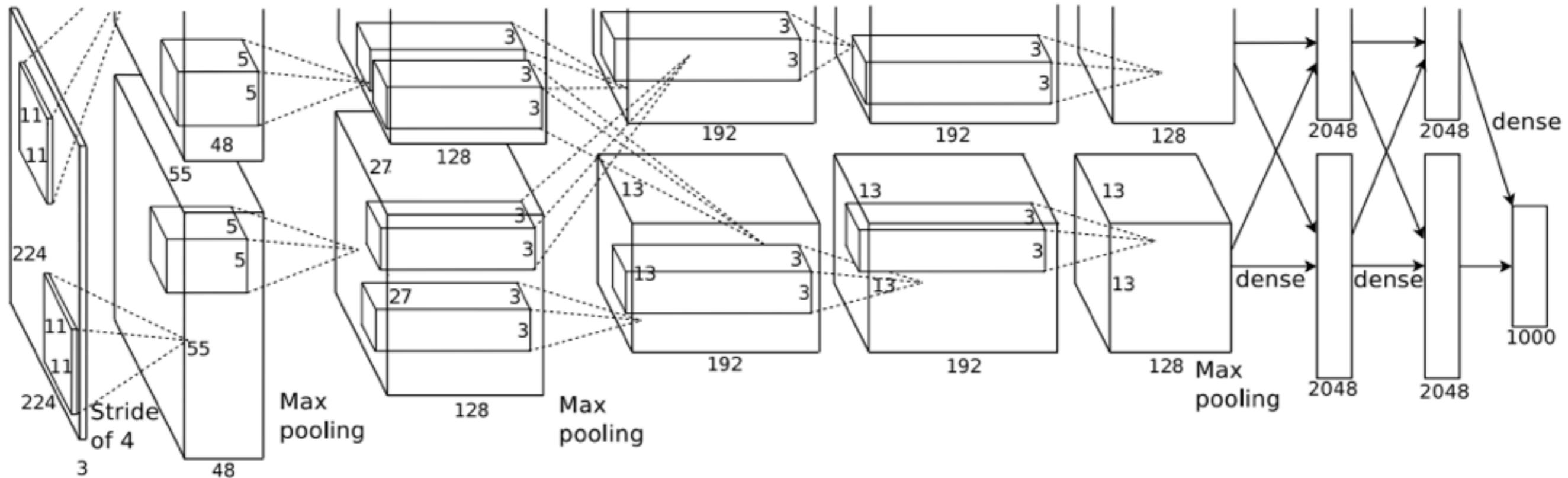


Génération d'images par apprentissage profond



GIF-4105/7105 Photographie Algorithmique, Hiver 2017
Jean-François Lalonde

Merci à Alexei Efros, James Hays, Philip Isola, Andrew Owens, Andrea Vedaldi, Derek Hoiem



facebook®

140 milliard d'images
6 milliard ajoutée à chaque mois



flickr

6 milliard d'images



the simple image sharer
imgur

1 milliard d'images
accédées par jour



You Tube

72 heures téléversées à
chaque minute



3.5 trillion photographs

90% du trafic sur Internet sera des données *visuelles*

30% des vidéos sur Youtube ont moins de 10 vues

«Digital Dark Matter»

[Perona 2010]

Défi principal

- Comment utiliser toutes ces données?
- Avec l'apprentissage profond!
- Mais tout d'abord, utilisons un exemple de système (sans «deep learning») qui utilise des données massives







Diffusion



Efros and Leung



Compléter l'image par appariement de scènes



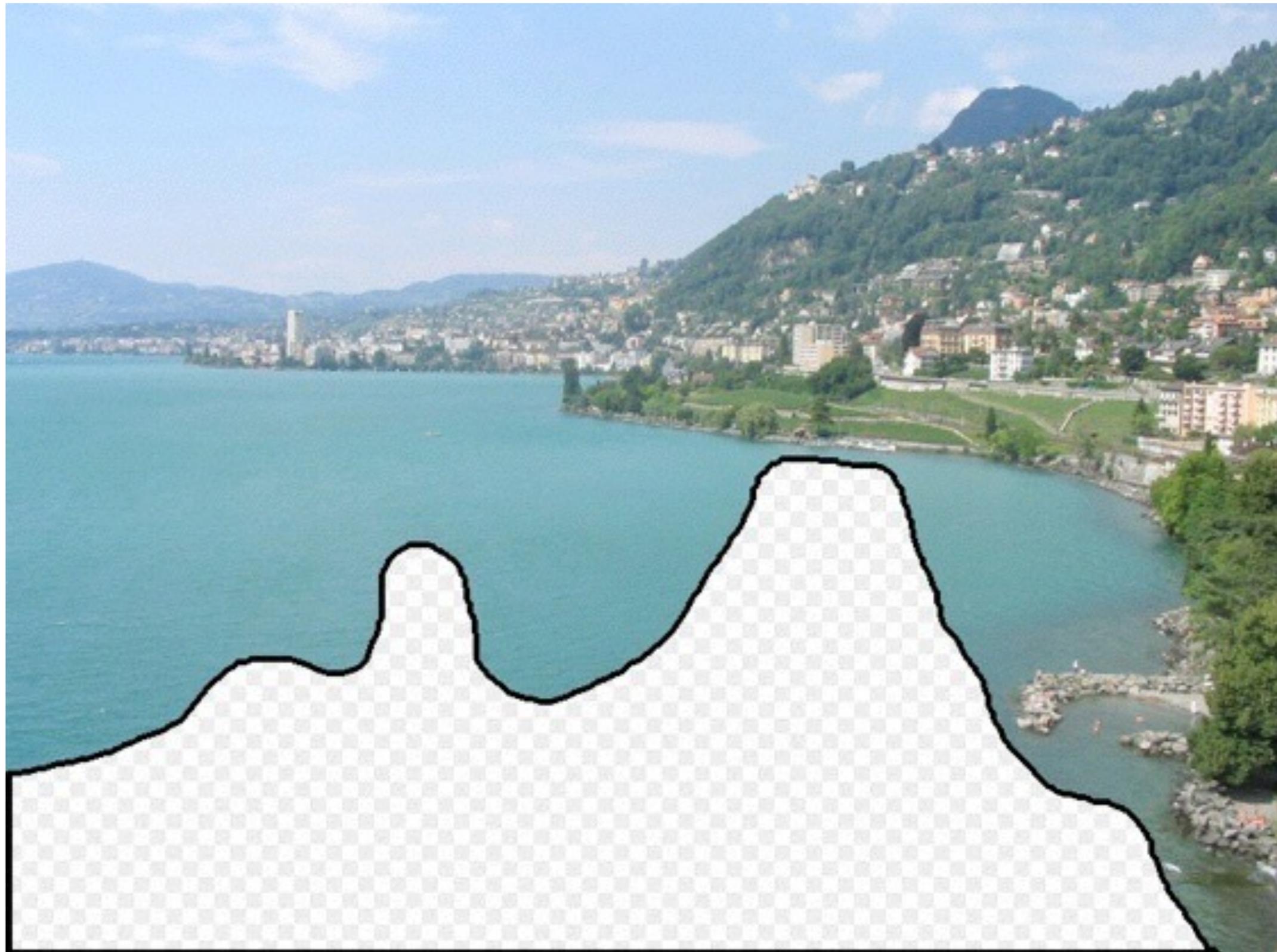


“Scene Completion”

