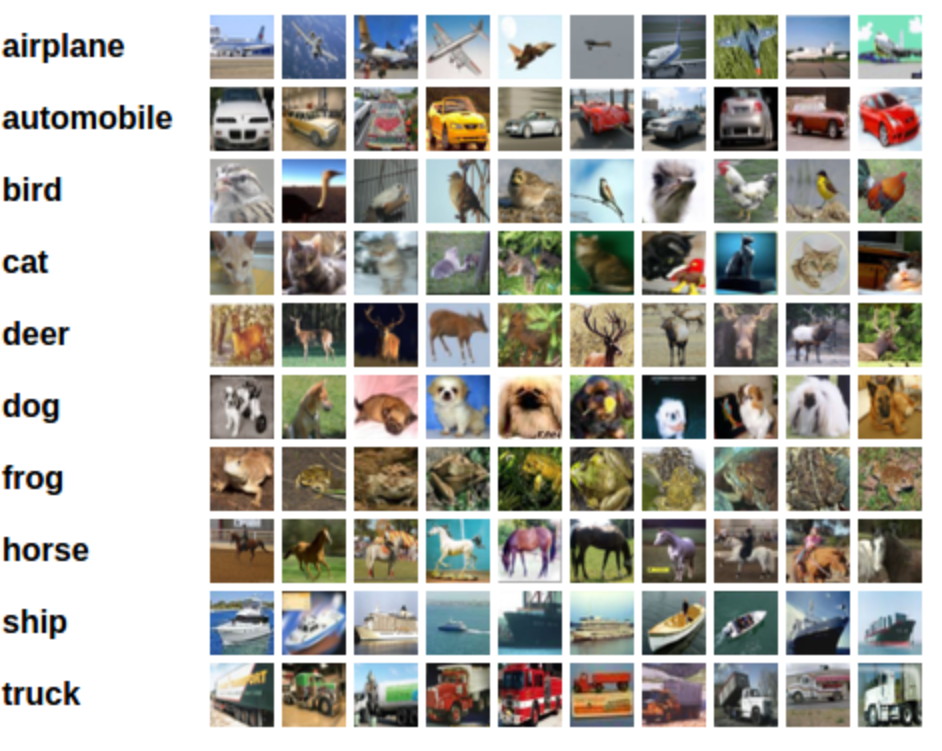
## Intelligence Artificielle et Analyse de données

TP2 : Classification d’images (CIFAR10)

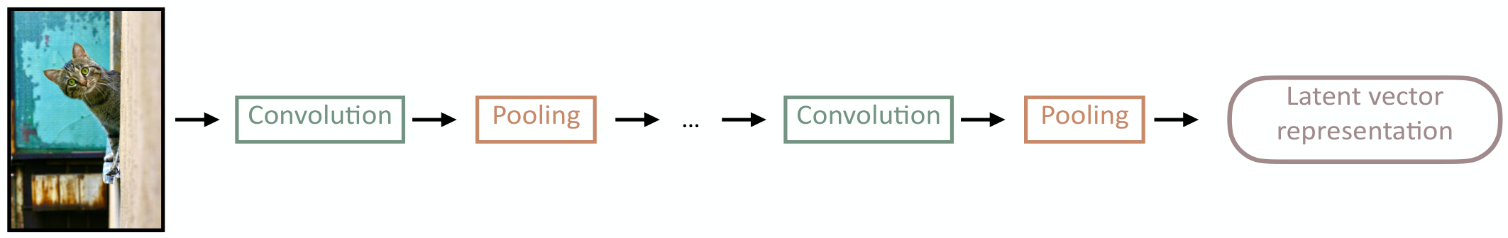
**Données : CIFAR10**

On s’intéresse au jeu de données CIFAR10, illustré ci-dessous. A chaque image (donnée d’entrée) correspond une étiquette (*label*) parmi 10 possibles (donnée de sortie, i.e. à prédire).



