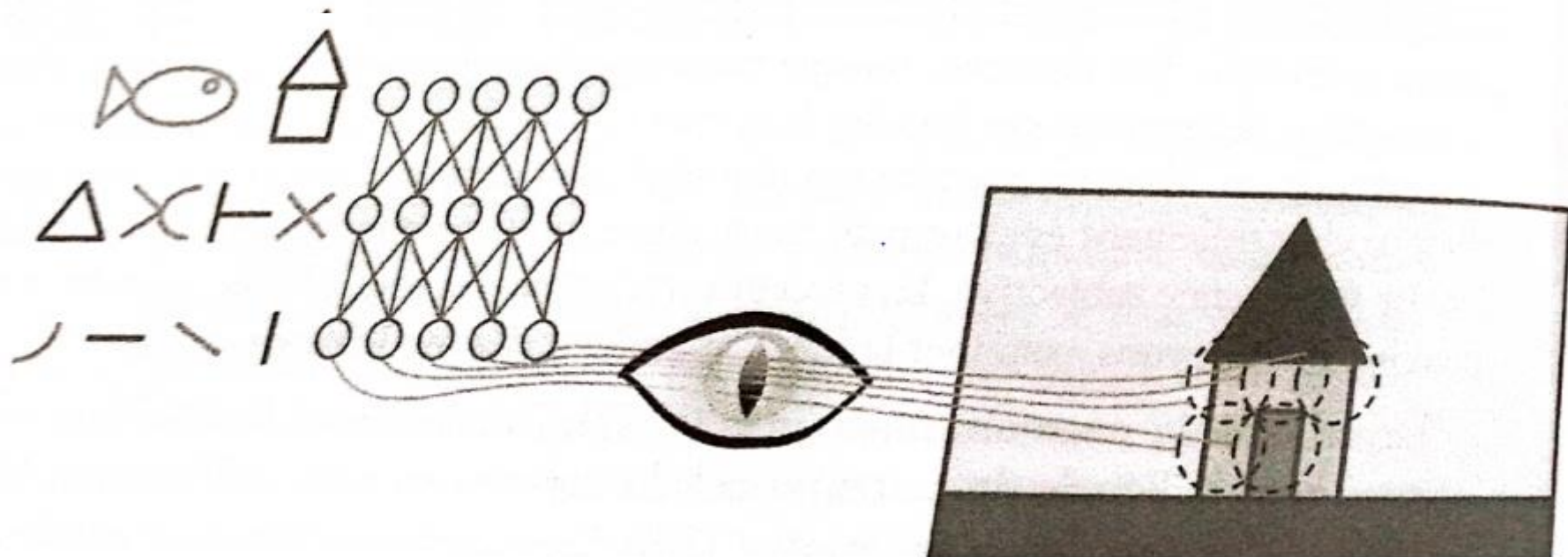
Réseaux de convolution CNN

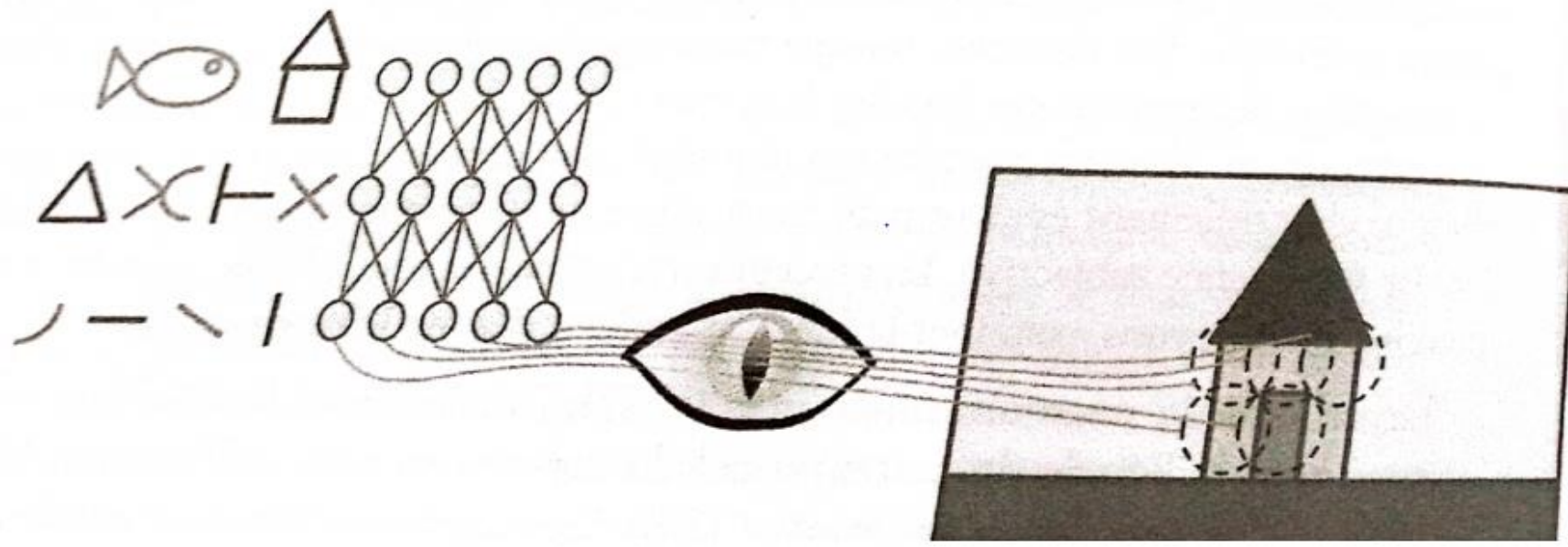
Sommaire

- Introduction : un peu d'histoire ...
- Couche de convolution
- Filtre (kernel ou matrice de convolution)
- Stride
- Padding
- Pooling
- Classification-Fully Connected Layer
- Des liens ...