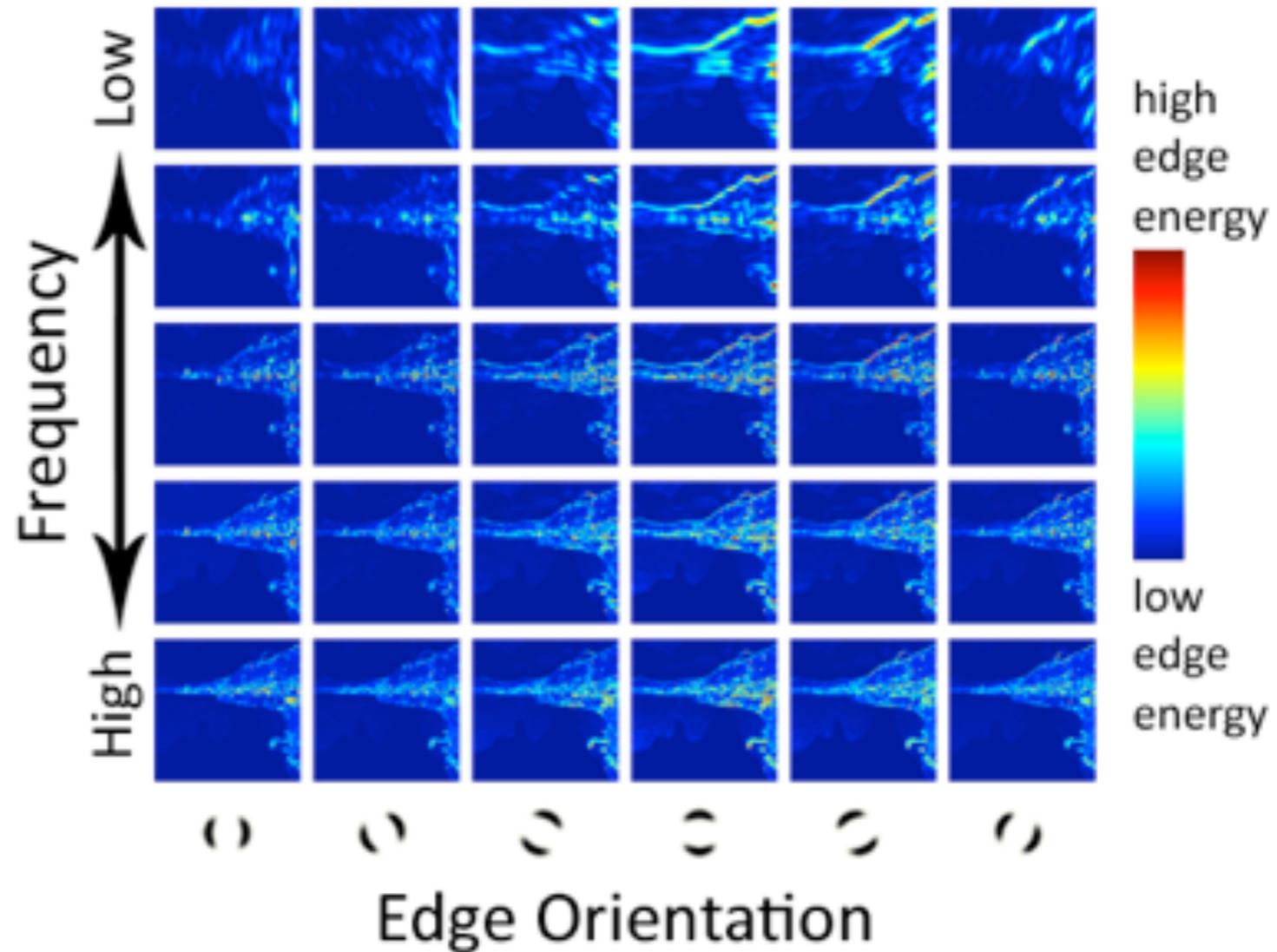
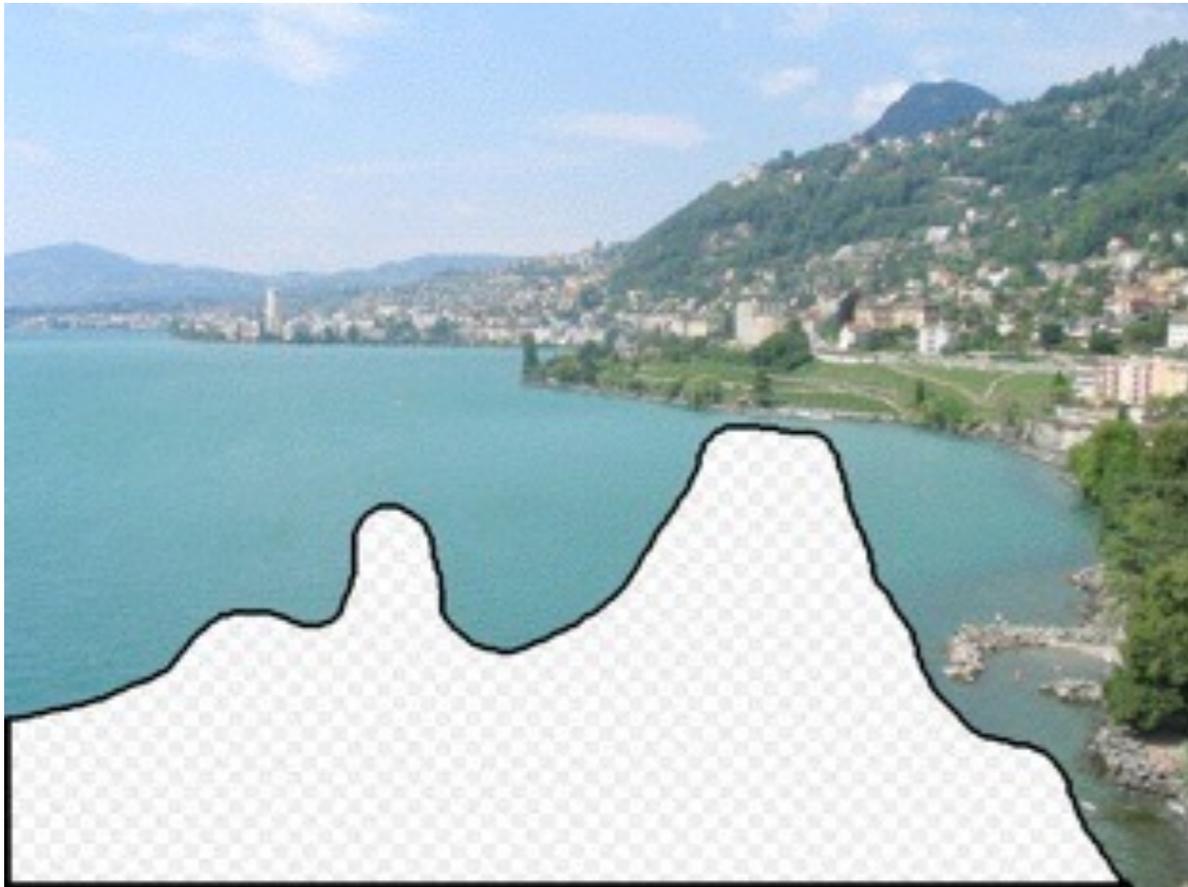
Algorithme



