Réseaux de neurone pour la vision par ordinateur

Deep learning lecture

Kevin Helvig, ONERA/DTIS / Aurélien Plyer, ONERA/DTIS
kevin.helvig@onera.fr
aurelien.plyer@onera.fr

ONERA
THE FRENCH AEROSPACE LAB

retour sur innovation

I – Retour sur le cours précédent : MLP, Perceptron

Qu'est ce qui vous revient en tête ?

I – Retour sur le cours précédent : MLP, Perceptron

Loss/Perte : fonction qu'on va chercher à optimiser via notre réseau de neurones

- Typiquement : **une distance qu'on va chercher à minimiser**

- **Est-ce que vous en avez en tête ?**

Retour sur le cours précédent : MLP, Perceptron

Loss/Perte : fonction qu'on va chercher à optimiser via notre réseau de neurones

- Typiquement : **une distance qu'on va chercher à minimiser**

- **Est-ce que vous en avez en tête ?**

Cross-entropie

MSE/moindres carrés

L1/L2 (distances euclidiennes et variants)

Retour sur le cours précédent : MLP, Perceptron

Notion générale d'entraînement de modèle, backpropagation

Retour sur le cours précédent : MLP, Perceptron

Loss de classification : cross-entropie

Equivaut à chercher **une frontière dont la distance entre tous les points est optimale**

Retour sur le cours précédent : MLP, Perceptron

Loss de régression : norme L1

Equivaut à chercher la droite qui minimise l'écart entre tous les points

I – Les limites du MLP en vision

Première idée pour traiter des images : **empiler les pixels en vecteurs 1D**

I – Les limites du MLP en vision

Première idée pour traiter des images : **empiler les pixels en vecteurs 1D**

Petites matrices tabulaires, imagettes : ok

Images full HD = c'est non

MNIST

*Image 32*32 px*

Vecteur linéaire

$V \in \mathbb{R}^{784}$

Image full-HD

*Image 1280*720 px*

Vecteur linéaire

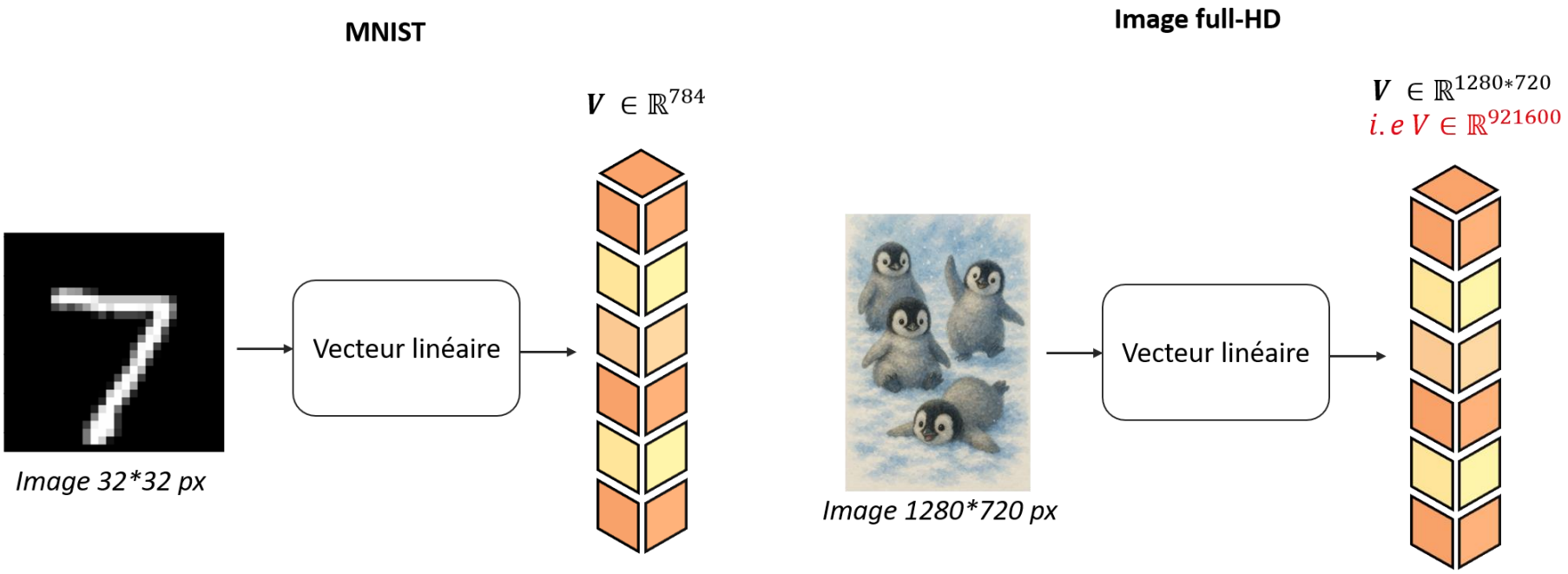
$V \in \mathbb{R}^{1280*720}$

$i.e\ V \in \mathbb{R}^{921600}$

I – Les limites du MLP en vision

Petites matrices tabulaires, imagettes : ok

Images full HD = c'est non



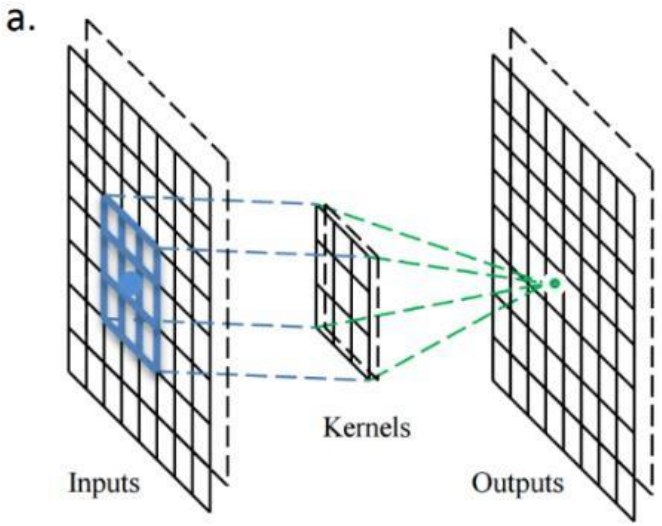
Autre problème : perte de **cohérence spatiale de l'information**

II– La convolution à la rescousse

Comment capturer mieux l'information spatiale ? (données « 2-3D »)

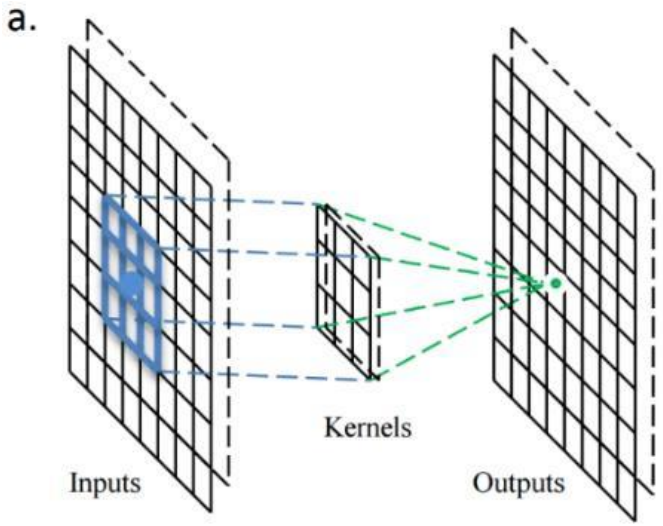
II– La convolution à la rescousse

La solution : un neurone qui capture **un masque de l'image**
La convolution



II– La convolution à la rescousse

La solution : un neurone qui capture **un masque de l'image**
La convolution



*Convolution M*N (1 canal)*

$$y_{i,j} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{i+m,j+n} \cdot w_{m,n}$$

*Convolution M*N (3 canal)*

$$y_{k,i,j} = \sum_{c=1}^{C} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{c,i+m,j+n} \cdot w_{k,c,m,n}$$

K noyaux appris = k neurones en //

II– La convolution à la rescousse

L’invariance par translation des neurones de convolution : au tableau

II– La convolution à la rescousse

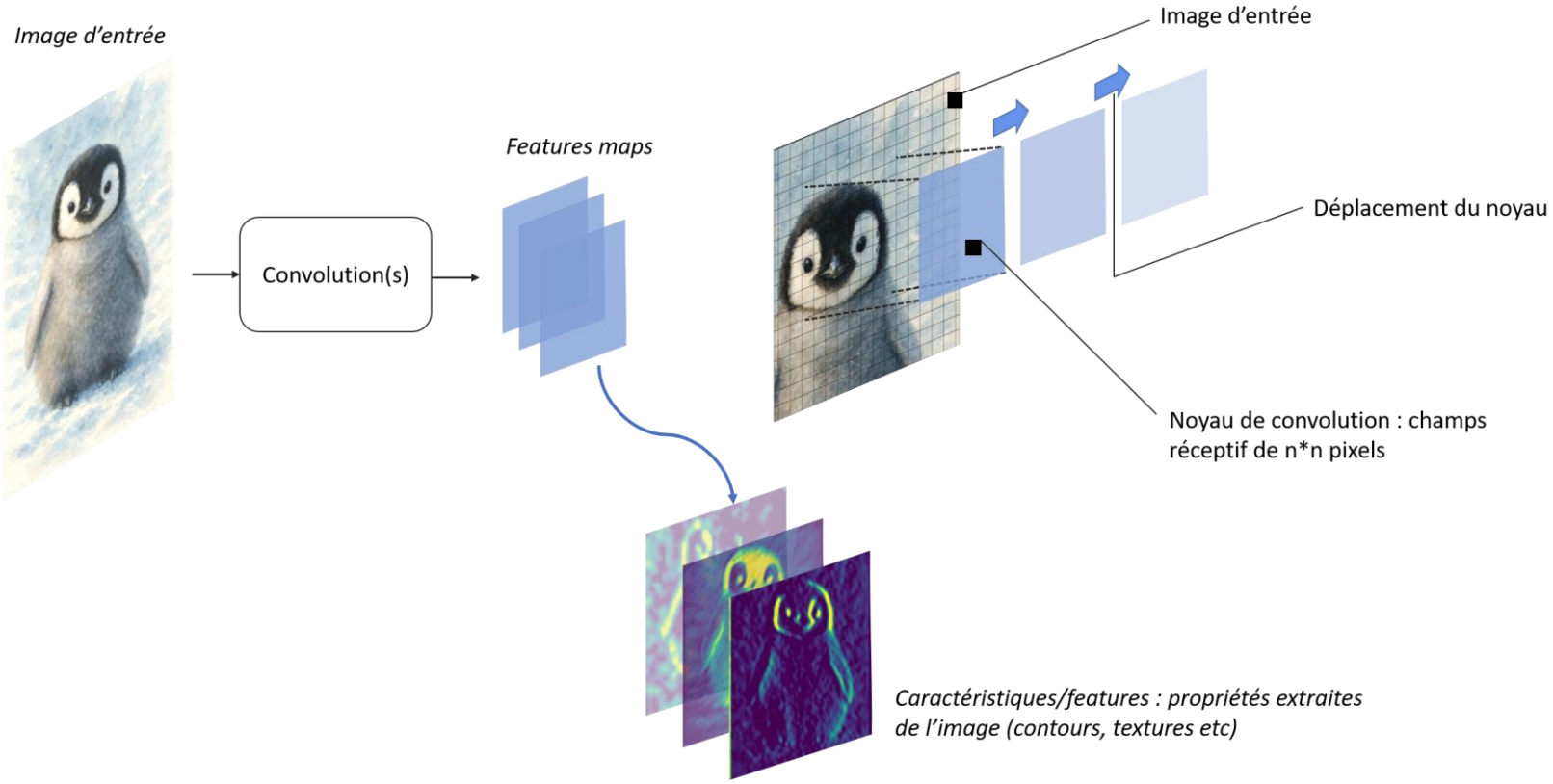
Synthèse visuelle sur la convolution

II– La convolution à la rescousse

Synthèse visuelle : intuition de la convolution comme « filtre appris »