Un peu d'histoire

En 1958⁷⁶ et en 1959⁷⁷, David H. Hubel et Torsten Wiesel ont mené une série d'expériences sur des chats (et, quelques années plus tard, sur des singes⁷⁸), apportant des informations essentielles sur la structure du cortex visuel (en 1981, ils ont reçu le prix Nobel de physiologie ou médecine pour leurs travaux). Ils ont notamment montré que de nombreux neurones du cortex visuel ont un petit *champ récepteur local* et qu'ils réagissent donc uniquement à un stimulus visuel qui se trouve dans une région limitée du champ visuel (voir la figure sur laquelle les champs récepteurs locaux de cinq neurones sont représentés par les cercles en pointillés).



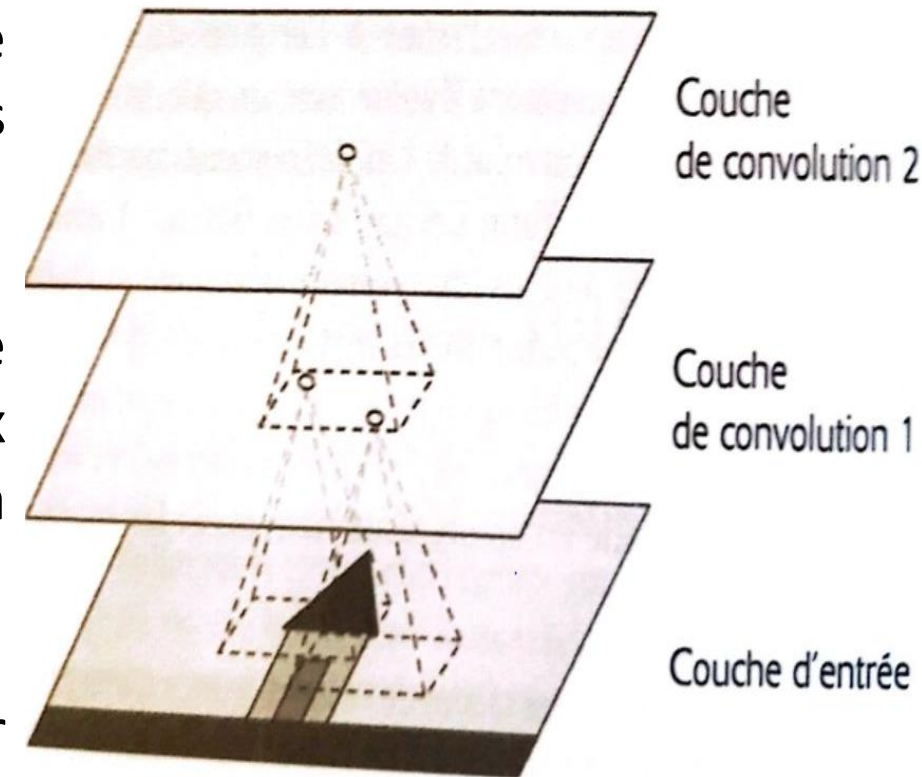


Les champs récepteurs des différents neurones peuvent se chevaucher et ils couvrent ensemble l'intégralité du champ visuel. Ils ont également montré que certains neurones réagissent uniquement aux images de lignes horizontales, tandis que d'autres réagissent uniquement aux lignes ayant d'autres orientations (deux neurones peuvent avoir le même champ récepteur mais réagir à des orientations de lignes différentes). Ils ont remarqué que certains neurones ont des champs récepteurs plus larges et qu'ils réagissent à des motifs plus complexes, correspondant à des combinaisons de motifs de plus bas niveau. Ces observations ont conduit à l'idée que les neurones de plus haut niveau se fondent sur la sortie des neurones voisins de plus bas niveau (sur la figure , chaque neurone est connecté uniquement à quelques neurones de la couche précédente). Cette architecture puissante est capable de détecter toutes sortes de motifs complexes dans n'importe quelle zone du champ visuel.

Pourquoi, pour les tâches de reconnaissance d'images, ne pas simplement utiliser un réseau de neurones profond classique, avec des couches intégralement connectées? Malheureusement, bien qu'un tel réseau convienne parfaitement aux petites images (par exemple, celles du jeu MNIST), il n'est pas adapté aux images plus grandes en raison de l'énorme quantité de paramètres qu'il exige. Par exemple, une image 100×100 est constituée de 10 000 pixels et, si la première couche comprend uniquement 1 000 neurones (ce qui limite déjà beaucoup la quantité d'informations transmises à la couche suivante), cela donne 10 millions de connexions. Et nous ne parlons que de la première couche. Les CNN résolvent ce problème en utilisant des couches partiellement connectées.