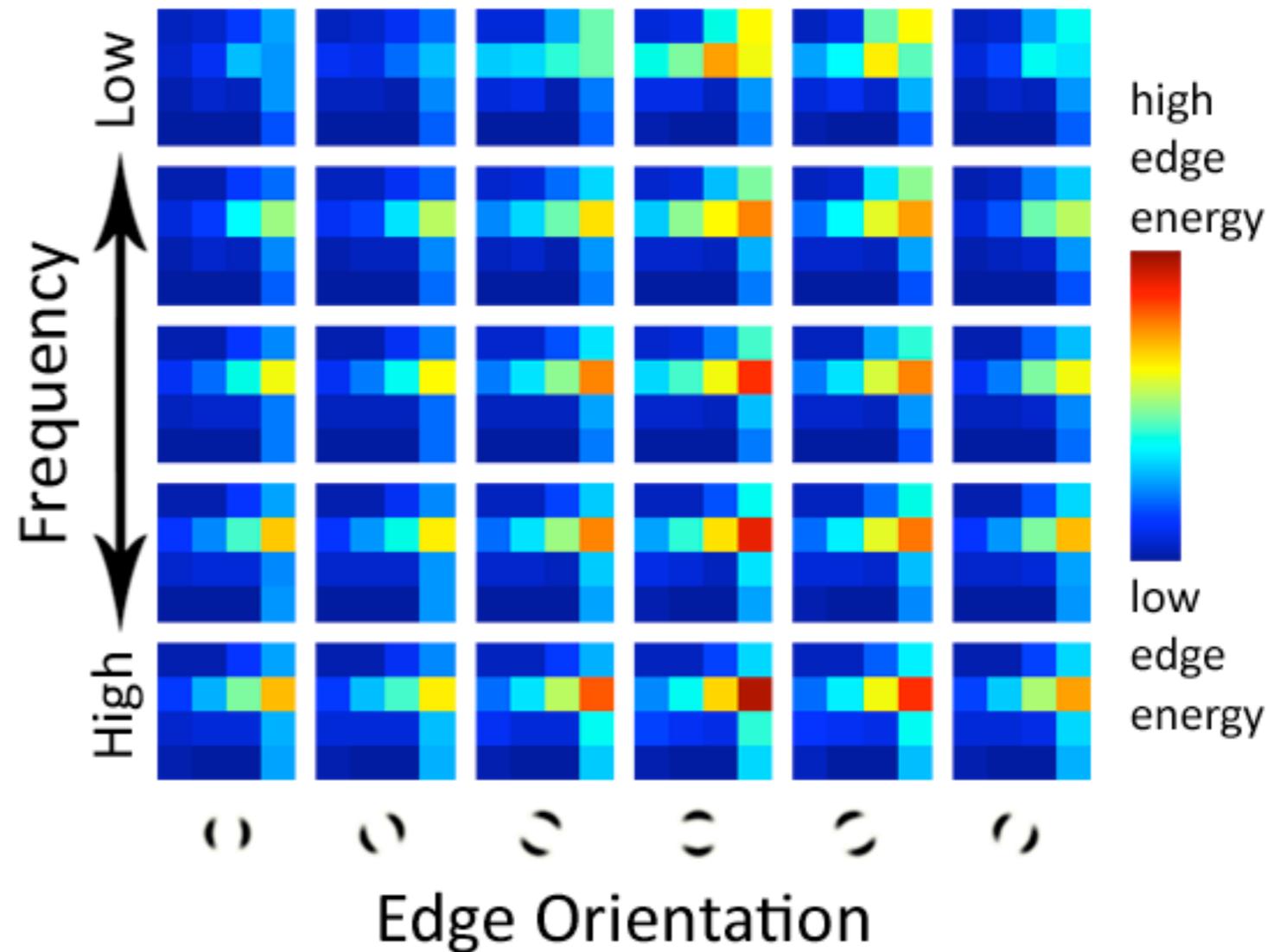
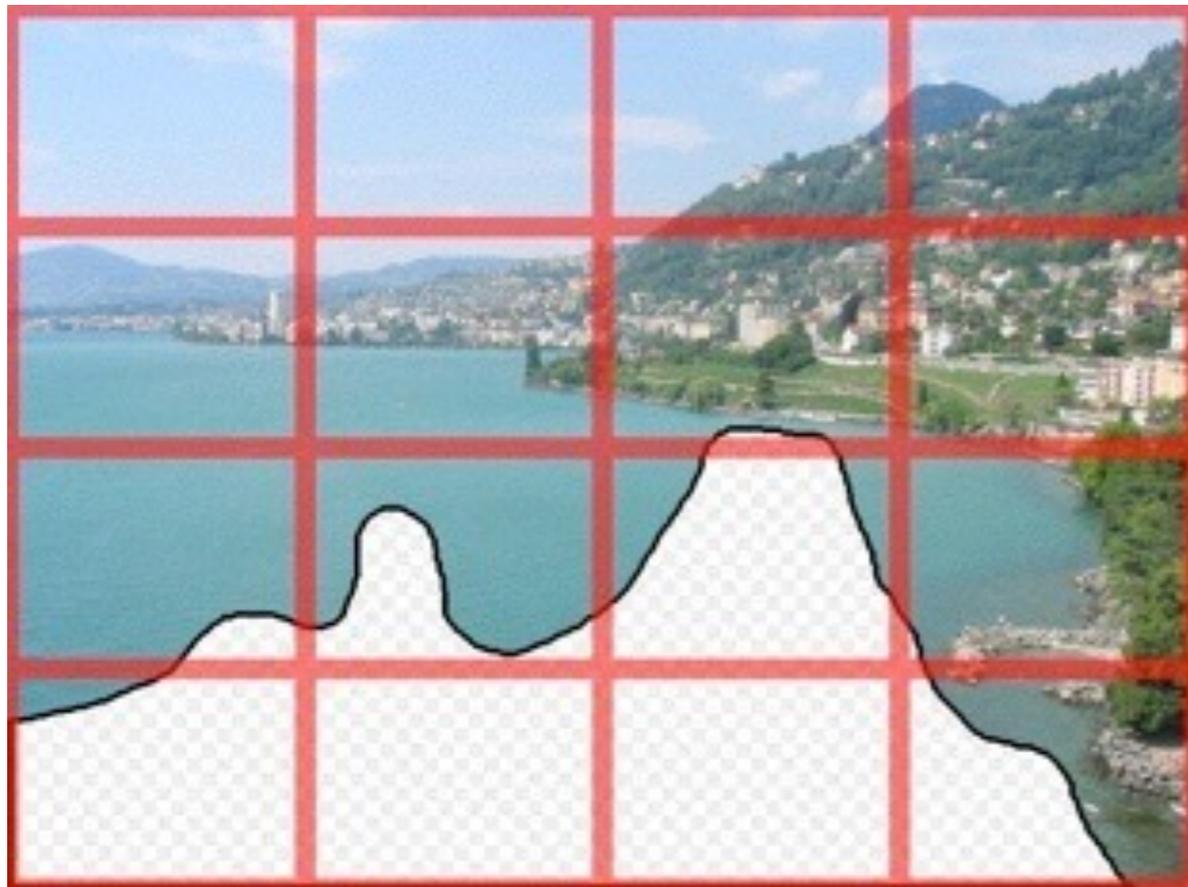
Appariement de scènes