II– La convolution à la rescousse

Récapitulatif visuelle : convolution



II– La convolution à la rescousse

Récapitulatif visuelle : convolution (formation de la sortie)

II– La convolution à la rescousse

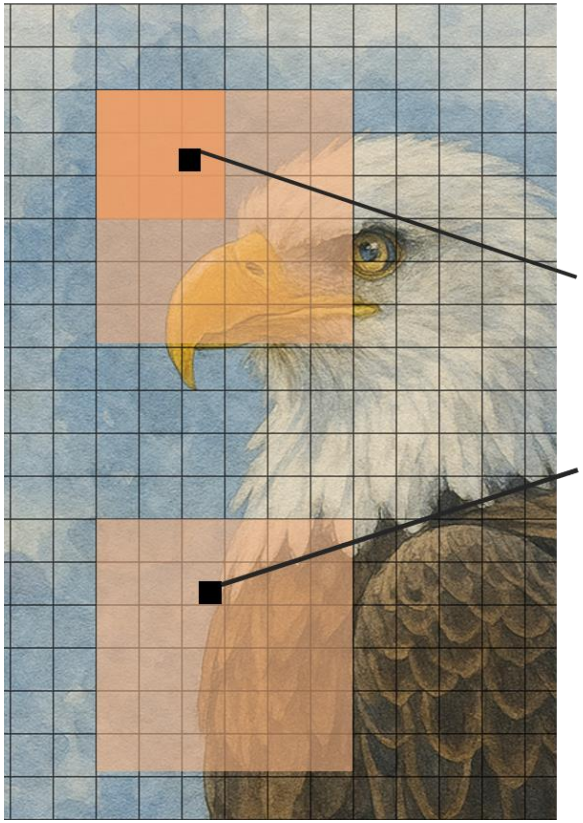
Les différents paramètres de la convolution neuronale : récap'

Compromis taille de sortie – champs observé par le masque

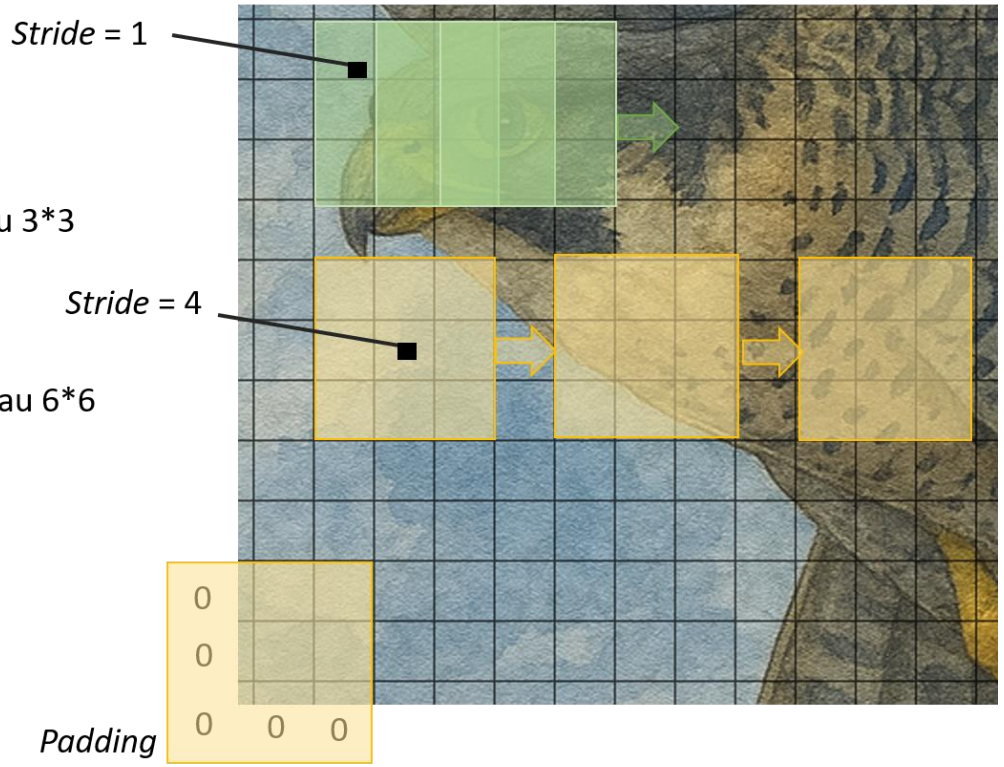
II– La convolution à la rescousse

Les différents paramètres de la convolution neuronale : récap'

Taille de noyau



Stride et Padding



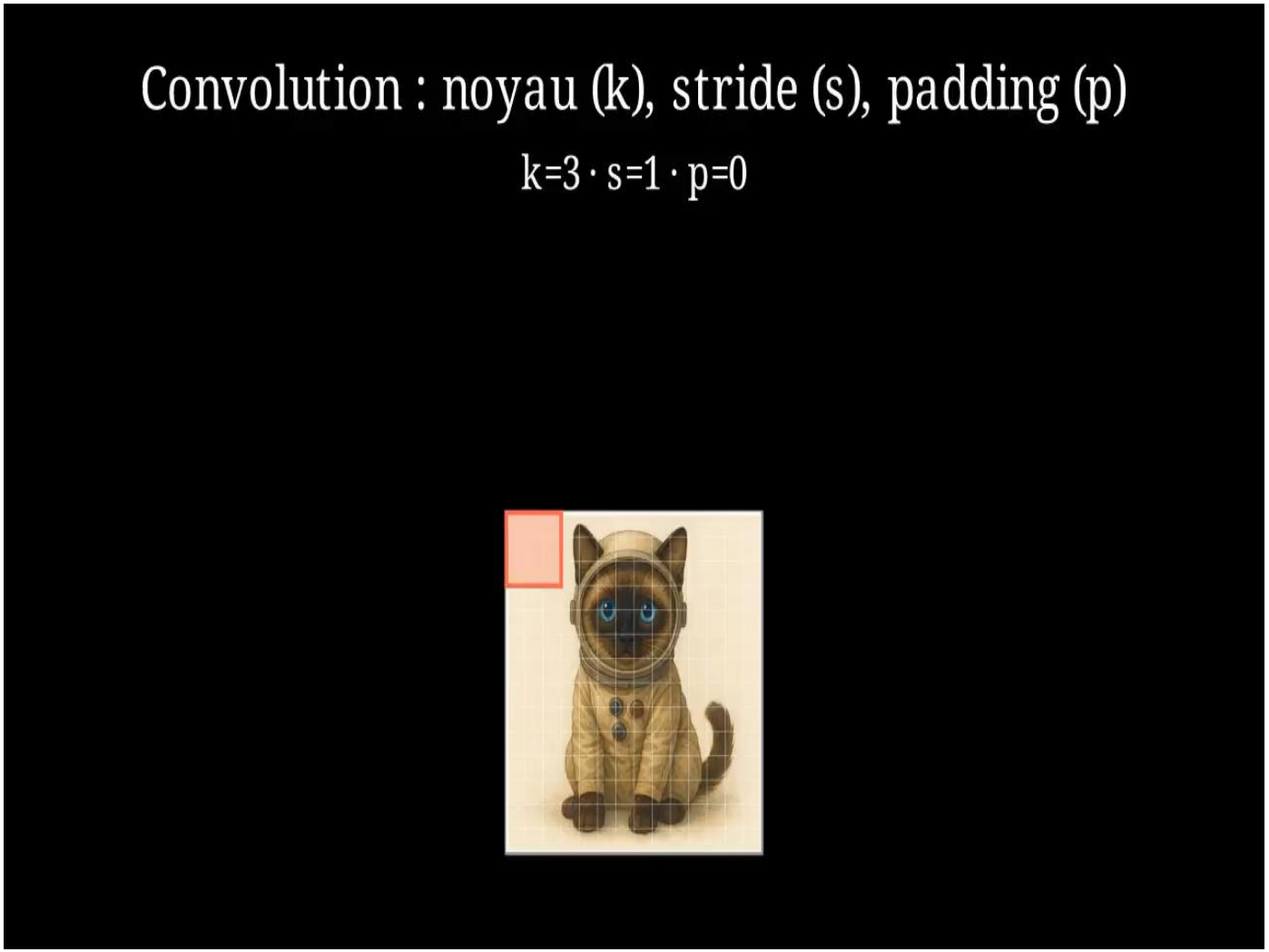
II– La convolution à la rescousse

La taille de sortie

$$\text{Output size} = \left\lfloor \frac{N + 2P - K}{S} \right\rfloor + 1$$

II– La convolution à la rescousse

Les différents paramètres de la convolution neuronale



II– La convolution à la rescousse

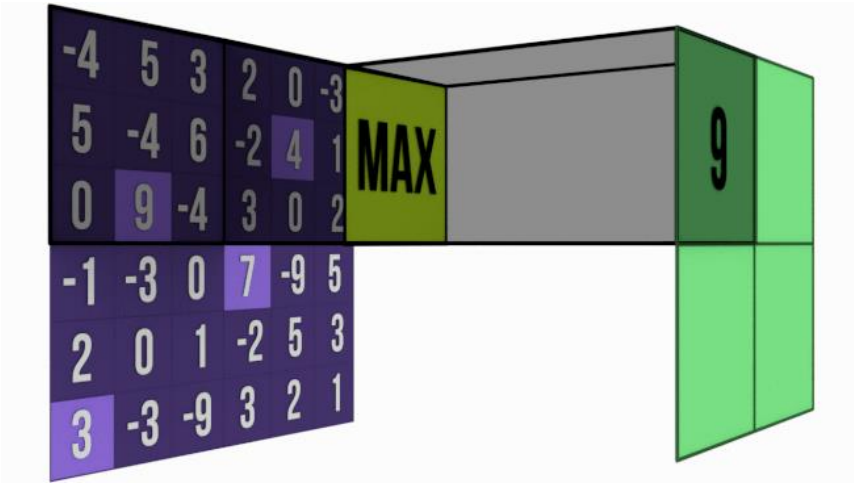
Ces paramètres : intérêt technique = réduire le nombre de paramètres, cas d'images très hautes définition, jouer sur la talle de la sortie

- Maxpooling/average pooling : kesako ?