Couche de convolution

- La couche de convolution est le bloc de construction le plus important d'un CNN. Dans la première couche de convolution, les neurones ne sont pas connectés à chaque pixel de l'image d'entrée mais uniquement aux pixels dans leurs champs récepteur.
- A leur tour, les neurones de la deuxième couche de convolution sont chacun connecté uniquement aux neurones situés à l'intérieur d'un petit rectangle de la première couche.
- Cette architecture permet au réseau de se focaliser sur des caractéristiques de bas niveau dans la première couche cachée, puis de les assembler en caractéristiques de plus haut niveau dans la couche suivante



The Convolution Operation

Convolutional Operation means for a given input we re-estimate it as the weighted average of all the inputs around it.

We have some weights assigned to the neighbor values and we take the weighted sum of the neighbor values to estimate the value of the current input/pixel.

1	1	1	0	0
0	1 _{x1}	1 _{x0}	1 _{x1}	0
0	0 _{x0}	1 _{x1}	1 _{x0}	1
0	0 _{x1}	1 _{x0}	1 _{x1}	0
0	1	1	0	0

Image

4	3	4
2	4	

Convolved
Feature

Convoluting a 5x5x1 image with a 3x3x1 kernel to get a 3x3x1 convolved feature

Image Dimensions = 5 (Height)
x 5 (Breadth) x 1 (Number of
channels, eg. RGB)

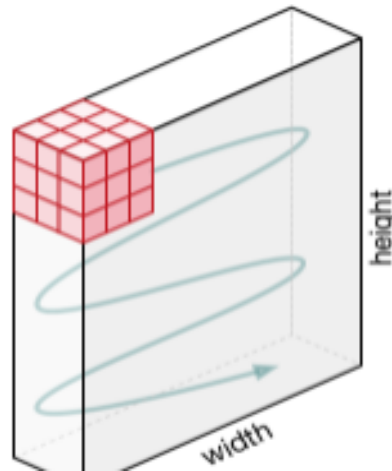
In the above demonstration, the
green section resembles our
5x5x1 input image, I. The
element involved in carrying
out the convolution operation in
the first part of a Convolutional
Layer is called the
Kernel/Filter, K, represented in
the color yellow. We have
selected **K as a 3x3x1 matrix**.

Filtre (kernel ou matrice de convolution)

```
Kernel/Filter, K =
```

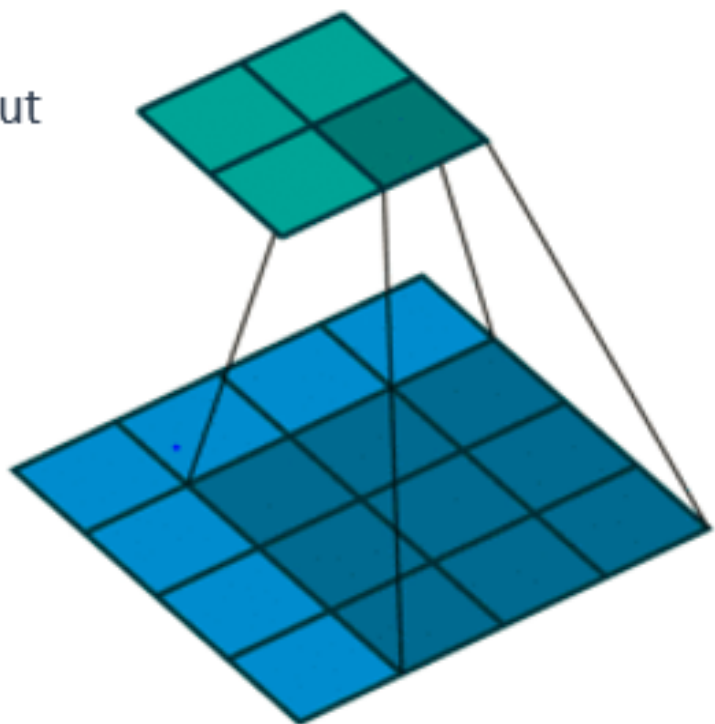
```
1  0  1  
0  1  0  
1  0  1
```

The Kernel shifts 9 times because of **Stride Length = 1 (Non-Strided)**, every time performing a **matrix multiplication operation between K and the portion P of the image** over which the kernel is hovering.