Descripteur de scène

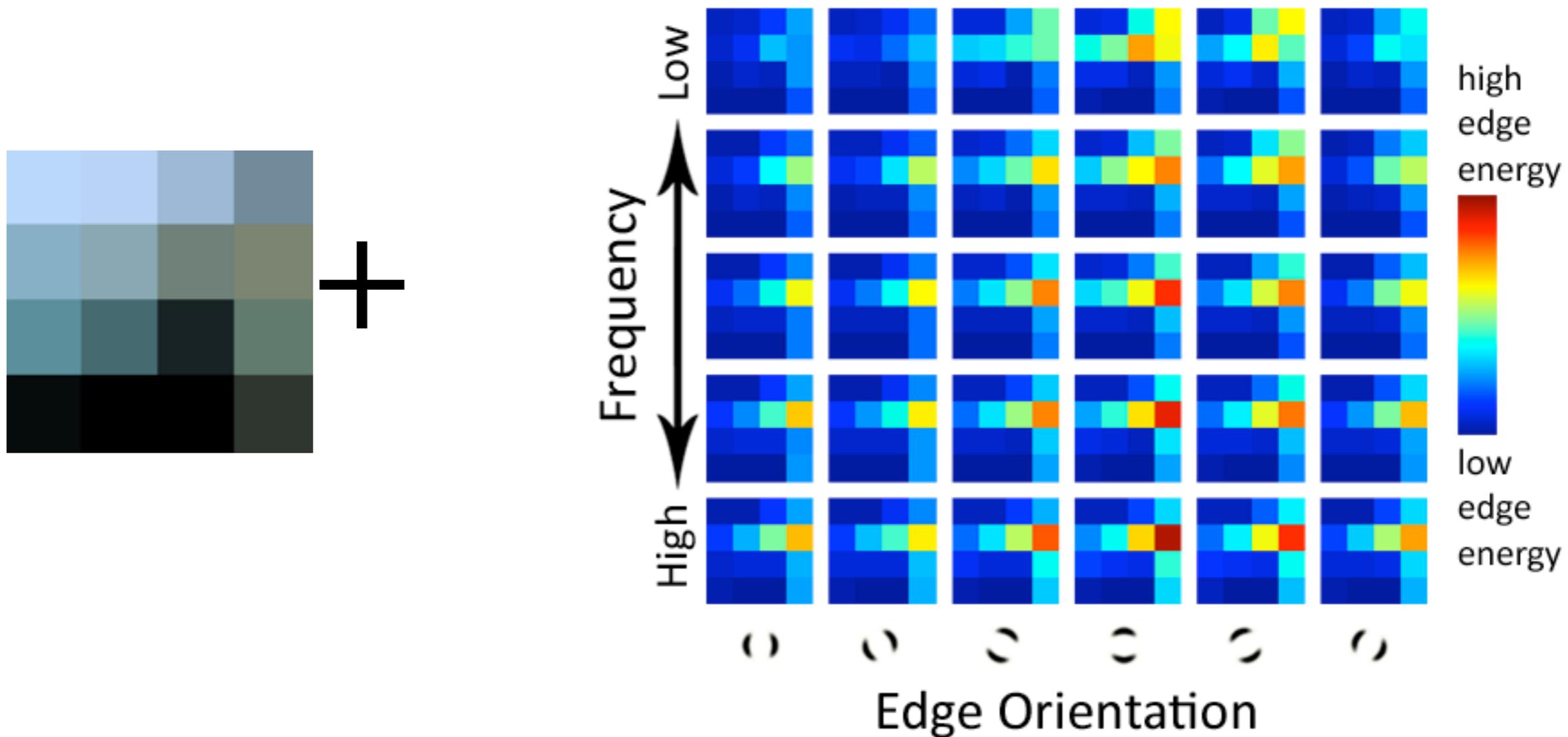


Descripteur de scène



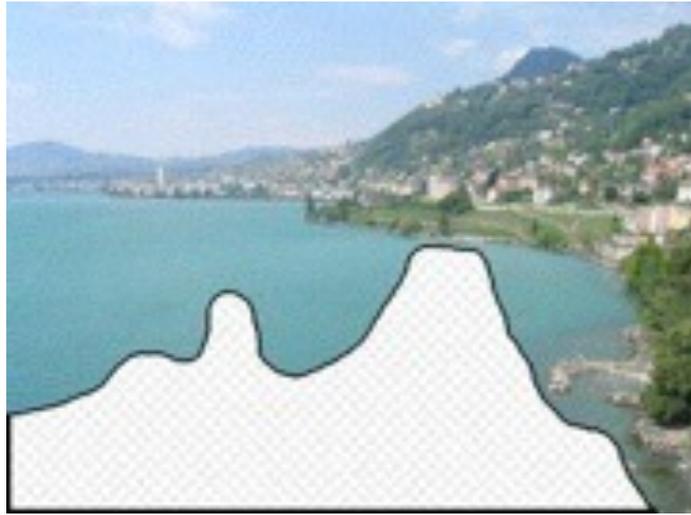
Scene Gist Descriptor
(Oliva and Torralba 2001)

Descripteur de scène



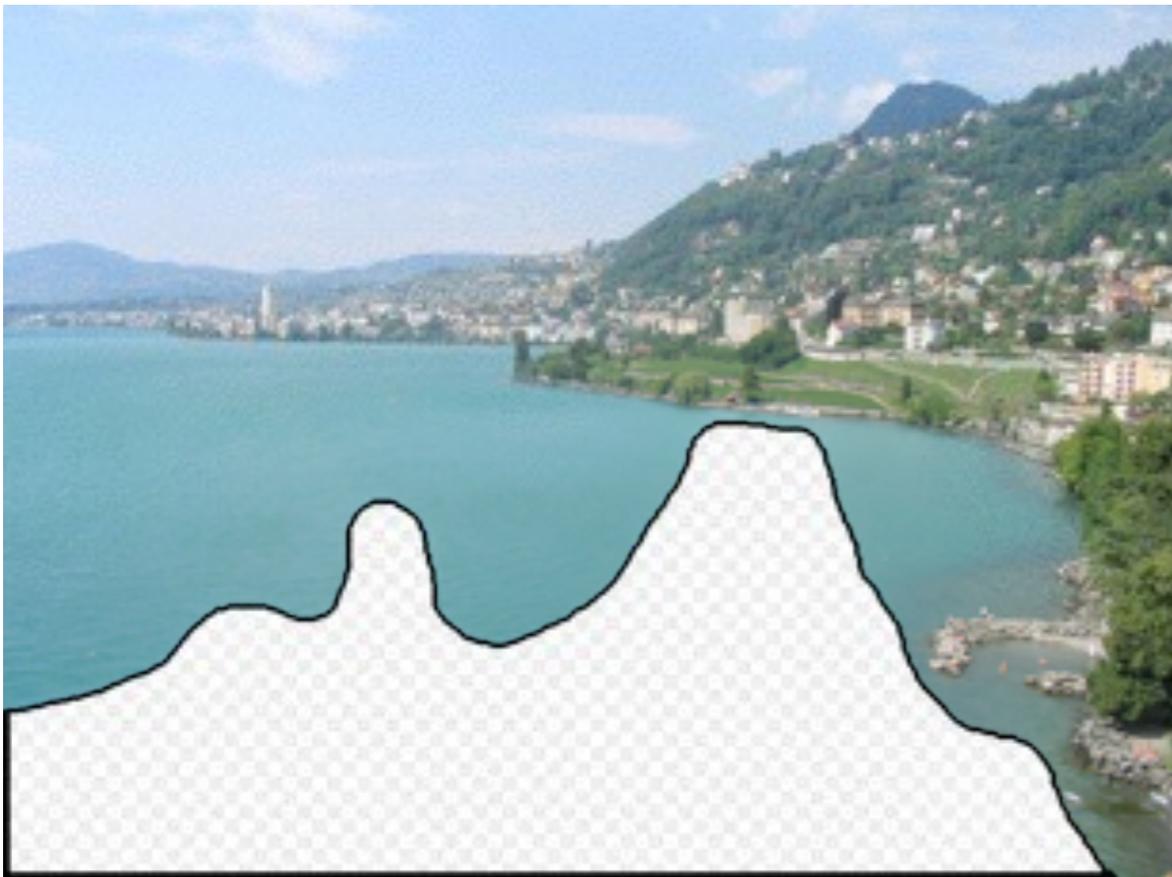
Descripteur nommé "gist"
(Oliva and Torralba 2001)