II– La convolution à la rescousse

Ces paramètres : intérêt technique = réduire le nombre de paramètres, cas d'images très hautes définition, jouer sur la taille de la sortie

- Maxpooling/average pooling :
  - **Maximiser/normaliser la réponse d'une convolution** suivant le besoin (notion de filtrage de l'input)



II– La convolution à la rescousse

Ces paramètres : intérêt technique = réduire le nombre de paramètres, cas d'images très hautes définition

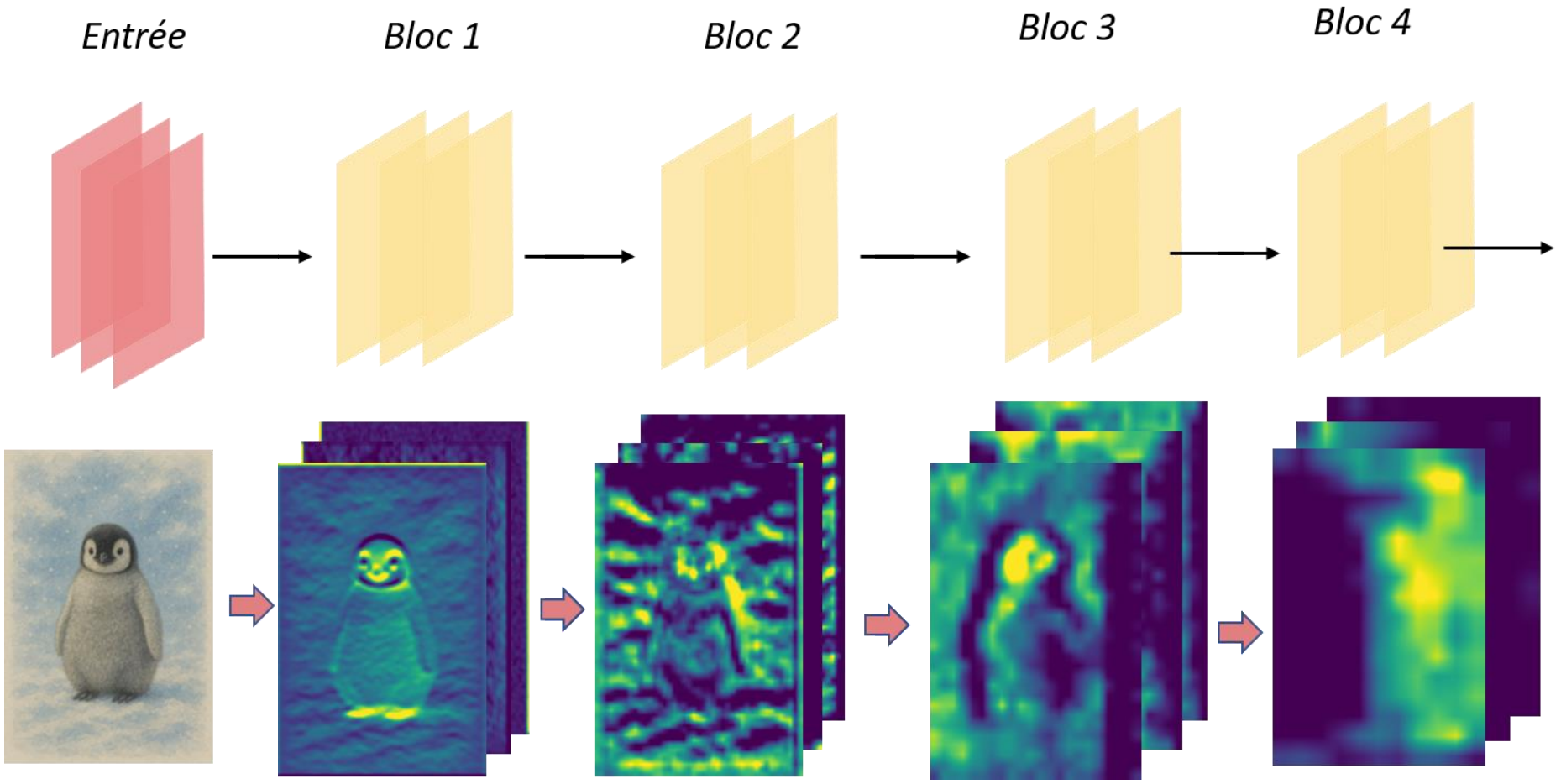
- Batch-normalization : moyenner la réponse à l'échelle du minibatch