Visualization of convolution

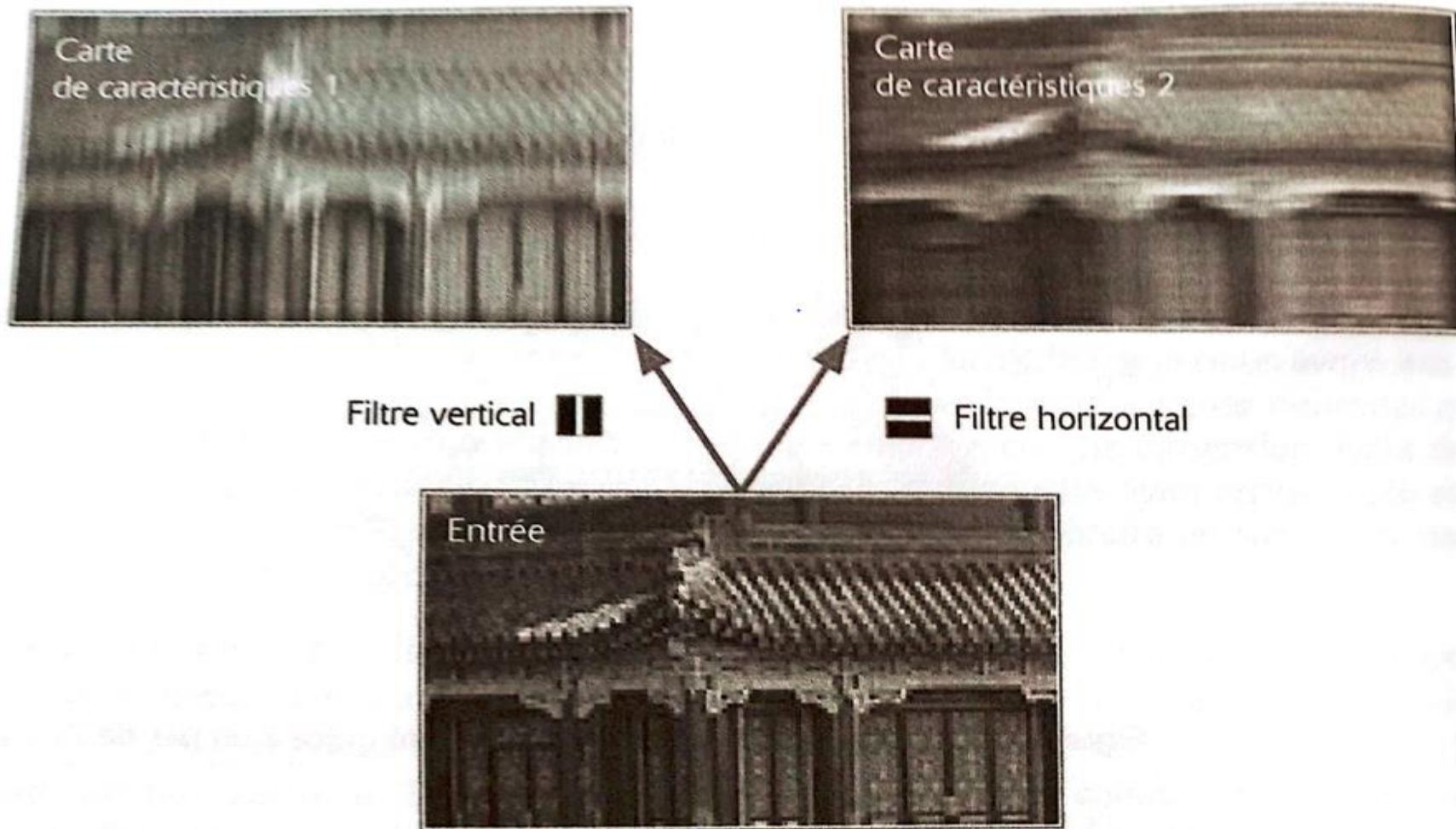
Having a 4x4 image and a 3x3 filter, hence we are getting output after convolution 2x2



If we have $N \times N$ image size and $F \times F$ filter size then after convolution result will be $(N-F+1) \times (N-F+1)$

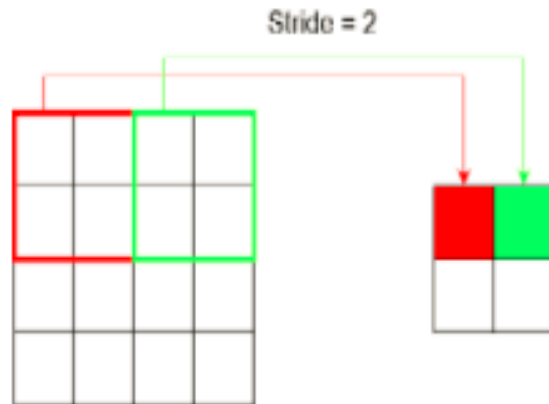
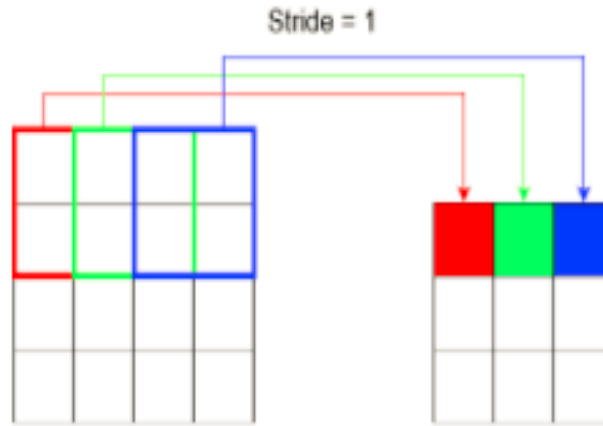
Exemple : deux filtres...

- Les neurones qui utilisent le premier filtre (carré noir avec ligne blanche verticale) ignorant tout ce qui trouve dans leur champ récepteur , à l'exception de la ligne verticale centrale
- Pour le second filtre, les neurones utilisant ses poids, ignoreront tout hormis la ligne horizontale.



Stride

Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on.



Padding

Without image padding, the result of convolution will be smaller than the original image size. Thus, the image shrinks every time a convolution operation is performed.