2 millions d'images de Flickr



... 200 total

Appariement local

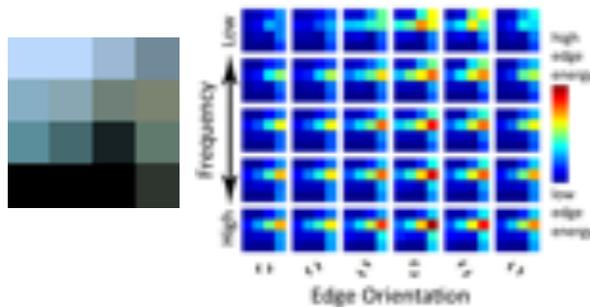




Graph cut + mélange par gradients

Ordonner les résultats

Score final est la somme de:



L'appariement de scènes



L'appariement local (couleur + texture)



Le coût de la coupure de graphe

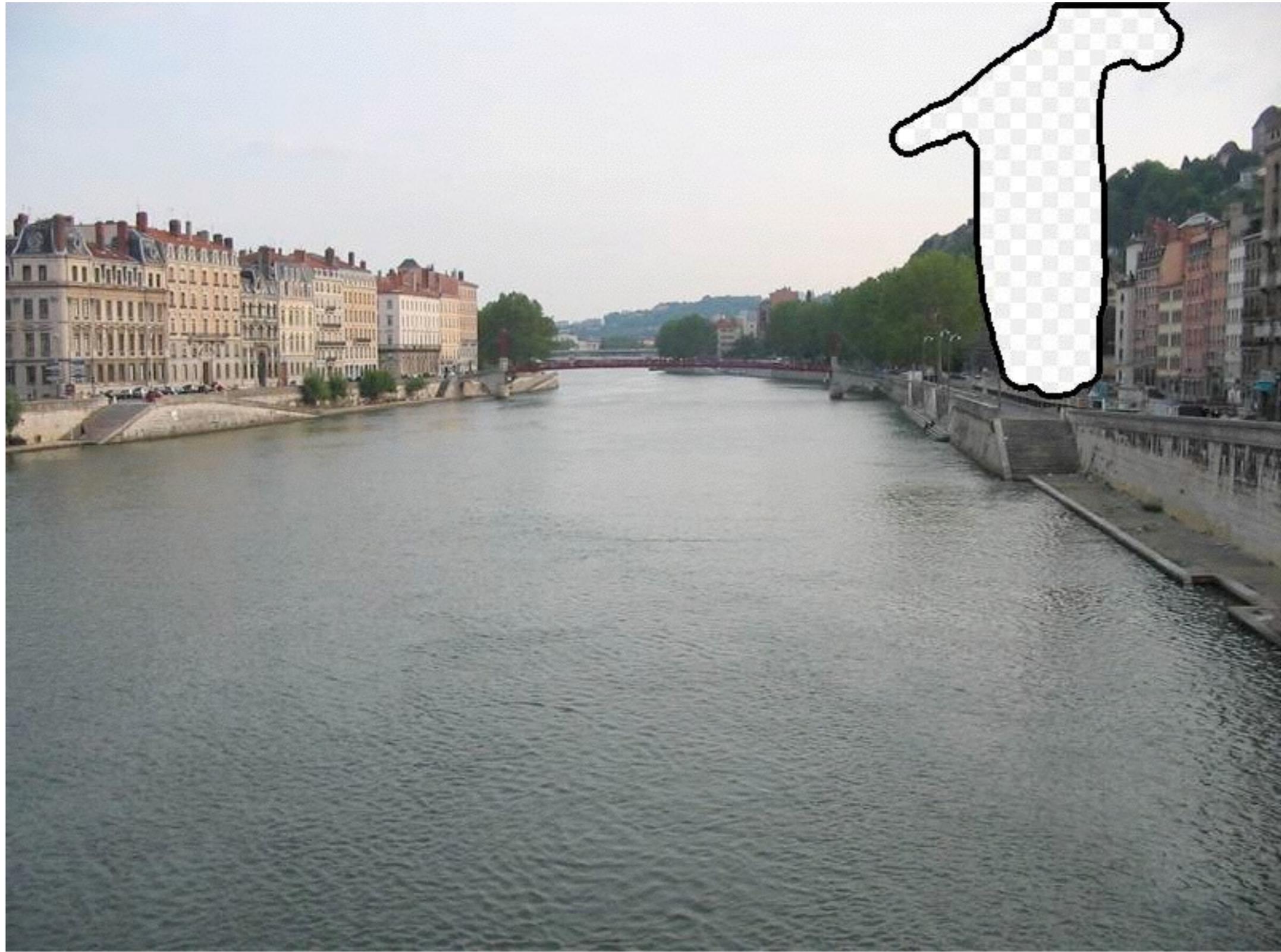




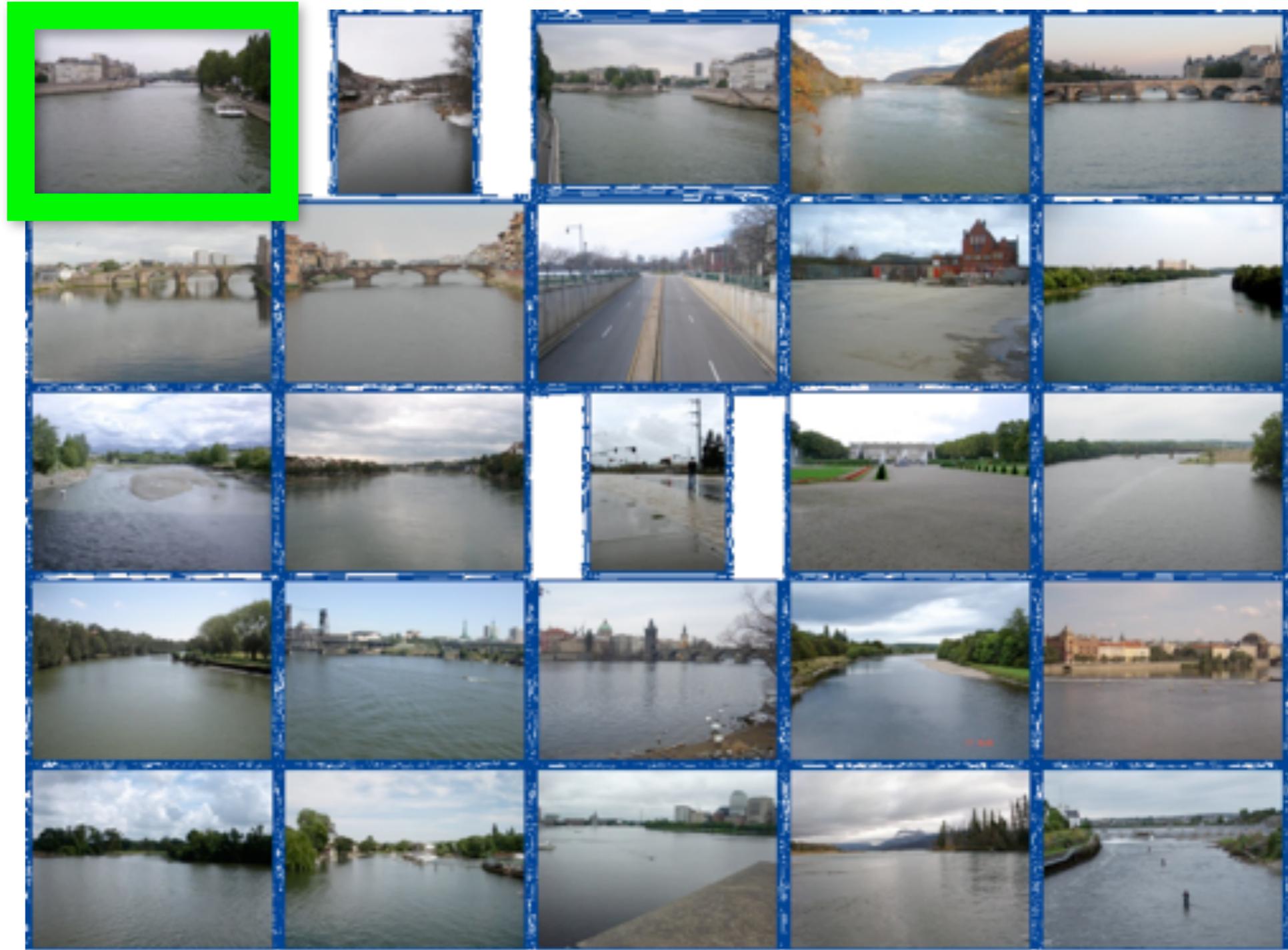








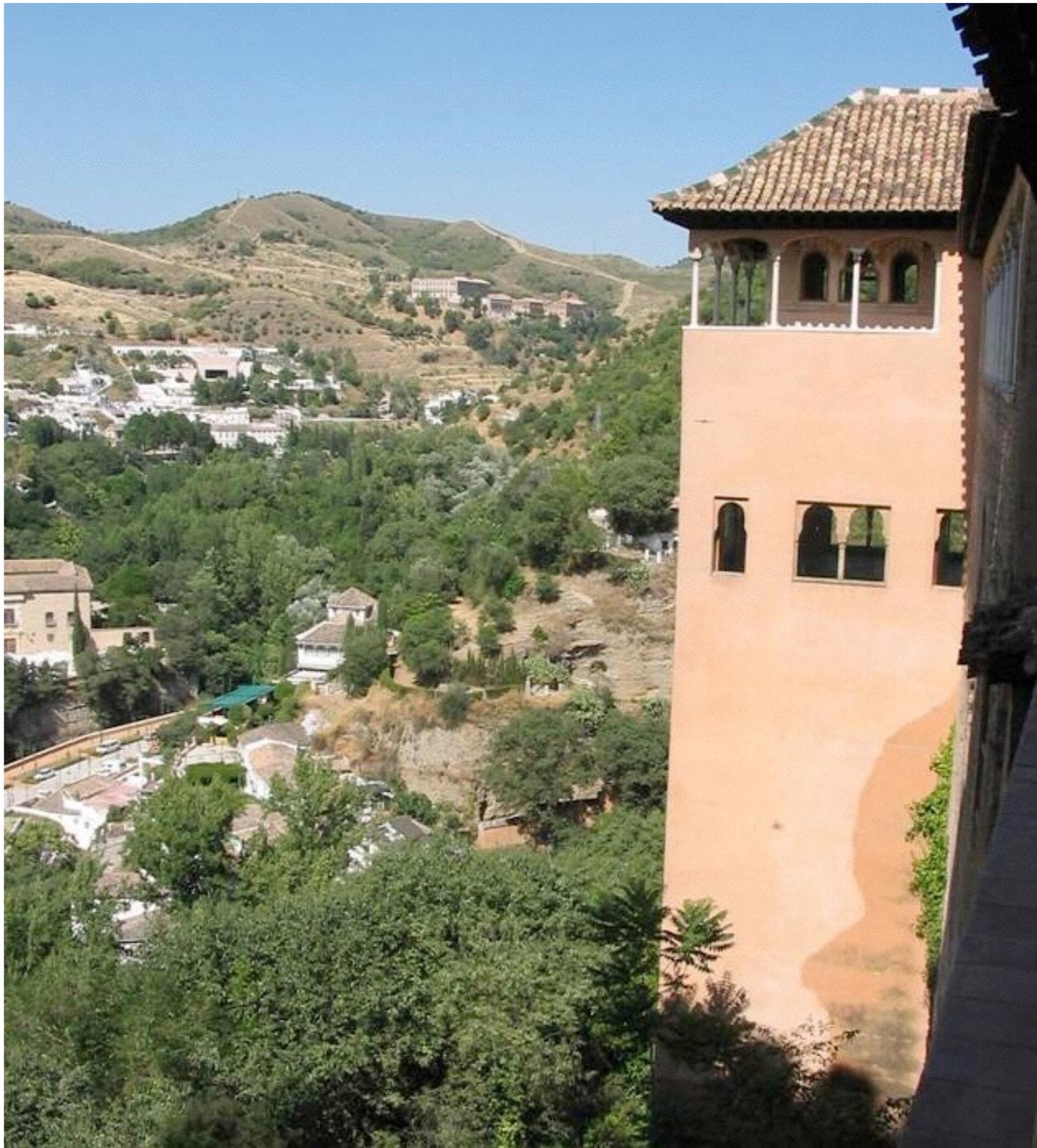