III– Les convnets

De la même manière qu'on forme le MLP en empilant des perceptrons…

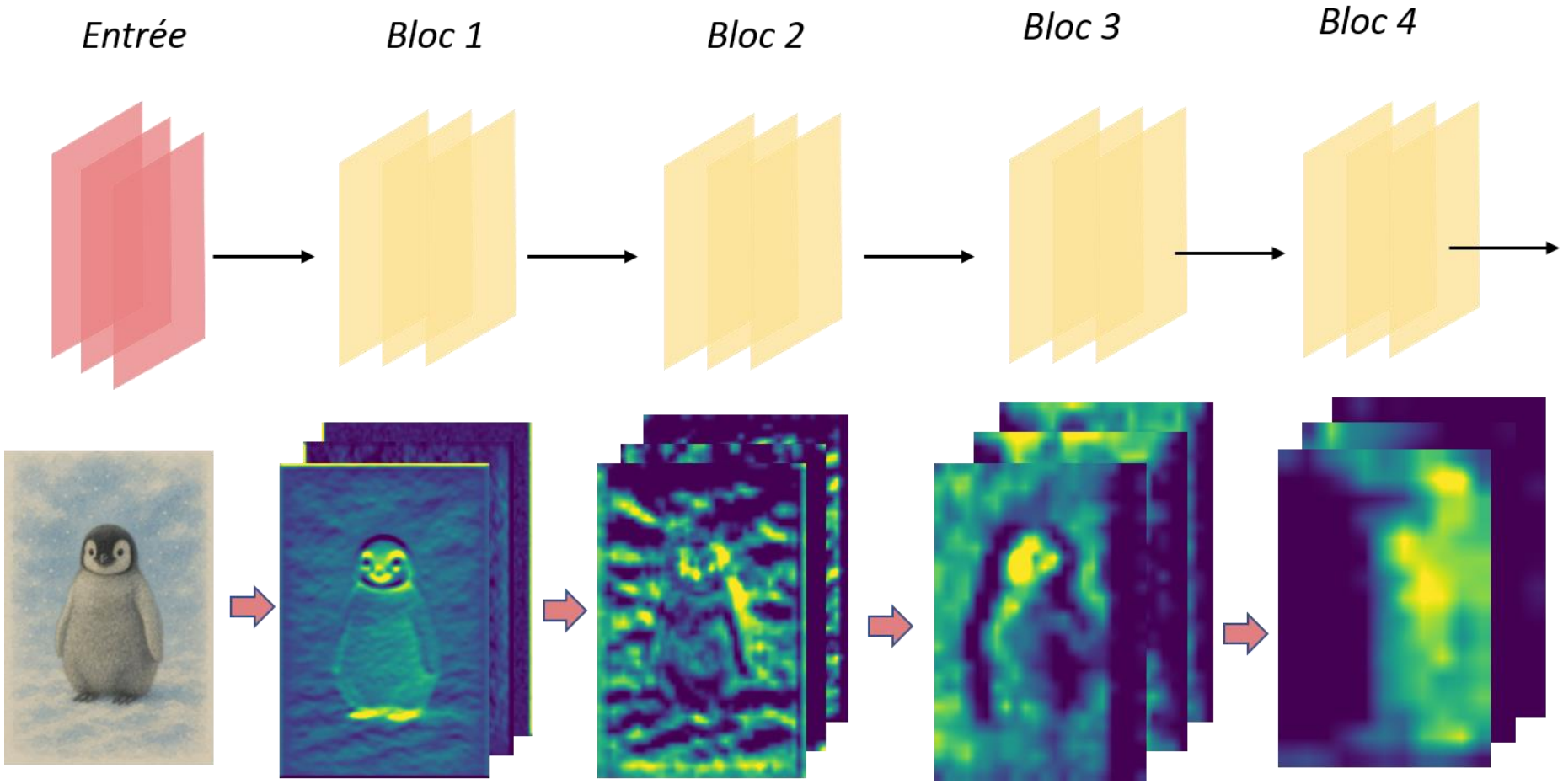
III– Les convnets

Empiler des couches de convolution neuronale



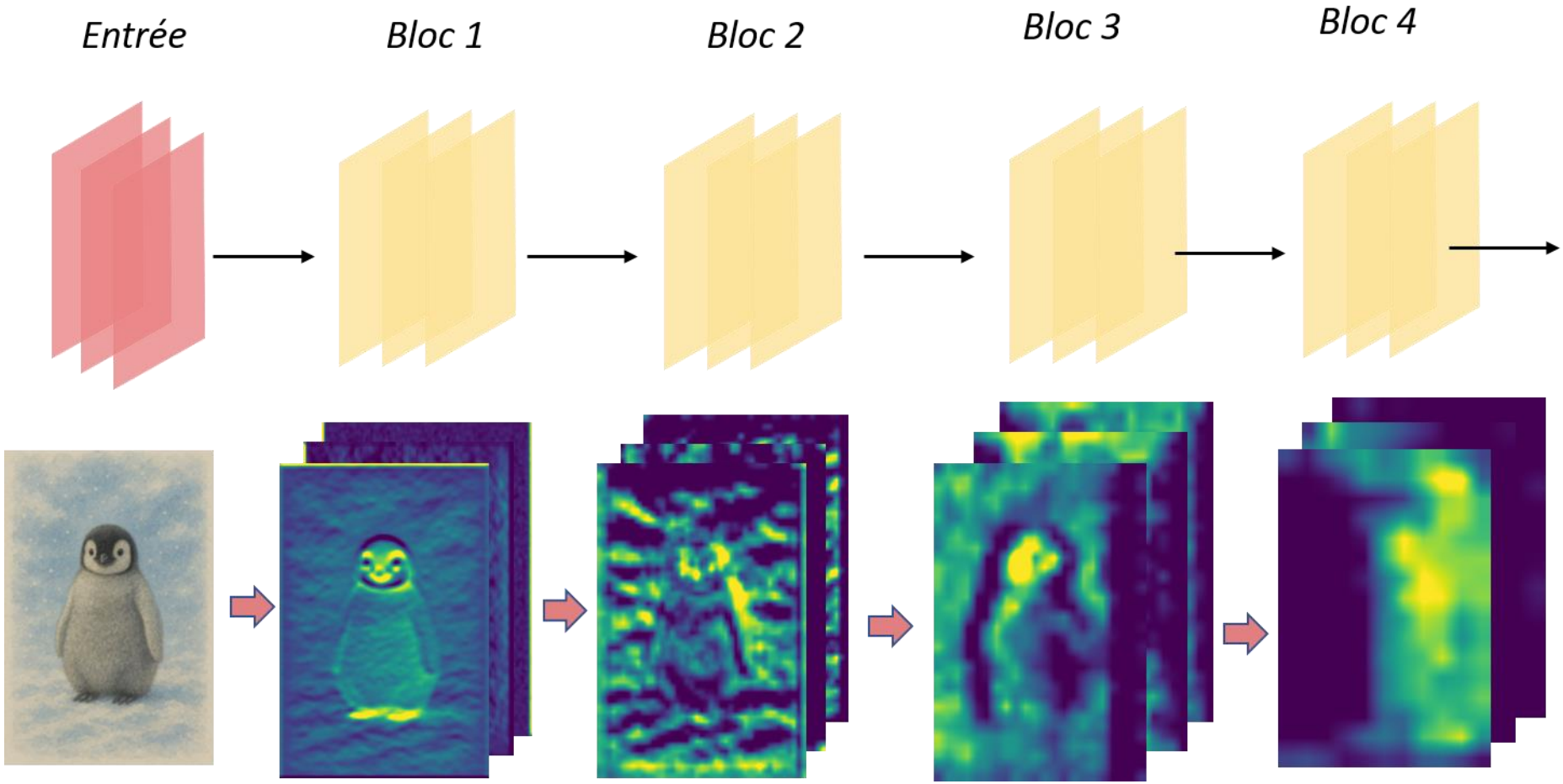
III– Les convnets

Extraire des propriétés de plus en plus fines de l'image



III– Les convnets

Extraire des propriétés de plus en plus fines (et abstraites) de l'image



III– Les convnets

Structure de base d'architecture de vision : le convnet

- Une partie convolutionnelle : extrait des caractéristiques de l'image

III– Les convnets

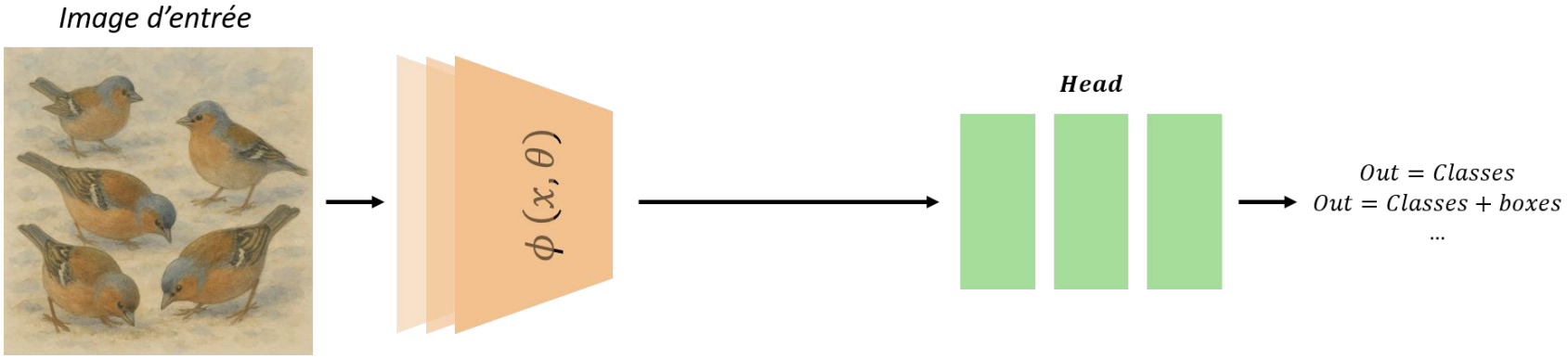
Structure de base d'architecture de vision : le convnet

- **Une partie convolutionnelle : extrait des caractéristiques de l'image**

- **Une tête (souvent : MLP)** : sépare **l'espace statistique** des entrées à partir de ces propriétés (la frontière en classification)

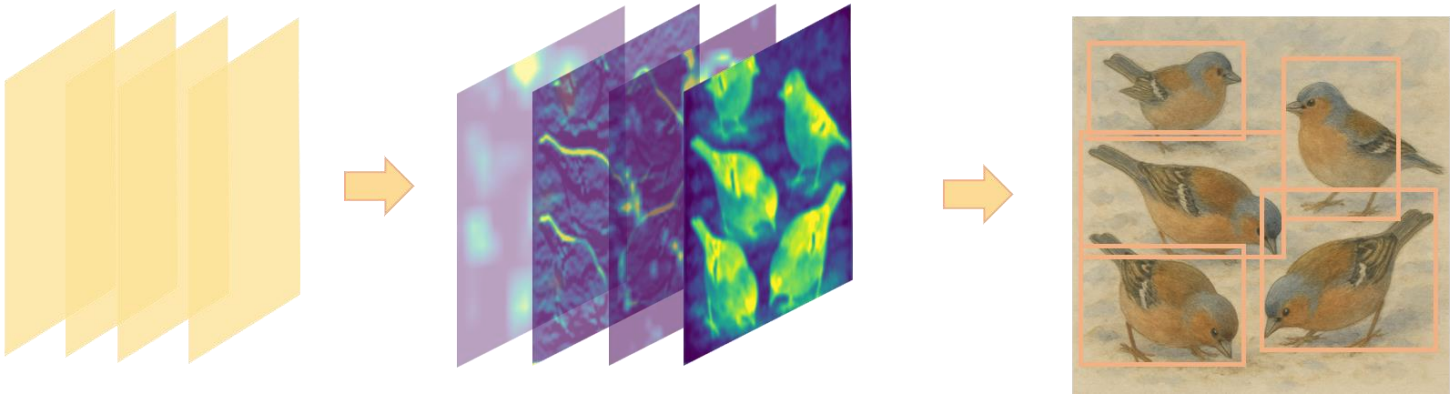
III– Les convnets

Structure de base d'architecture de vision : le convnet



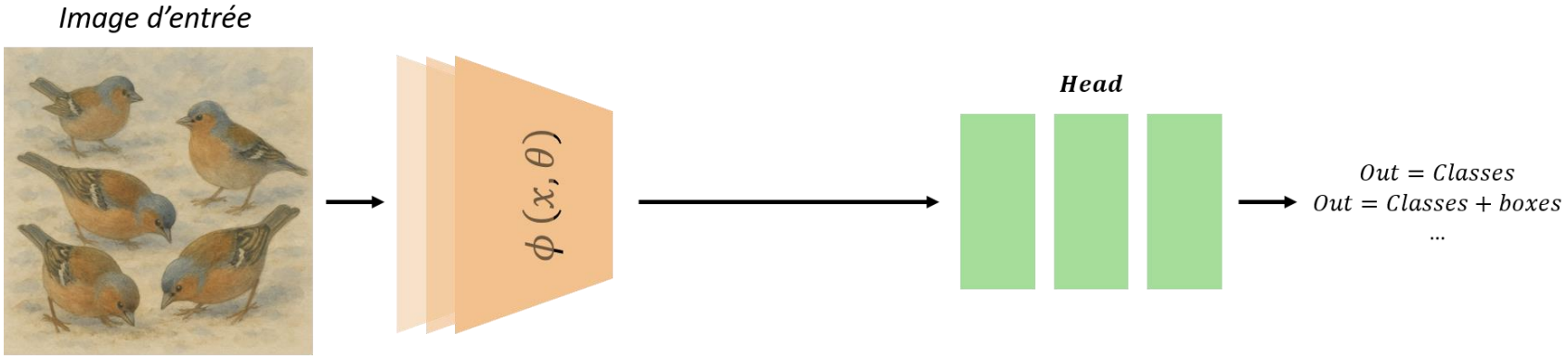
Backbone ϕ : extrait des features maps (CNN)

Head : réalisation de la tâche (fully-connected)



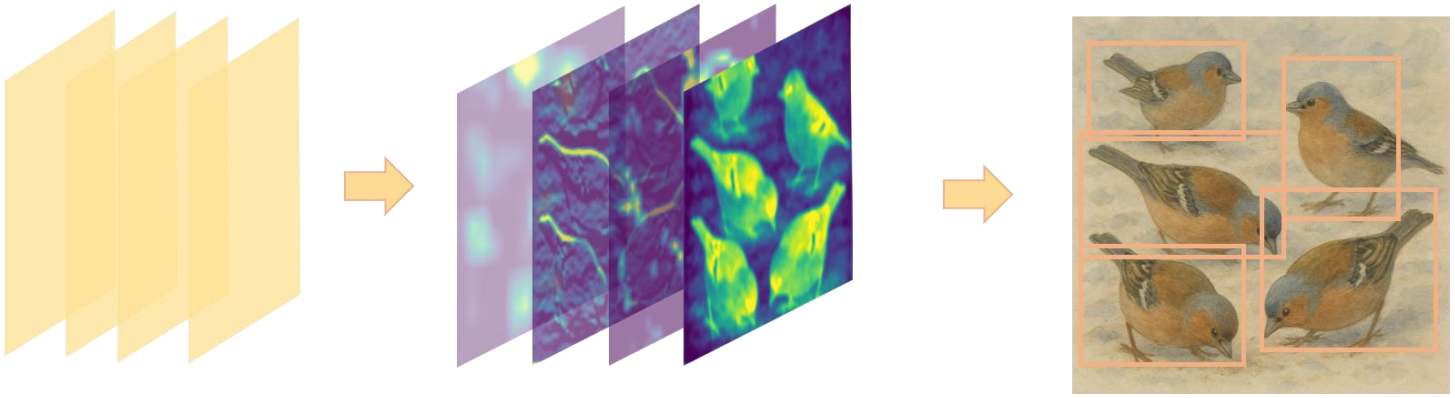
III– Les convnets

Structure de base d'architecture de vision : le convnet



Backbone ϕ : extrait des features maps (CNN)