Also, pixels on the corner are much less used than those in the middle of the image.

To avoid these we add layers of zeros to our input image, we'll call this process padding.

0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

Types of padding :

- **Valid padding** : It implies no padding at all.
- **Same padding** : In this case, we add p layers such that the output image has the same dimensions as the input image.

Input Kernel Output

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

*

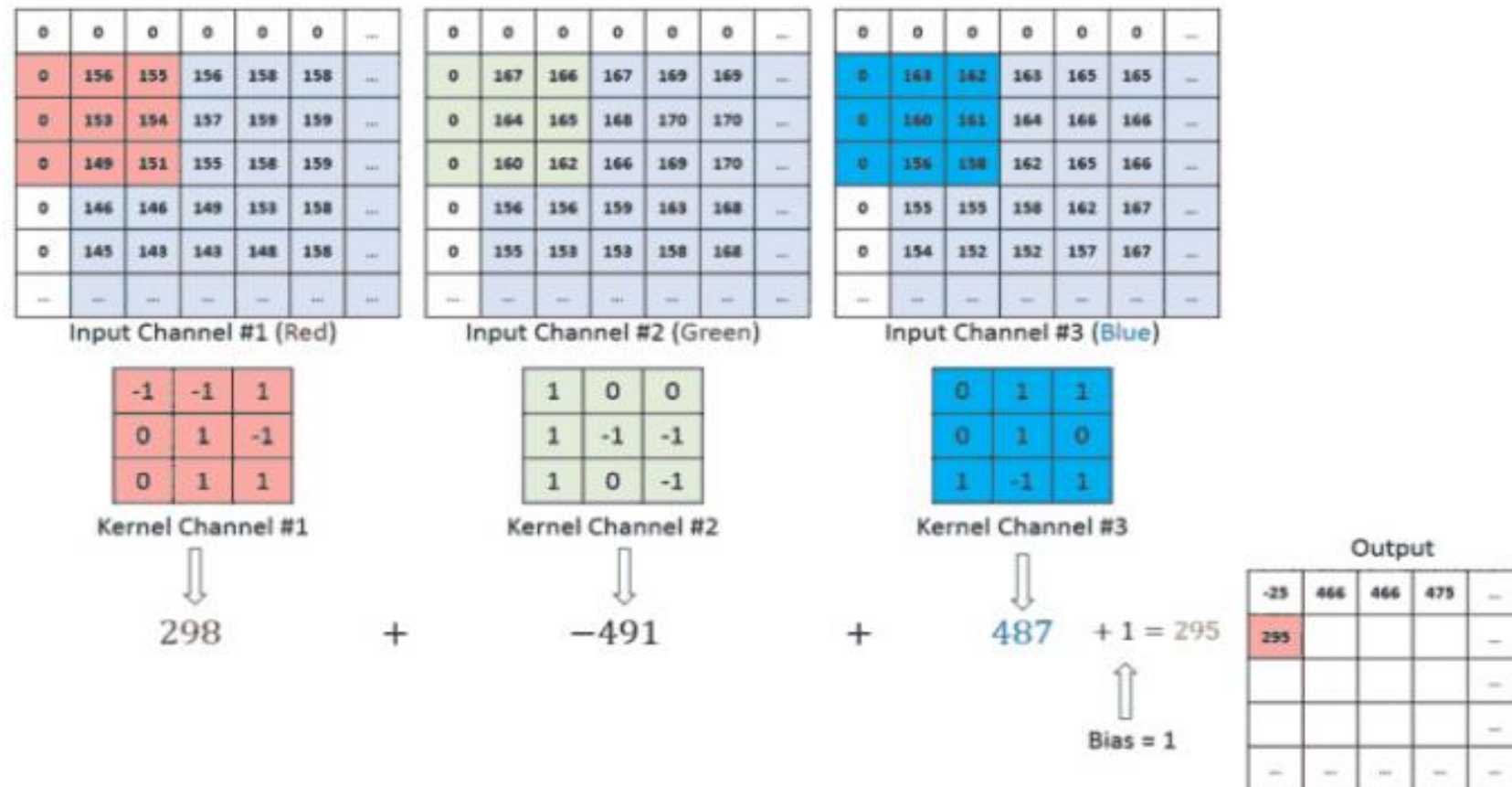
0	1
2	3

=

0	3	8	4
9	19	25	10
21	37	43	16
6	7	8	0

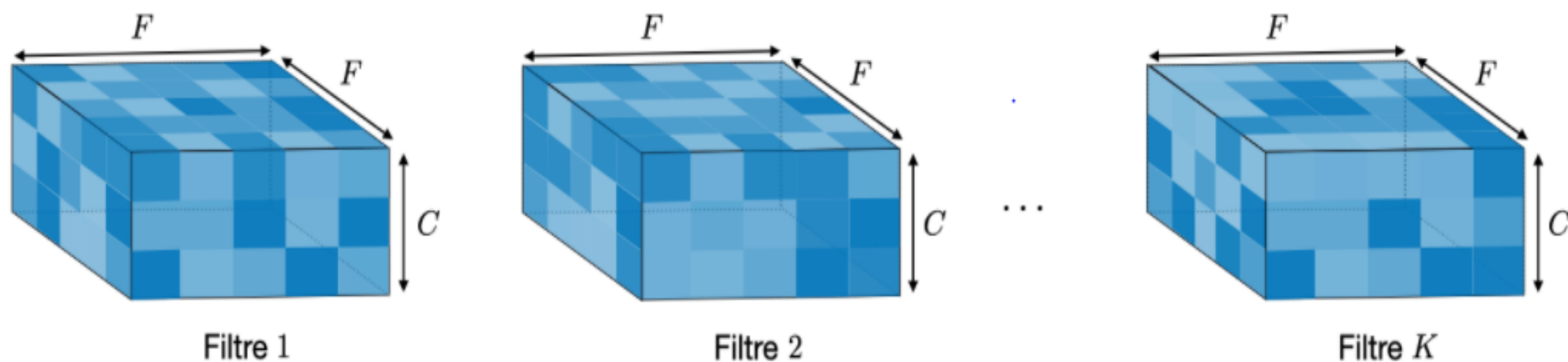
The result size of a convolution will be $((N - F + 2P) / S) + 1$

In the case of images with multiple channels (e.g. RGB), the Kernel has the same depth as that of the input image. Matrix Multiplication is performed between K_n and I_n stack ($[K1, I1]; [K2, I2]; [K3, I3]$) and all the results are summed with the bias to give us a squashed one-depth channel Convolved Feature Output.



Convolution operation on a $M \times N \times 3$ image matrix with a $3 \times 3 \times 3$ Kernel

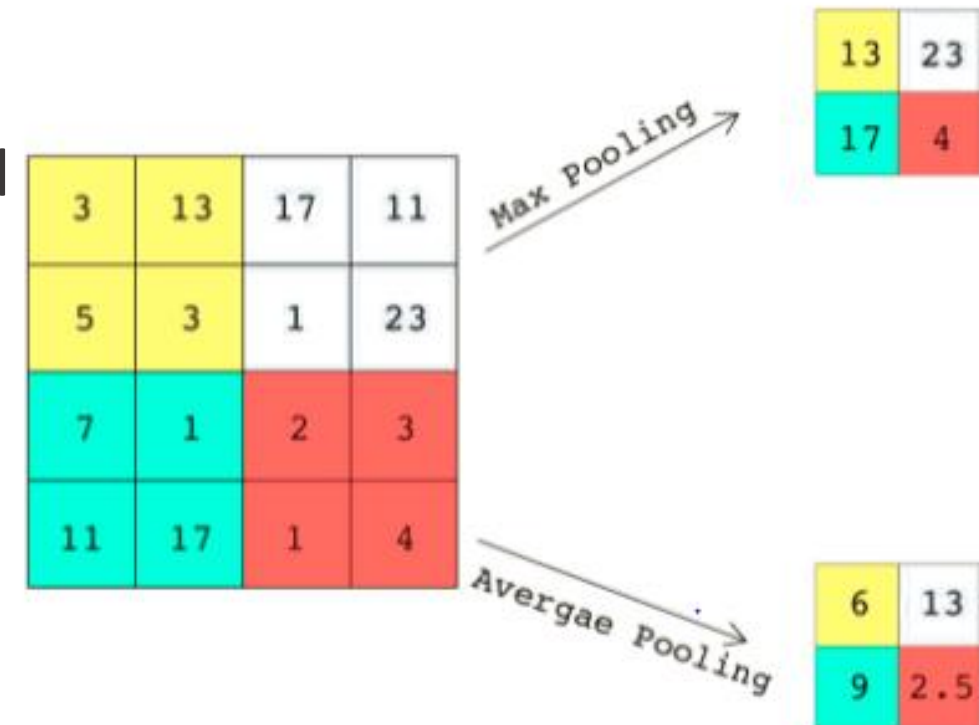
□ **Dimensions d'un filtre** — Un filtre de taille $F \times F$ appliqué à une entrée contenant C canaux est un volume de taille $F \times F \times C$ qui effectue des convolutions sur une entrée de taille $I \times I \times C$ et qui produit un *feature map* de sortie (aussi appelé *activation map*) de taille $O \times O \times 1$.



Remarque : appliquer K filtres de taille $F \times F$ engendre un feature map de sortie de taille $O \times O \times K$.

Pooling

- Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.
- In the example below we will reduce each spatial dimension in half :