... 200 scenes





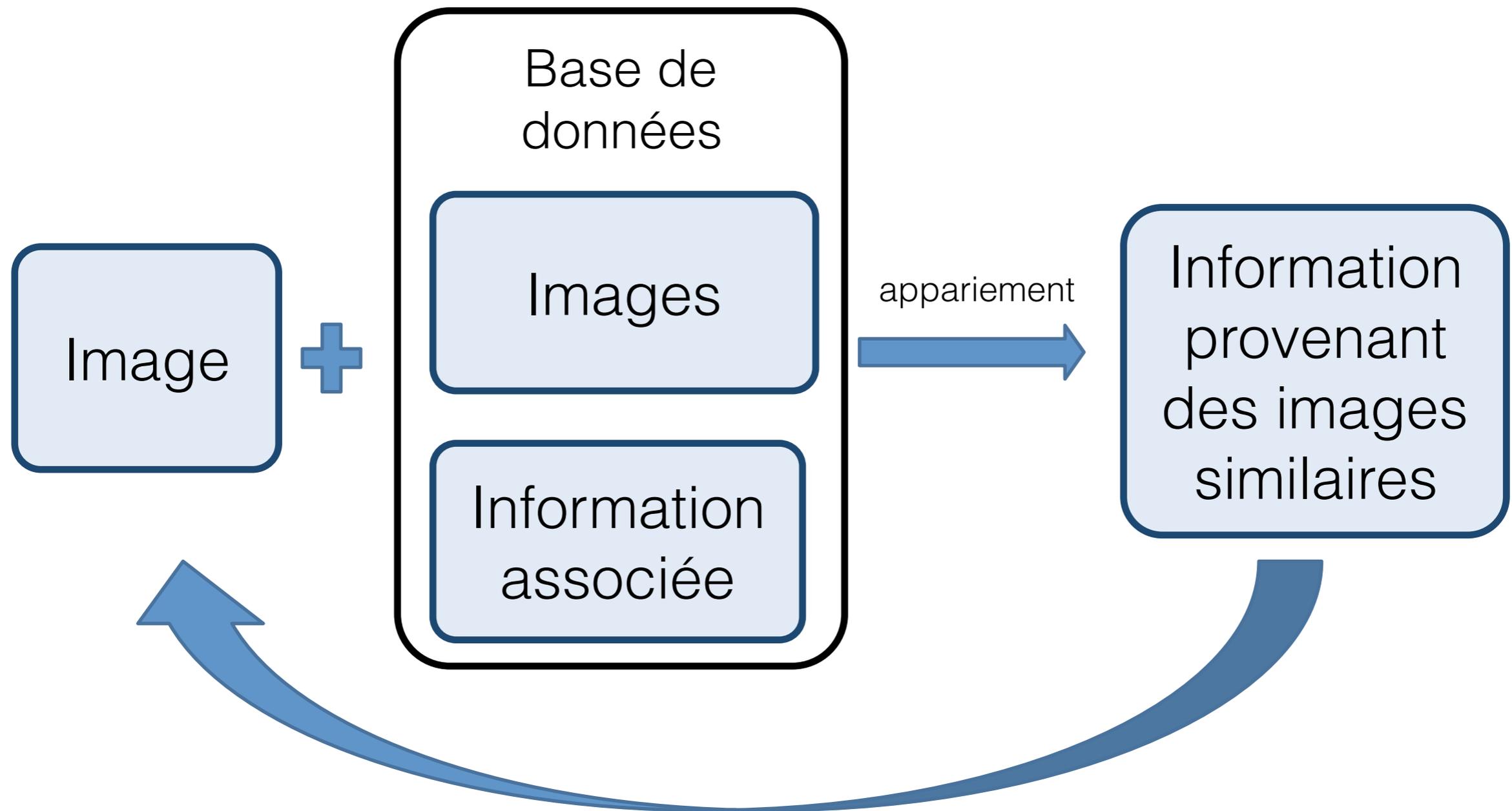






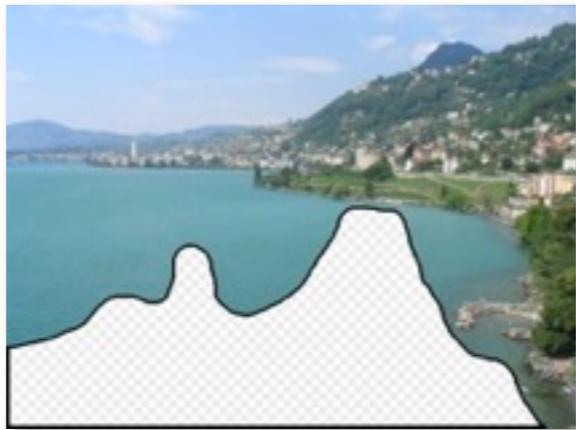


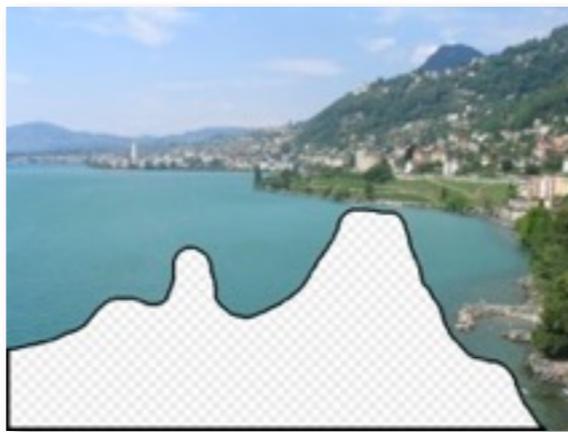
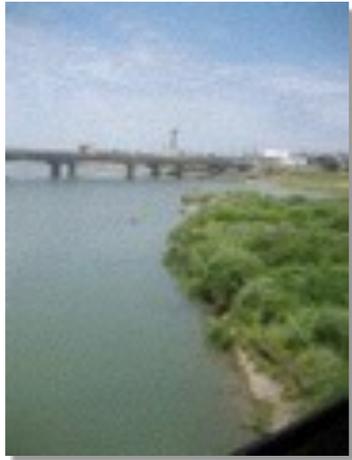
Utiliser beaucoup de données!