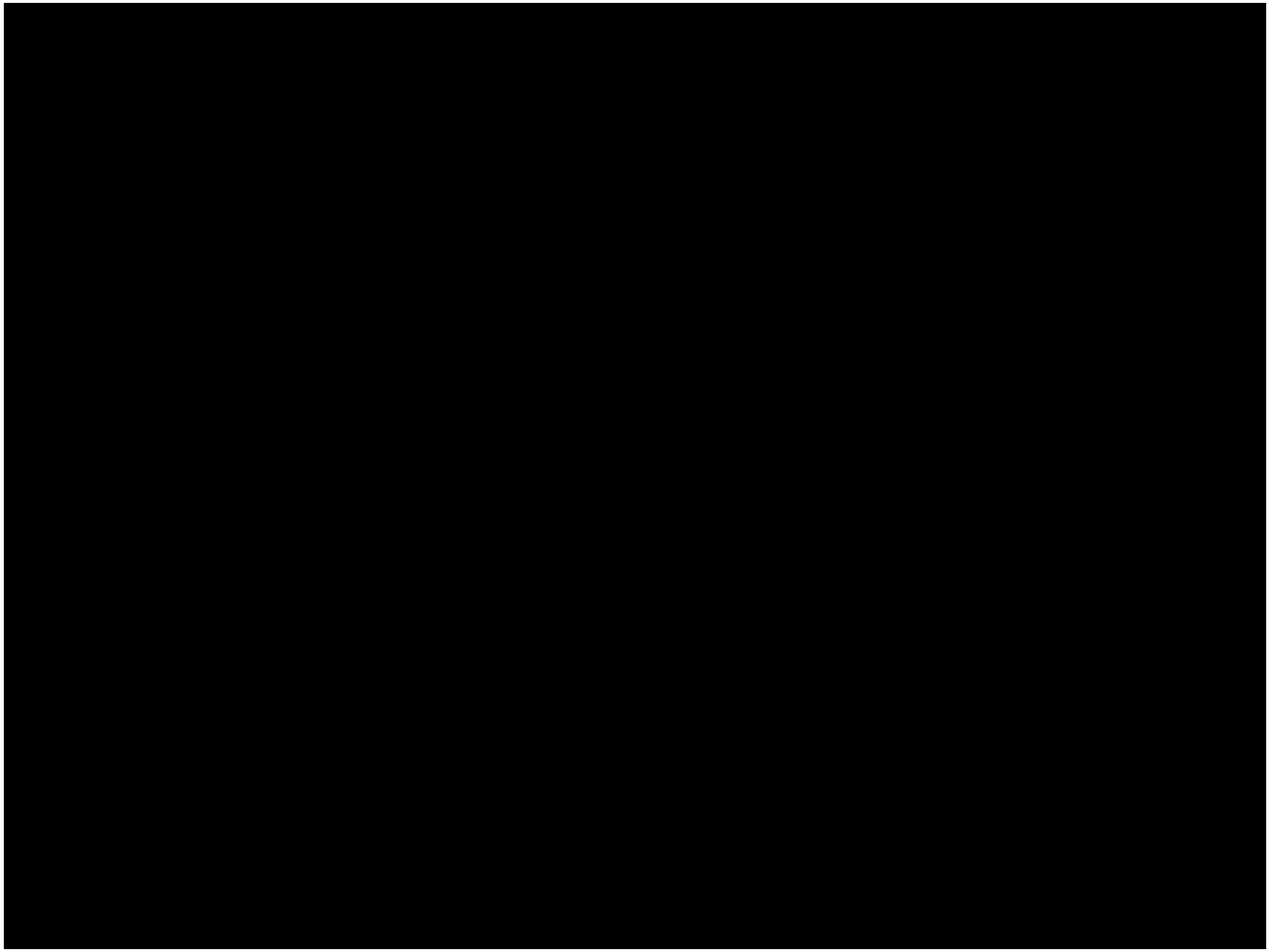
Head : réalisation de la tâche (fully-connected)



A noter : on peut tout remplacer par l'attention aujourd'hui (teaser)

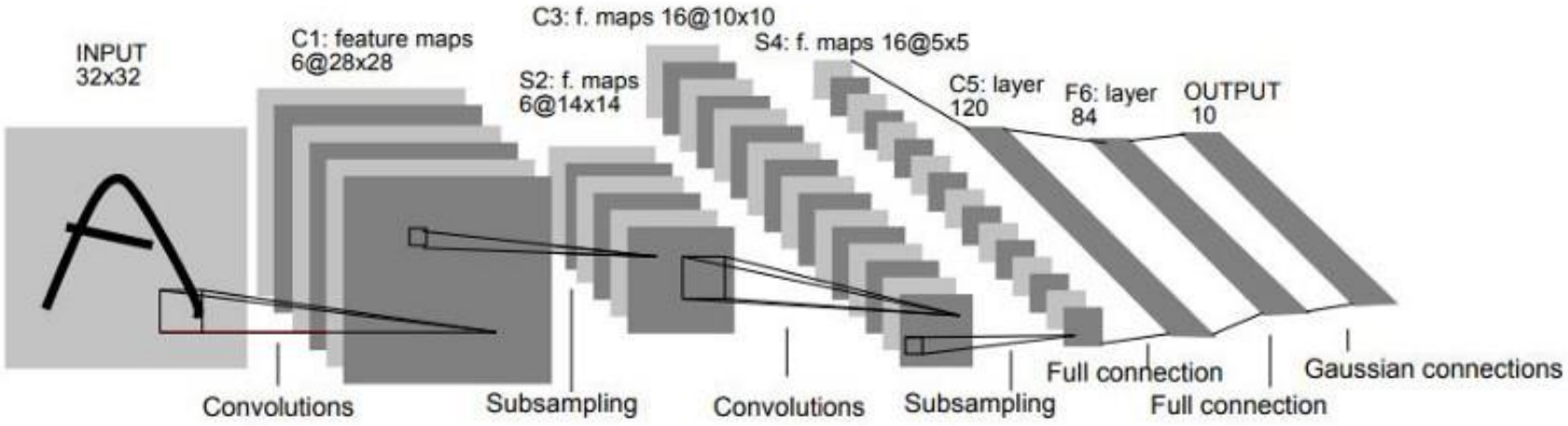
III– Les convnets

Récap’ visuel : modèle de base de vision



III– Les convnets

LeNet : 1er CNN ? *(entraîné par backpropagation en tout cas)*
[LeCun,1989]

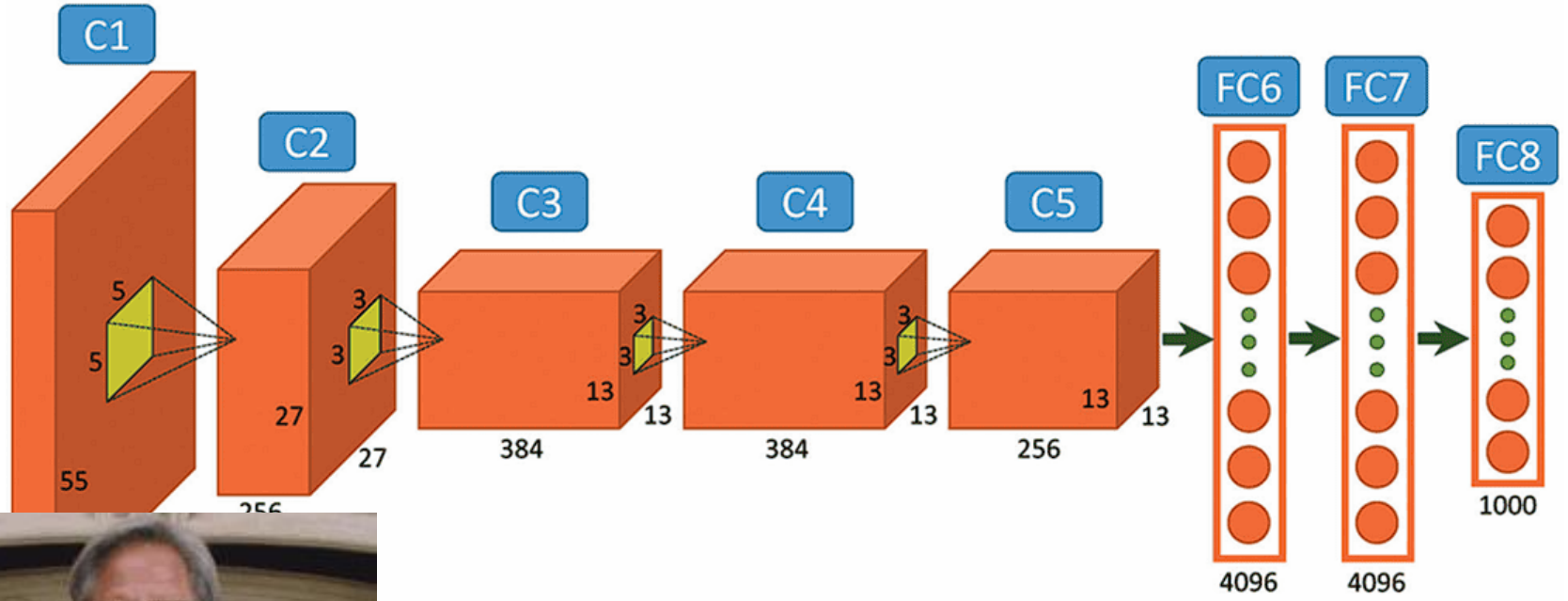


Couches de convolution « cousues mains »

III– Les convnets

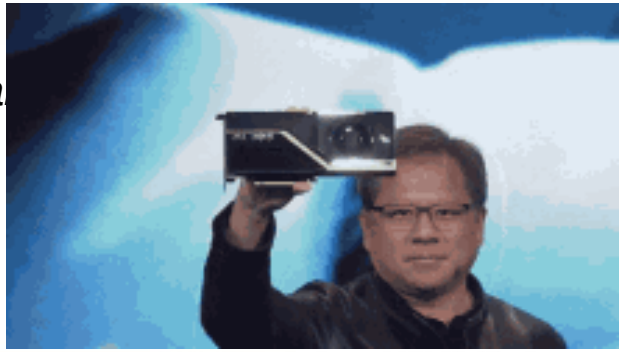
Alexnet : premières architectures « profondes » *(c'est relatif)*

[Krizhevsky et al., 2012]



Couches de convolution « cousues ma

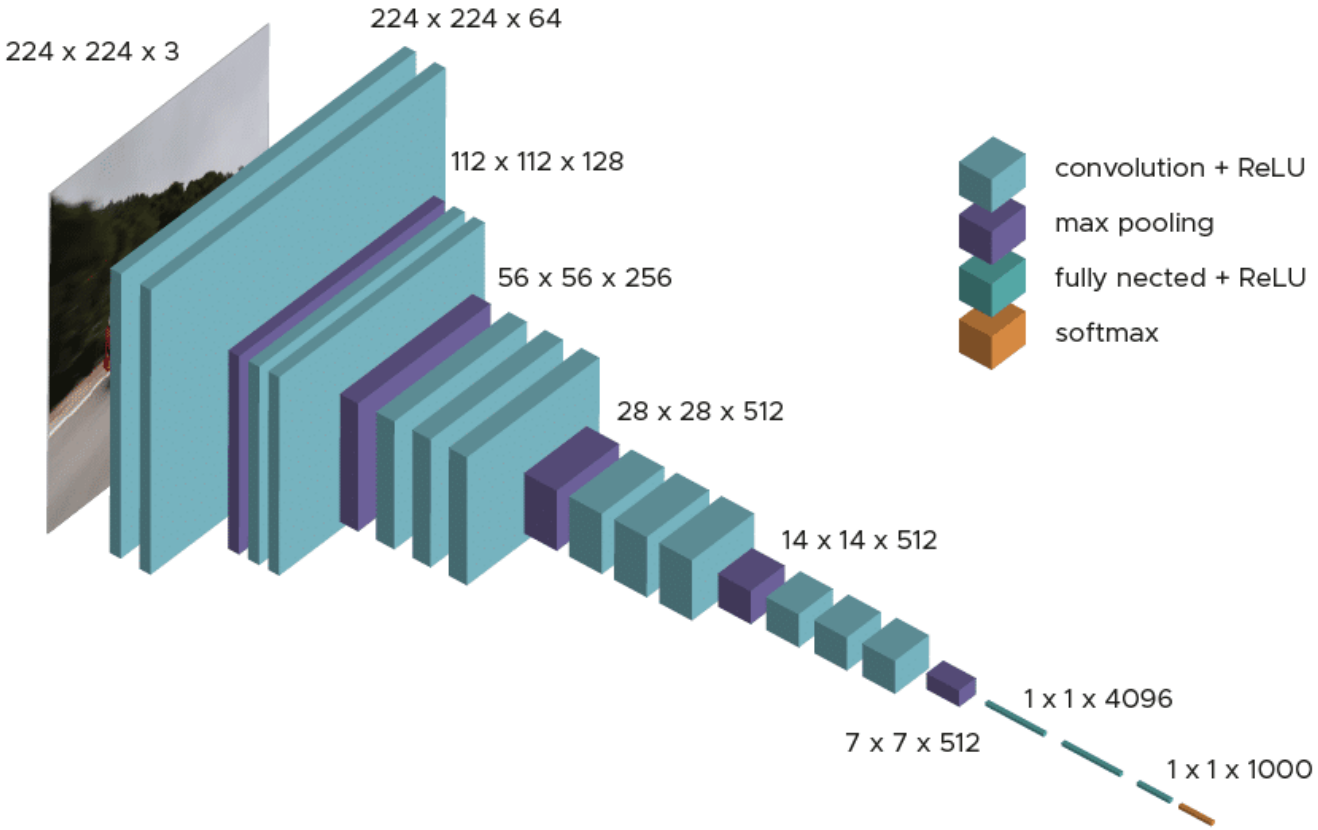
Merci NVIDIA



III– Les convnets

VGG : on augmente la profondeur / on forme des blocs de convolution

[Simonyan & Zisserman, 2014]