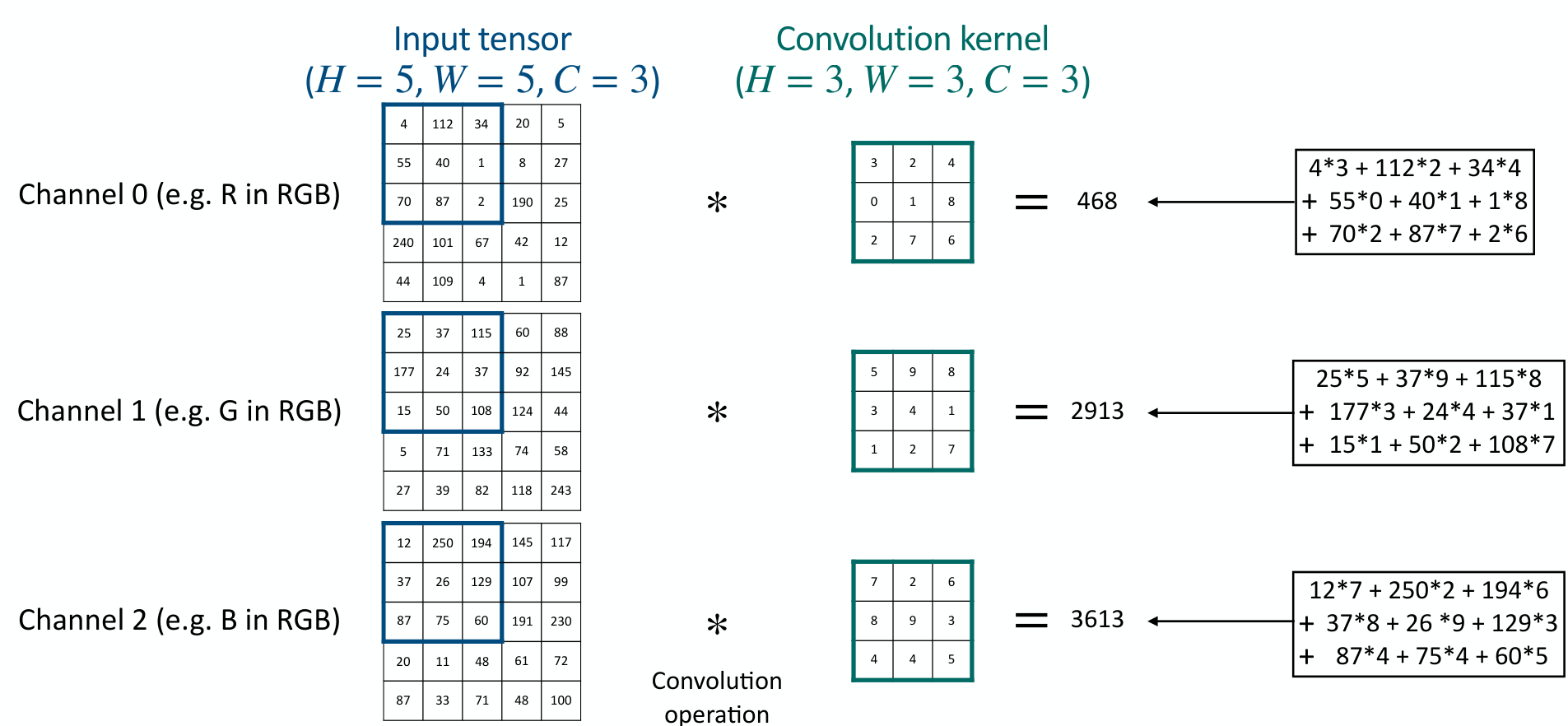
**Réseaux convolutionels (CNN) :**

Les réseaux convolutionels (*CNN*) exploitent 2 propriétés généralement partagées par les images :