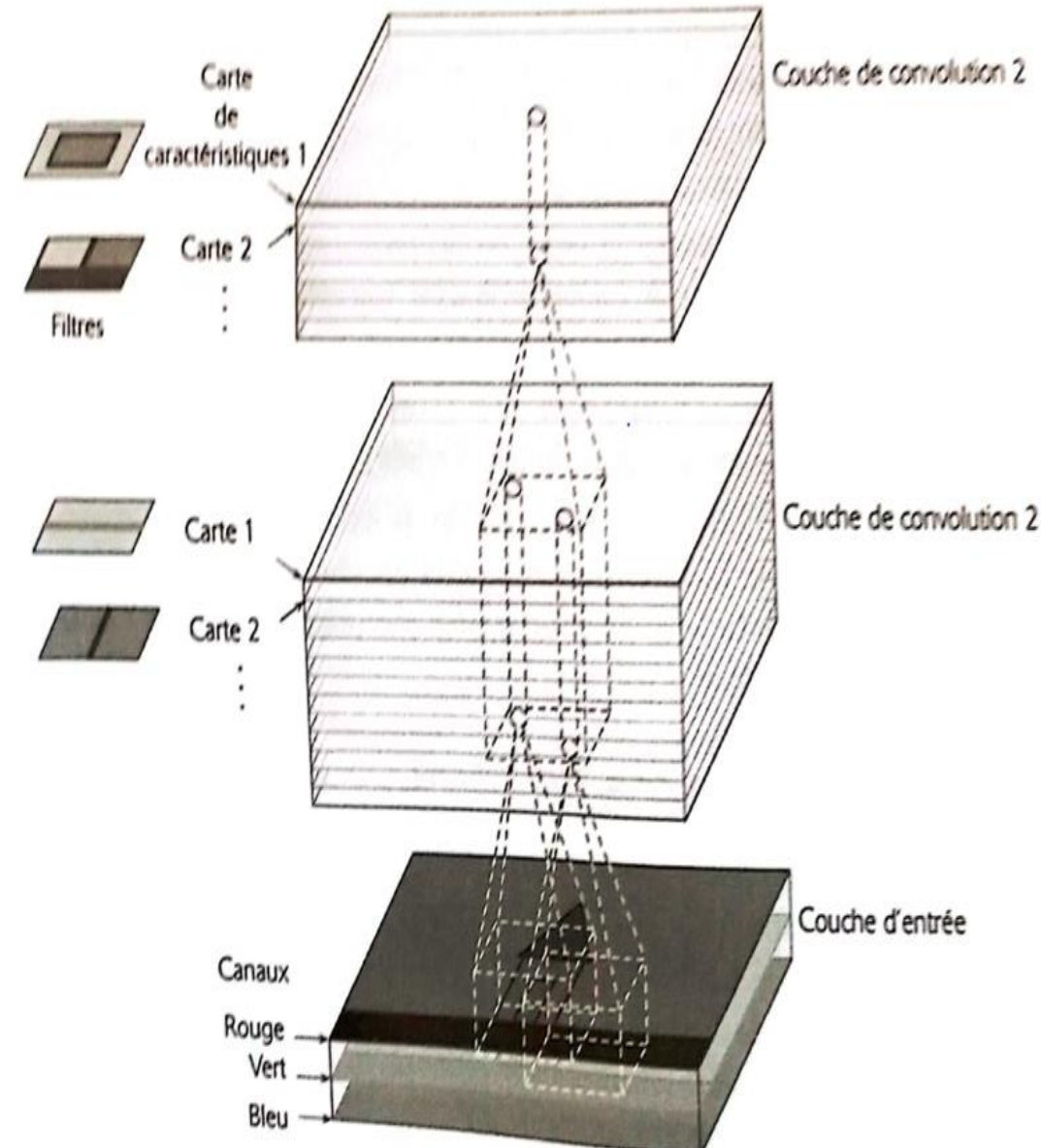


There are two types of Pooling: Max Pooling and Average Pooling

- . **Max Pooling** returns the **maximum value** from the portion of the image covered by the Kernel.
- On the other hand, **Average Pooling** returns the **average of all the values** from the portion of the image covered by the Kernel.
- Max Pooling also performs as a **Noise Suppressant**. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism.
- Hence, we can say that **Max Pooling performs a lot better than Average Pooling**.

Empiler plusieurs cartes de caractéristiques (Feature map)

- Jusqu'à présent, nous avons représenté chaque couche de convolution comme une fine couche à deux dimensions, mais en réalité elle est constituée de plusieurs cartes de caractéristiques de taille égale.
- Il est donc plus correct de la représenter en 3 dimensions.
- A l'intérieur d'une carte de caractéristiques, tous les neurones partagent les mêmes paramètres (poids et terme constant).
- Les différentes cartes de caractéristiques peuvent avoir des paramètres distincts.



- En résumé, une couche de convolution applique simultanément plusieurs filtres à ses entrées, ce qui lui permet de détecter plusieurs caractéristiques n'importe où dans ses entrées.

Remarque :