1er architecture « modulaire et scalable » (empilable = gain de performance)

III– Resnet/Inception

Intuition statistique : plus de couche = des propriétés de plus en plus spécifiques et abstraites

III– Resnet/Inception

Intuition statistique : plus de couche = des propriétés de plus en plus spécifiques et abstraites

- Plus faciles à séparer par un MLP

III– Resnet/Inception

Intuition statistique : plus de couche = des propriétés de plus en plus spécifiques et abstraites

- **Plus faciles à séparer par un MLP**
- **Sur-paramètrisation (problème de convergence, surapprentissage)**

III– Resnet/Inception

Intuition statistique : plus de couche = des propriétés de plus en plus spécifiques et abstraites

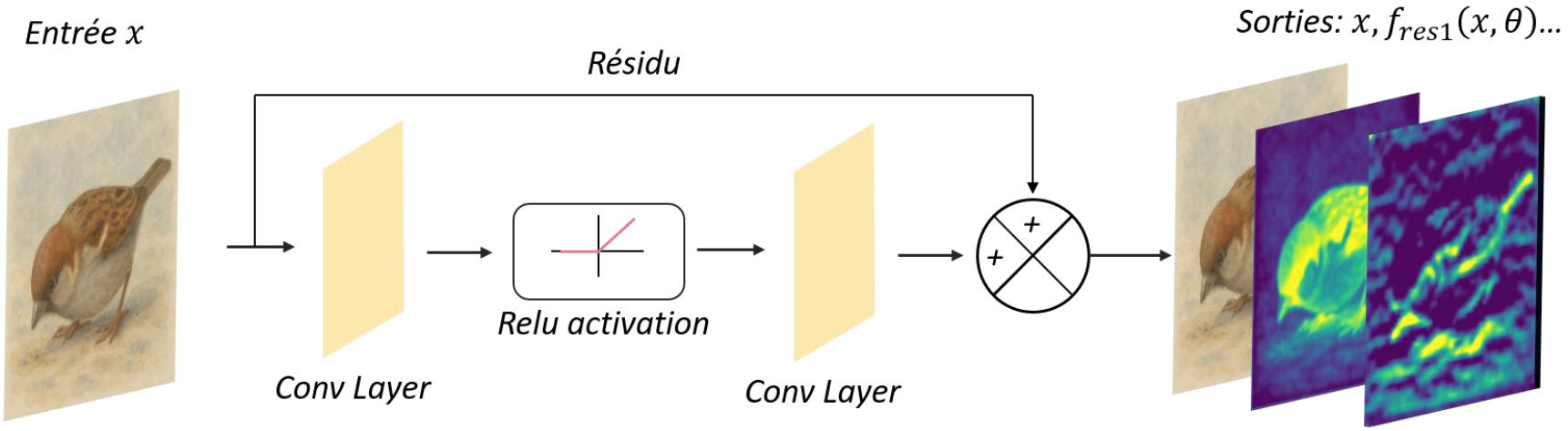
- **Plus faciles à séparer par un MLP**
- **Sur-paramètrisation (problème de convergence, surapprentissage)**
- **Epuisement du gradient (rends l'entraînement très long/impossible)**

III– Resnet/Inception

Resnet : ajouter des connexions identité

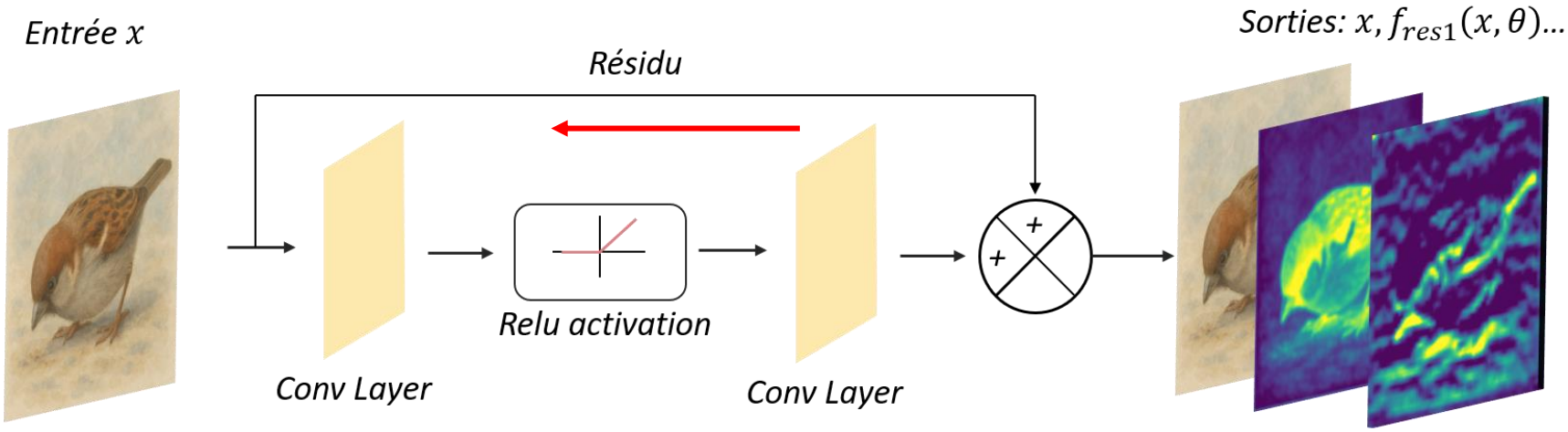
III– Resnet/Inception

Resnet : ajouter des **connexions identité, le résidu**



III– Resnet/Inception

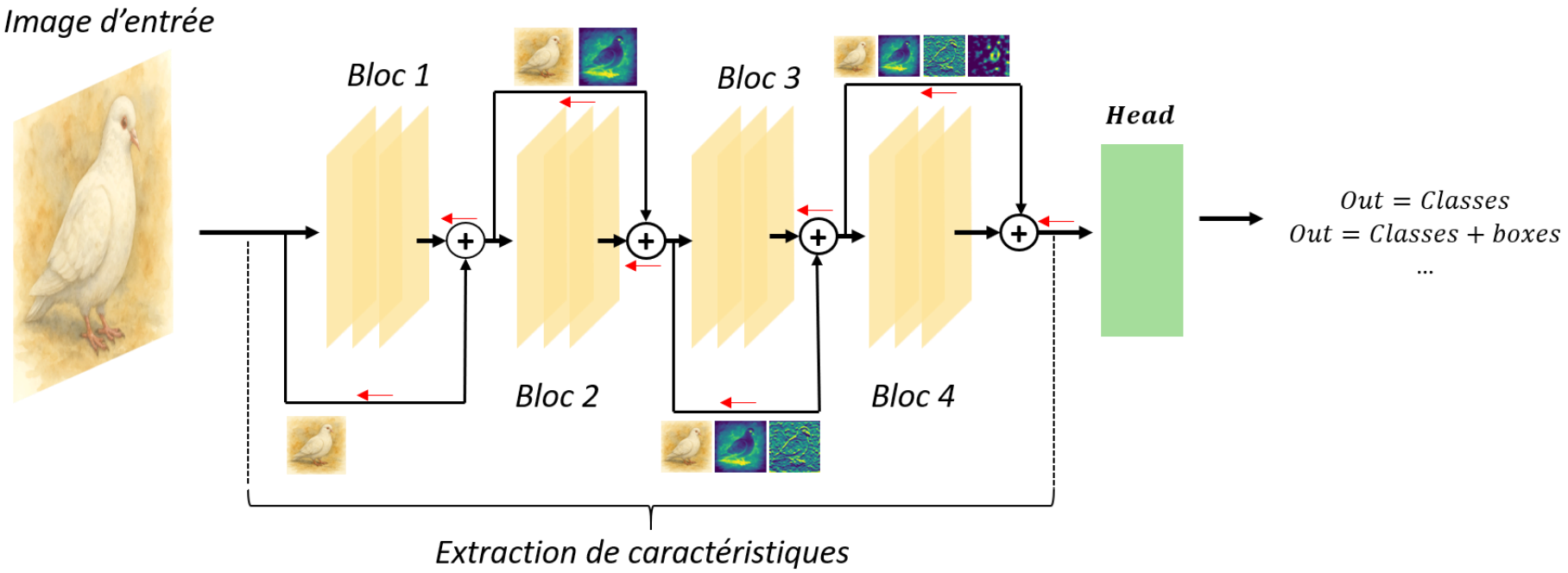
Resnet : ajouter des **connexions identité, le résidu**