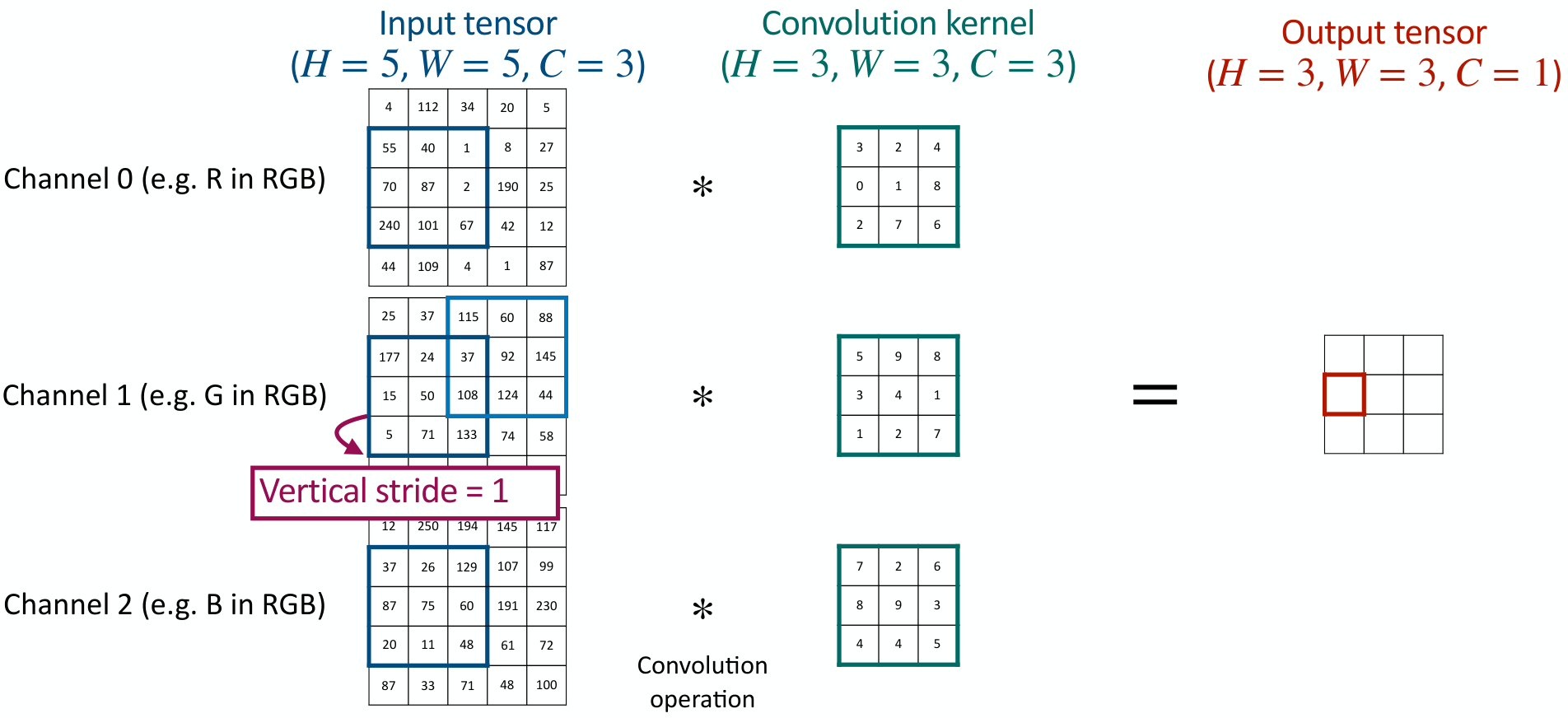
* Localité : les pixels proches partagent des informations ;
* Invariance (partielle) par rapport aux translations.