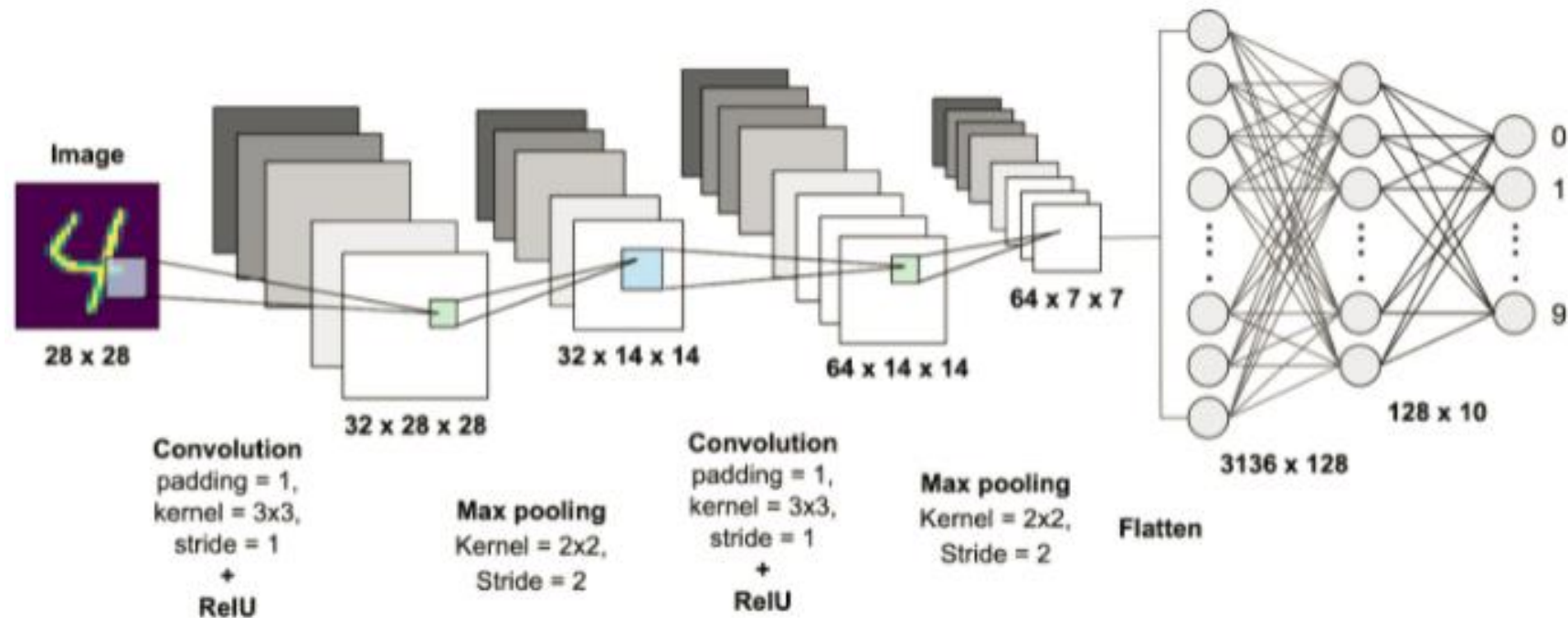
- Puisque tous les neurones d'une carte de caractéristiques partagent les mêmes paramètres (i.e. le même filtre), le nombre de paramètre du modèle s'en trouve considérablement réduit.
- Plus, important encore, cela signifie également que dès que le CNN a appris un motif en un endroit, il peut le reconnaître partout ailleurs
- A l'opposé, lorsqu'un réseau classique a appris à reconnaître un motif en un endroit, il ne peut le reconnaître qu'en cet endroit précis.

Classification-Fully Connected Layer (FC Layer)

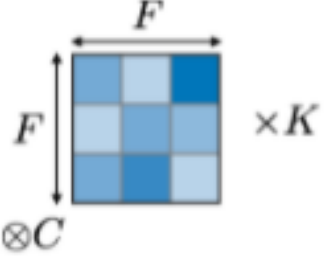
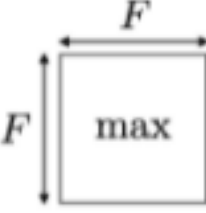
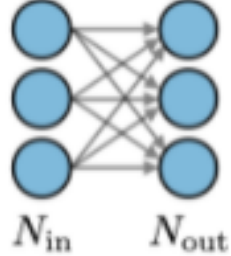
So far we've learned features through convolution, introduced non-linearity through activation functions and reduced dimensionality with pooling but we need to go further in order to use these features for classifying.

Convolution and Pool layers output high-level features of input that fully connected layers uses for classification ; the second part of the Neural Net will have the same architecture as Deep Feed Forward Neural Net.

We finally obtain a probability of image belonging to a particular class.



I : taille de l'entrée ; **F** = taille du filtre ; **C** = nombre de canaux ;
K : nombre de filtres ; **S** : stride

	CONV	POOL	FC
Illustration			
Taille d'entrée	$I \times I \times C$	$I \times I \times C$	N_{in}
Taille de sortie	$O \times O \times K$	$O \times O \times C$	N_{out}
Nombre de paramètres	$(F \times F \times C + 1) \cdot K$	0	$(N_{in} + 1) \times N_{out}$
Remarques	<ul style="list-style-type: none"> • Un paramètre de biais par filtre • Dans la plupart des cas, $S < F$ • $2C$ est un choix commun pour K 	<ul style="list-style-type: none"> • L'opération de pooling est effectuée pour chaque canal • Dans la plupart des cas, $S = F$ 	<ul style="list-style-type: none"> • L'entrée est aplatie • Un paramètre de biais par neurone • Le choix du nombre de neurones de FC est libre

Liens

- <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- <https://kharshit.github.io/blog/2018/12/14/filters-in-convolutional-neural-networks>
- <https://stanford.edu/~shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels>
- <https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/>

Youtube :

Thibault Neveu, Utiliser et sauvegarder un modèle - Se former à Tensorflow 2.0 #10

- https://www.youtube.com/watch?v=8l1LdhH2J9c&ab_channel=ThibaultNeveu

Convolutional Neural Networks (CNNs) explained

- https://www.youtube.com/watch?v=YRhxdVk_sIs&ab_channel=deeplizard

Les réseaux de convolution (CNN) | Intelligence artificielle 47

- https://www.youtube.com/watch?v=zG_5OtgfxAg&t=6s&ab_channel=Science4All