Permet **un passage pour le gradient (backward)**

III– Resnet/Inception

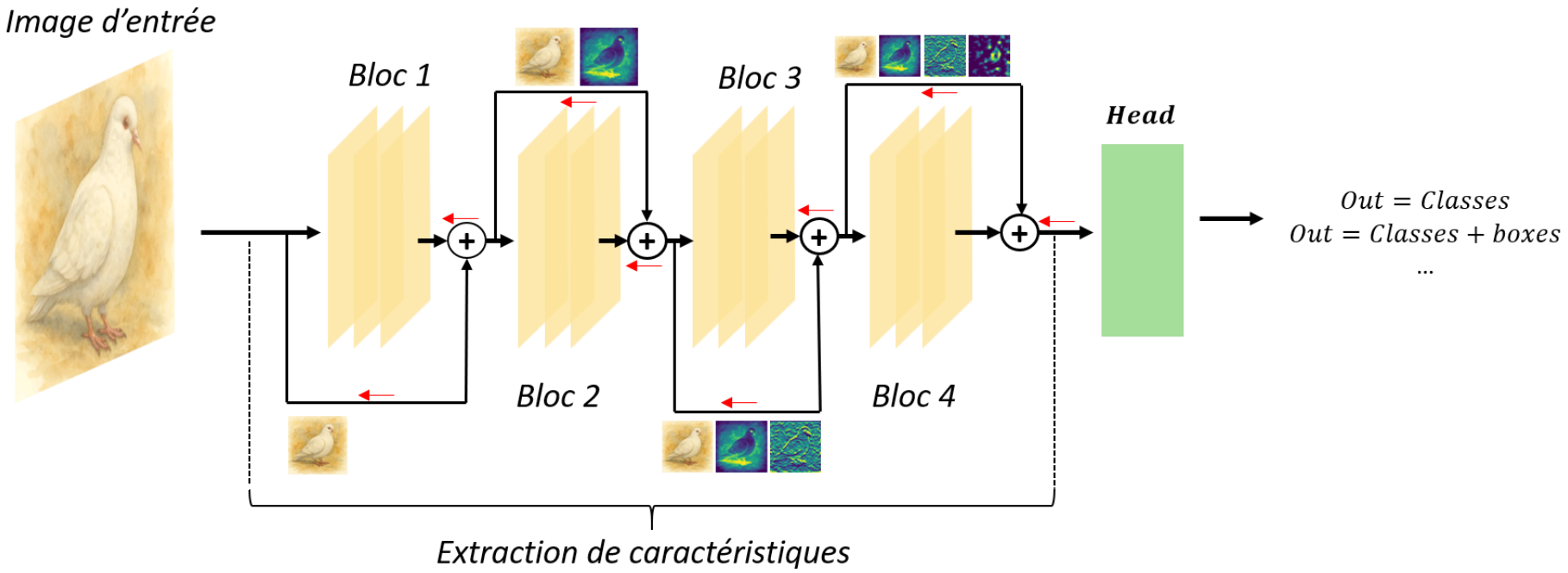
Réseau « resnet » : « blocs convolution+résidu »



III– Resnet/Inception

Réseau « resnet » : « blocs convolution+résidu »

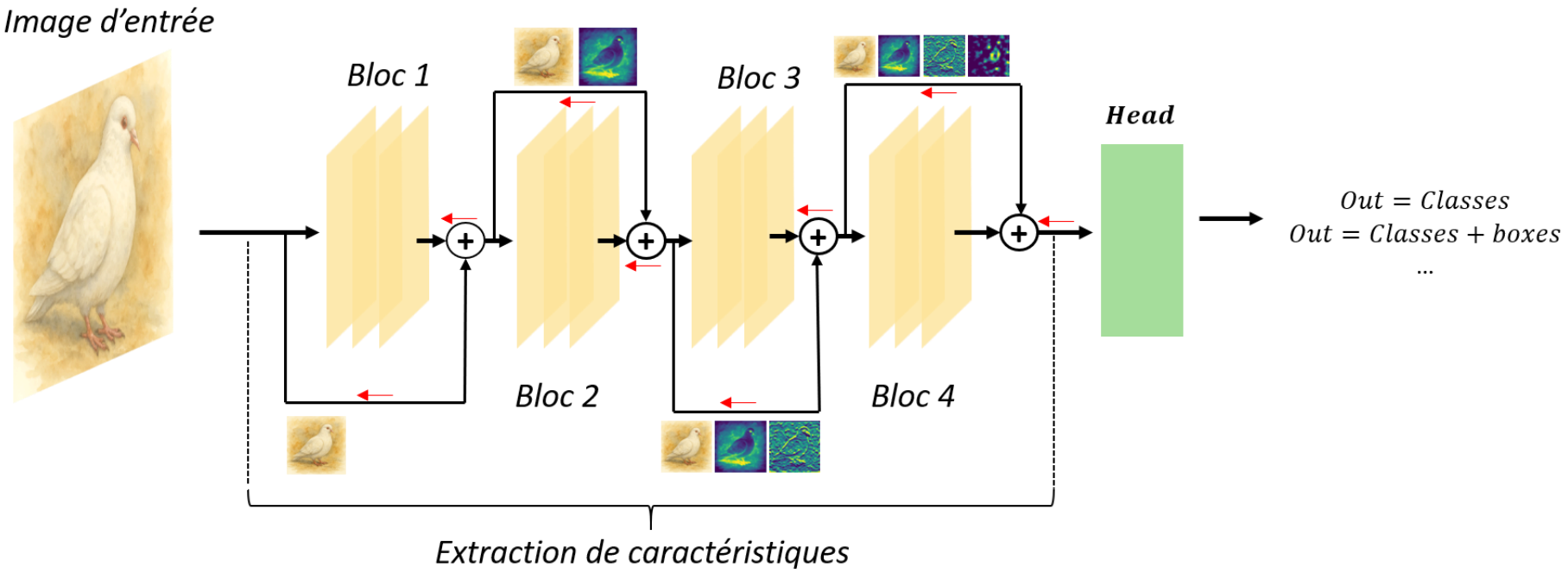
- Propagation du gradient vers l'amont plus simple



III– Resnet/Inception

Réseau « resnet » : « blocs convolution+résidu »

- **Propagation du gradient vers l'amont plus simple**
- **Ramène les propriétés image moins abstraites : combiner niveaux d'abstraction**



III– Resnet/Inception

Combiner plusieurs extractions de propriété spatiales? *(au tableau)*

III– Resnet/Inception

Combiner plusieurs extractions de propriété spatiales?

- Réseaux GoogleNet, Inception, Densenet ... Additionner des blocs de noyaux différents

III– Resnet/Inception

Combiner plusieurs Combiner plusieurs extractions de propriété spatiales?

- Réseaux GoogleNet, Inception, Densenet ... Additionner des blocs de noyaux différents

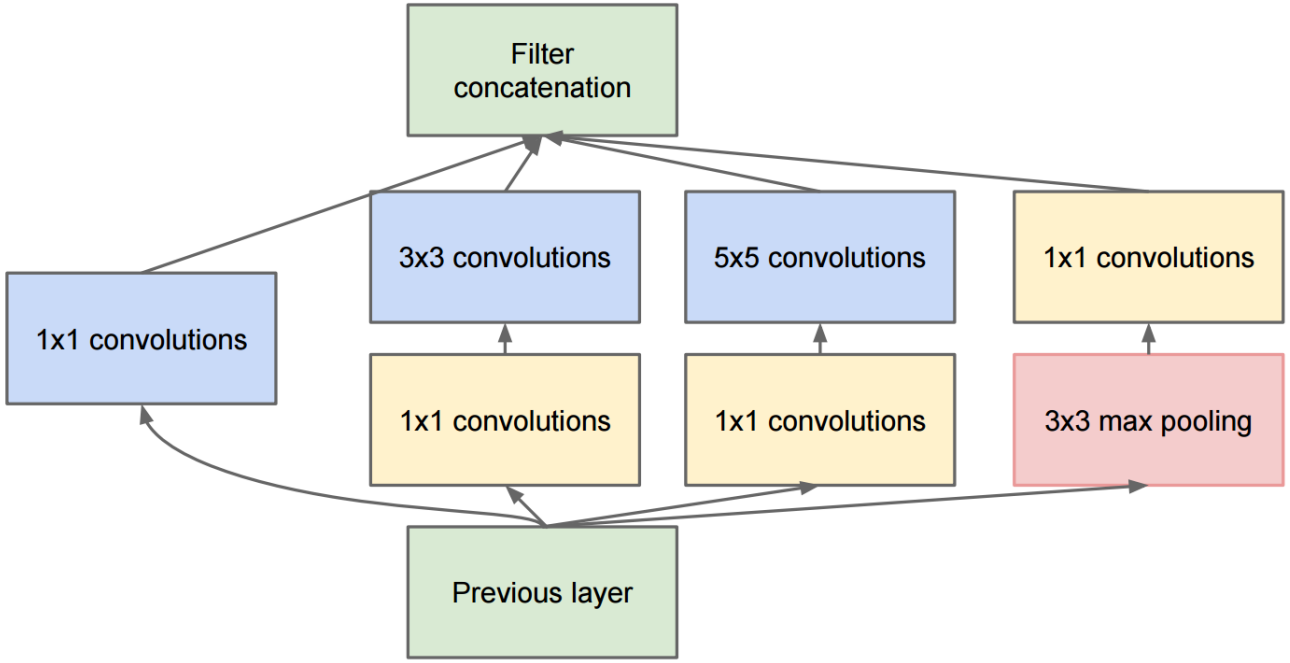
Capturer de l'information à plusieurs échelles au même degré d'abstraction

III– Resnet/Inception

Combiner plusieurs degrés d'abstraction ?

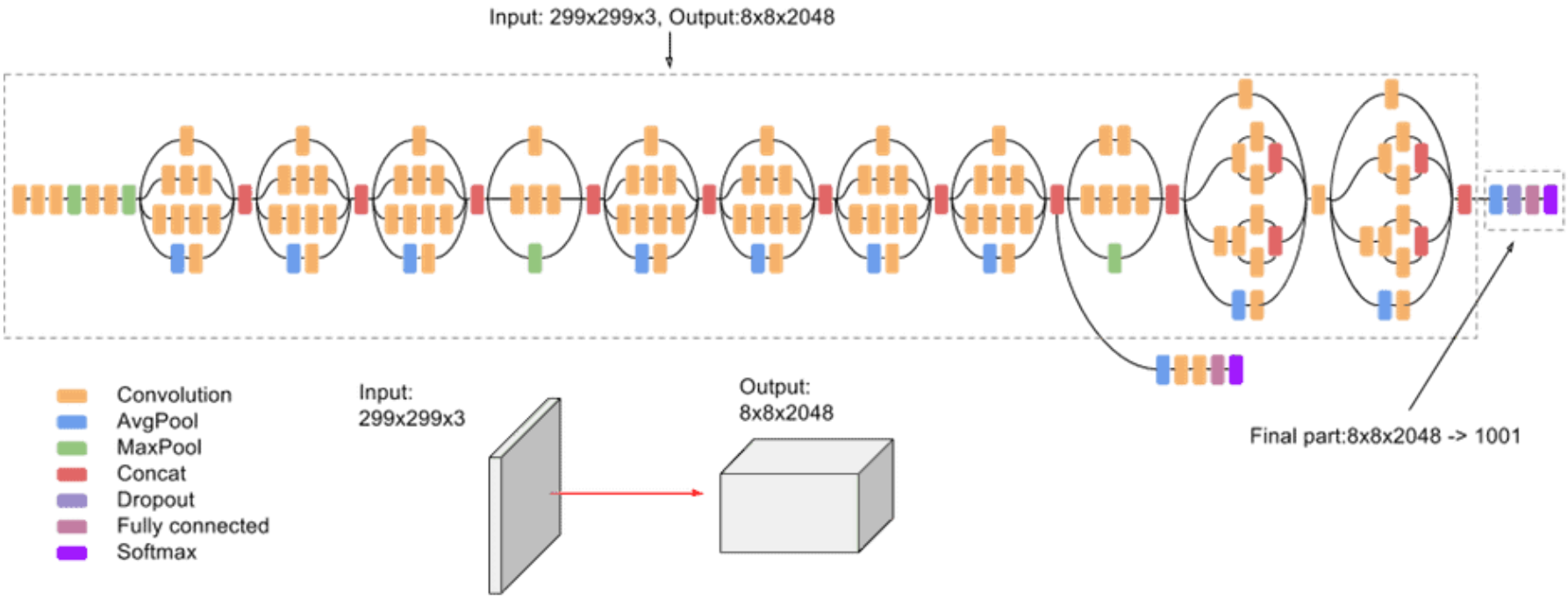
- Réseaux GoogleNet, Inception, Densenet ... Additionner des blocs de noyaux différents