Opération de convolution :

Applique une série de filtres (*kernels*) à une partie de l’image. Illustration pour un *kernel* de taille 3x3 sur une image RGB :



Le filtre est décalé progressivement (*stride*) pour être appliqué à l’ensemble de l’image :

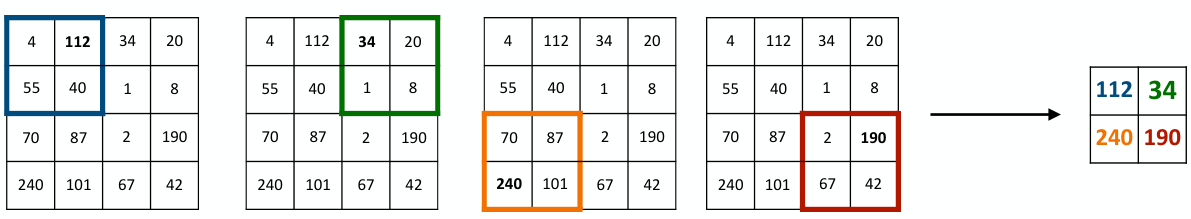


Remarque : Les valeurs (poids) des *filtres* de convolution sont apprises durant l’entraînement.

Opérations de *pooling* :

Appliquent une opération simple à une partie de l’image, par exemple prendre le maximum (*max pooling*), le minimum ou la moyenne.

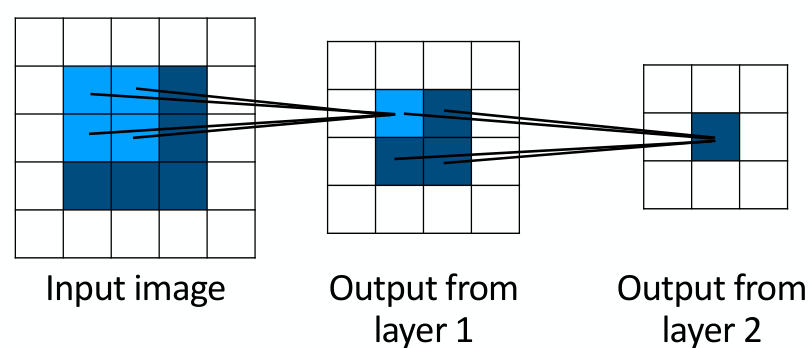
Exemple de *max pooling* avec un *kernel* 2x2 et *stride* 2 :



Le *pooling* permet de :

* Rendre le réseau moins sensible aux petites variations de l'entrée en écartant certains détails sans importance. La mise en commun fournit une forme d'invariance locale ;
* Réduire la résolution => les calculs de convolution sont plus rapides ;
* Augmente la taille du champ réceptif de chaque neurone.

Remarque : Ne contient pas de poids à apprendre.



1. Comprendre les données :

Compléter la cellule *comprendre les données* pour répondre aux questions suivantes :

* Quelle est la taille de chaque dataset (i.e. combien d’images) ?
* Visualiser les 5 premières images (un exemple pour visualiser une image est donné) avec le nom de leur classe ('plane', 'car', etc…).

1. Comprendre le modèle CNN :

A partir de la documentation et en complétant la cellule *modèle CNN*, répondre aux questions suivantes :

* Combien de couche (*layer*) sont implémentées ?
* Opération de convolution : quels sont les tailles des *kernel* et *stride* ?
* Opération de pooling : implémenter et afficher une opération de *max pooling* sur une image (en dehors de l’implémentation du modèle), et afficher le résultat.

1. Utilisation du GPU :

Dans le menu exécution (google colab) modifier le type d’exécution => GPU.

Lancer l’apprentissage avec une seule itération

* Comparer temps d’exécution d’une itération CPU/Cuda.

1. Lancer l’apprentissage :

Se limiter en nombre d’itération (20 ou 40 max) car temps d’apprentissage important.

Tracer / comparer les courbes d’erreur *train / valid*.

1. Exploiter le modèle CNN :

Adapter la fonction *get\_score*, implémentée dans l’exemple de classification *Titanic* => estimer en % le taux de réussite du modèle CNN sur chaque dataset.