Capturer de l'information à plusieurs échelles au même degré d'abstraction



III– Resnet/Inception

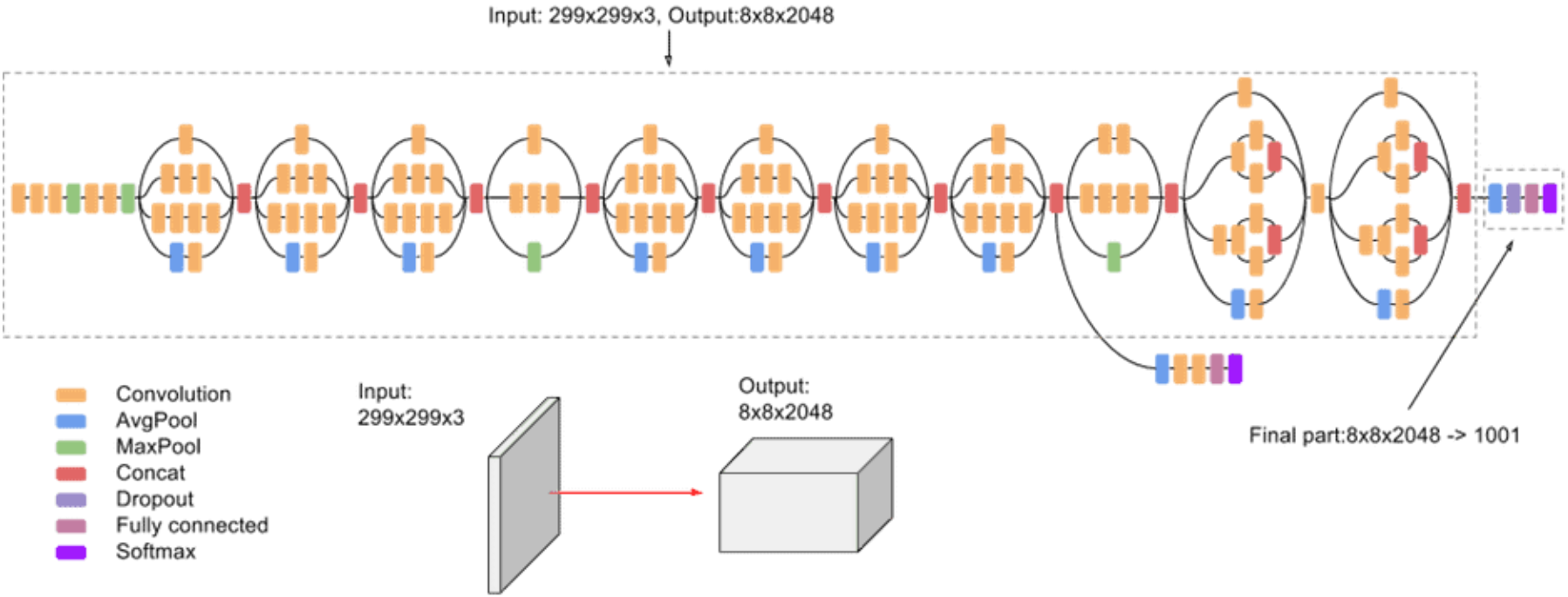
Les CNN les plus avancés (2019-2020) : **GoogleNet/Inception et le « vrai » Deep computer vision**



III– Resnet/Inception

Les CNN les plus avancés (2019-2020) : **GoogleNet/Inception et le « vrai » Deep computer vision**

Empilement assez terrifiant de filtres



IV – méthodes d'entraînement

La problématique : assez de gradient bien propagés

- **Architecture : Resnet**

IV – méthodes d'entraînement

La problématique : assez de gradient bien propagés

- **Architecture : Resnet**
- **Données : augmentation**

IV – méthodes d'entraînement

La problématique : assez de gradient bien propagés

- **Architecture : Resnet**
- **Données : augmentation**
- **Transfert d'apprentissage (transfer learning) – multi-task learning**

IV – méthodes d'entraînement

Augmentation de données « classiques »

IV – méthodes d'entraînement

Augmentation de données « classiques »

- Statistiquement : ajouter des **points** bruités **proches des points réels = mieux capturer la distribution si des données nous manquent**

Comment altérer une image ?

IV – méthodes d'entraînement

Augmentation de données « classiques » : récap' visuel

IV – méthodes d'entraînement

Intérêts supplémentaires de l'augmentation : **apprendre d'autres invariances**

- **Invariance par rotation apprise**
- **Robustesse au bruit**
- **…**

D'où **systématisation des augmentations** (même si volume de données importants)

IV – méthodes d'entraînement

Augmentation de données « classiques »

- Des pipelines assez variés … mais ce n'est pas fini !!

Augmentation de données par génération d'images (teaser séance 5 : IA générative)

IV – méthodes d'entraînement

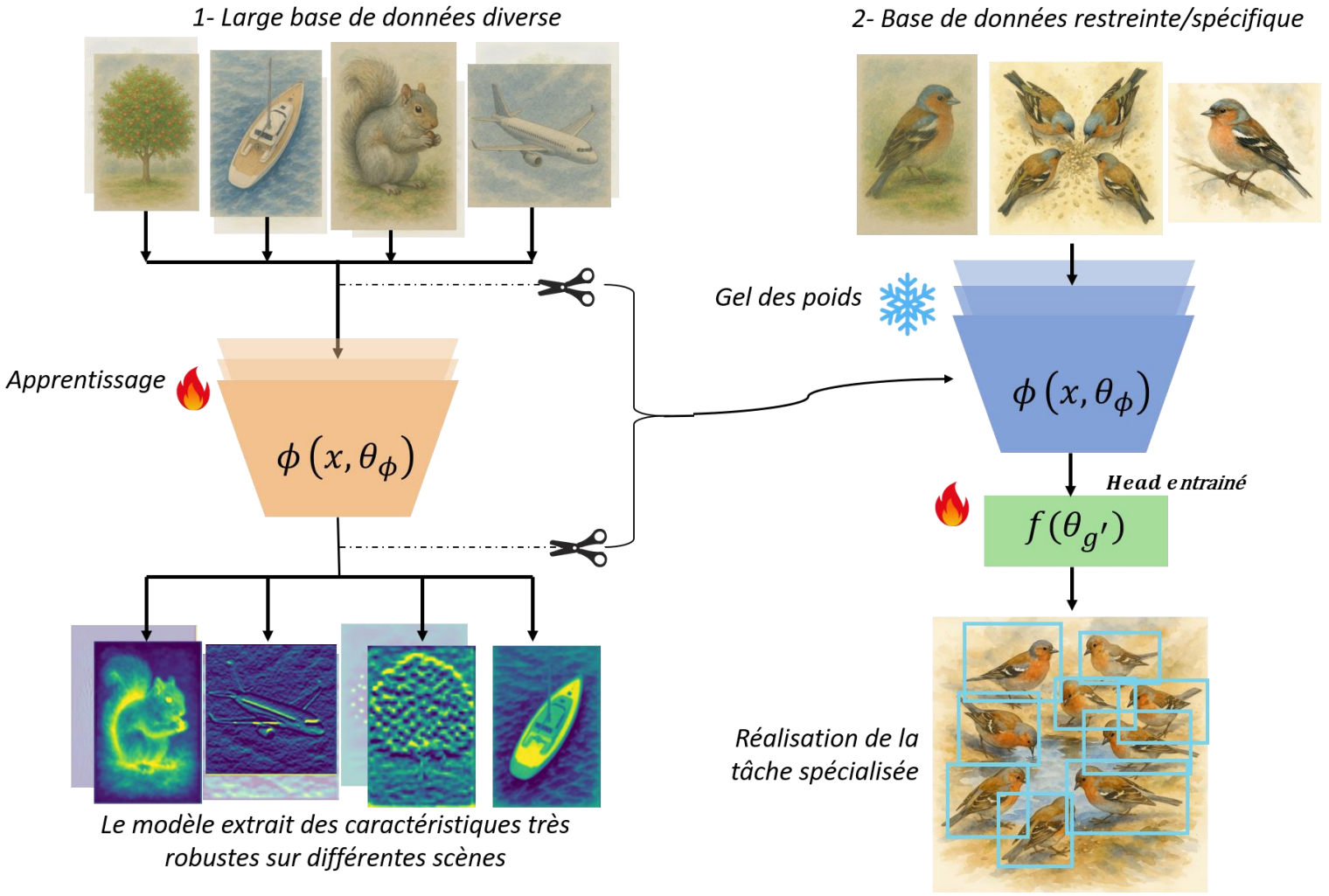
Transfer learning : apprendre sur une grande diversité de données

Si base de données suffisament variée (Imagenet) **: filtres très génériques**

Fournir des propriétés qui séparent « presque bien du premier coup » nos données

IV – méthodes d'entraînement

Transfer learning : apprendre sur une grande diversité de données



IV – méthodes d'entraînement

Transfer learning : apprendre sur une grande diversité de données

Si base de données suffisament variée (Imagenet) **: filtres très génériques**

Fournir des propriétés qui séparent « presque bien du premier coup » nos données

Reconstruction learning : transférer depuis une **tâche moins couteuse en données,** comme **la synthèse d'images**

IV – méthodes d'entraînement

Reconstruction learning : récap' visuel

IV – méthodes d'entraînement

Architectures modernes (Resnet…) + Pré-entraînement sur grands datasets + Augmentations de données – synthèse

Deep learning « sur étagère » actuel **: performances** parfois **spectaculaires sur des jeux d'images limités (-100 images)**

Attention à toujours garder des volumes de validation suffisants : risque empirique et cie …

V – Conclusion

Convolution : brique de base de l'IA de vision

- Masque/convolution : capture de **l'information spatiale des images d'entrées, Invariance par translation**
- **Paramètres techniques** : padding, stride, Batchnormalization …

Architecture convolutionnelle :

- **Empiler des filtres de convolution** = extraire des informations de plus en plus abstraites de la distrbution d'images
- **Resnets et cie** … : faciliter la prop. Du gradiend, caractéristiques multi-échelles, inter-corrélation propriétés amont (basse abstraction) – propriétés aval (haute abstraction

Entraîner les modèles de vision :

- **Augmentation de données** : ajouter des points « altérés » à la distribution originelle pour pouvoir mieux la capturer
- **Transfer learning** : apprendre sur une tâche « riche en données » et variée pour ensuite réadapter un faible nombre de paramètres sur tâche spécifique (fine-tuning)

V – Perspectives

Au-delà de juste « stacker » les propriétés multiéchelles (resnets et cie)

- **Corréler, associer les informations : les transformers et l'attention** (teaser séance 4)

Augmentation de données : employer la **synthèse d'images ?** (teaser séance 5)

Notion de modèle de fondation et SSL : architectures DINOv1-v3