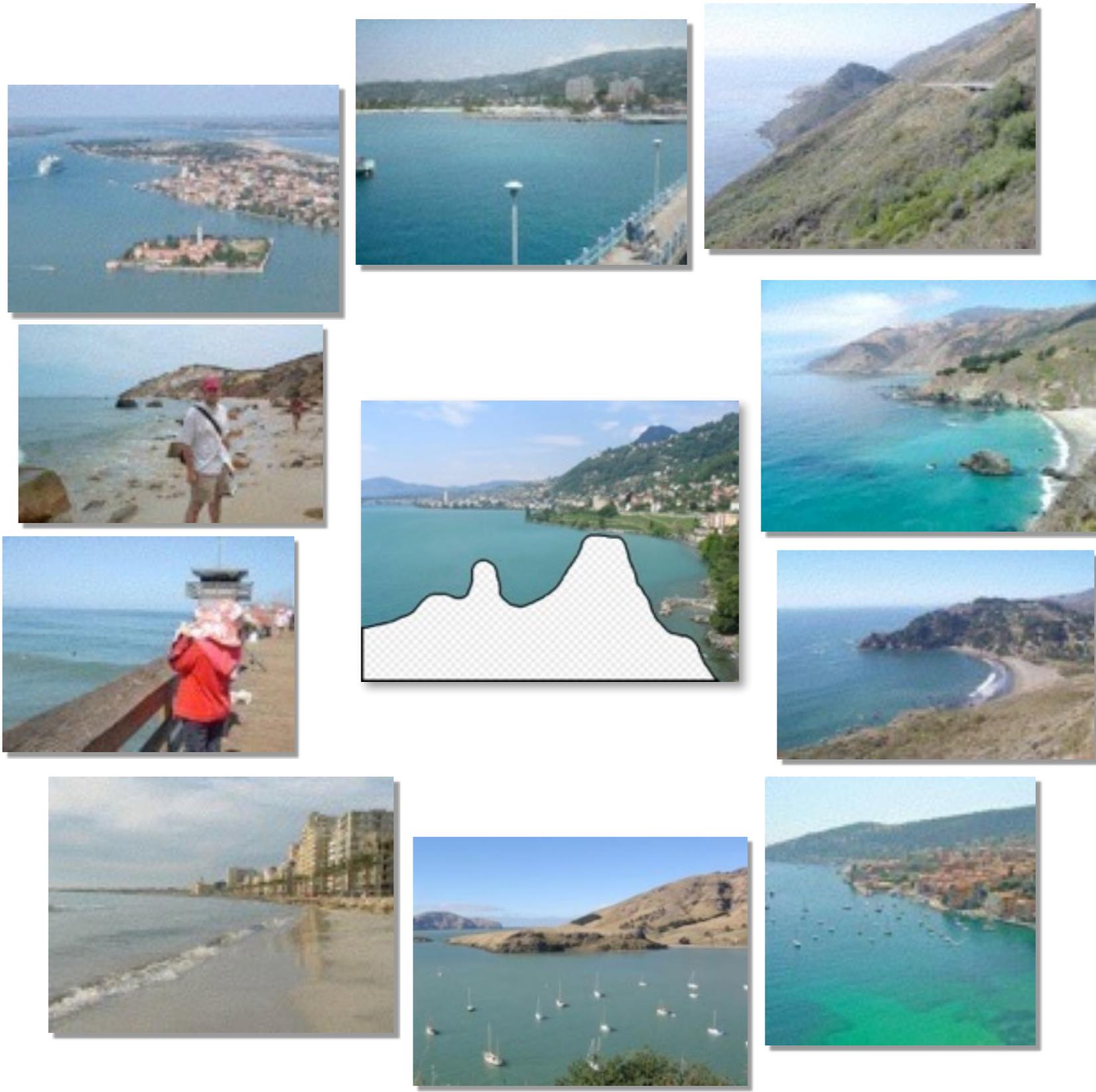
Truc: si vous avez assez d'images, la base de données devrait contenir des images suffisamment similaires, faciles à trouver!

Combien d'images?



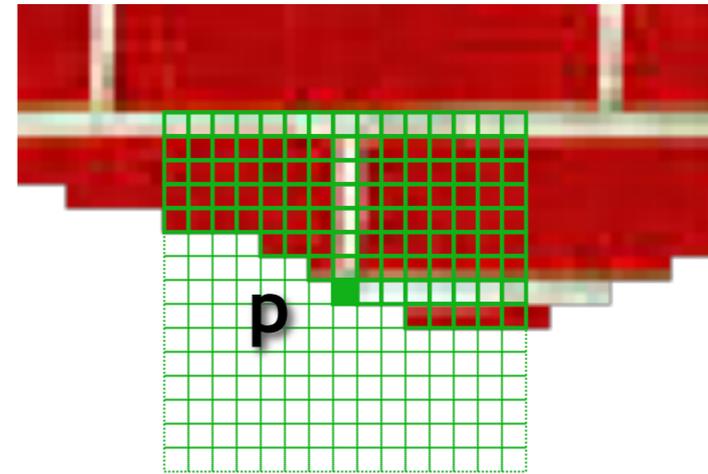
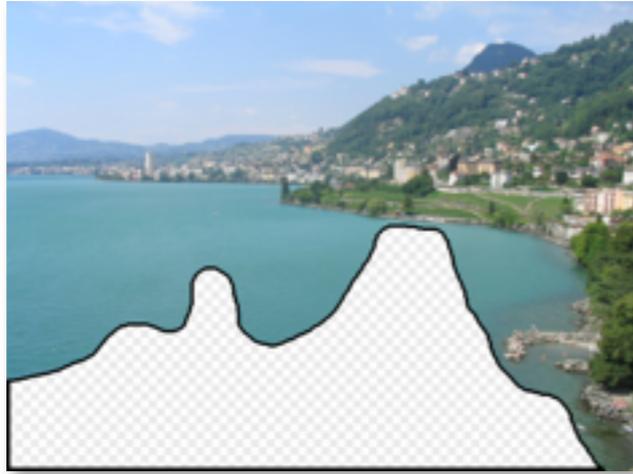


20,000 images

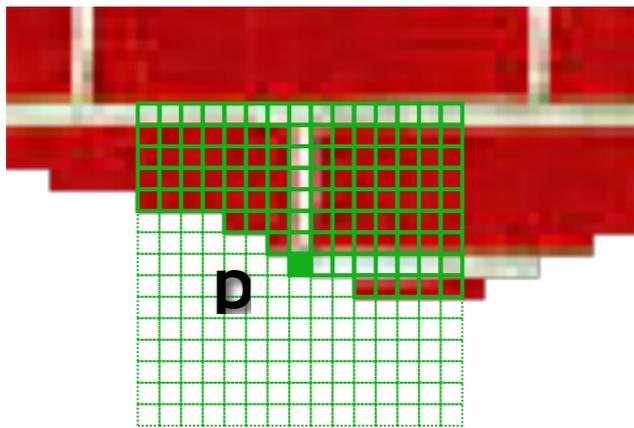
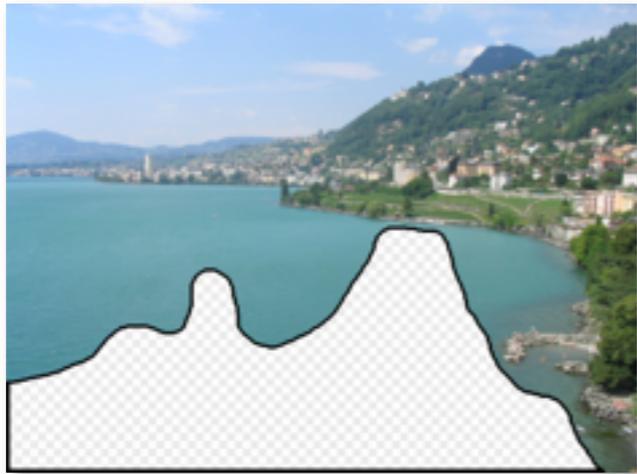


2,000,000 images

Limites



- Lent
- Fonction d'appariement définie manuellement (peut sembler arbitraire)
- Chaque méthode fonctionne seulement pour leur domaine particulier (e.g. scènes «typiques» à l'extérieur, textures semi-régulières, etc.)



Stocke toute l'information
dans une mémoire

«lookup table»



Assume que le monde
possède une structure simple

Apprend une représentation
qui capture cette structure



Classification units



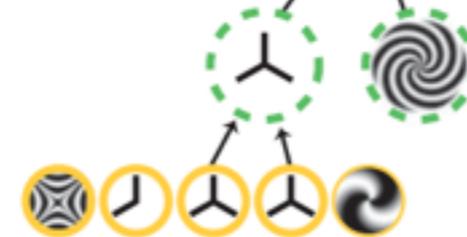
PIT/AIT



V4/PIT



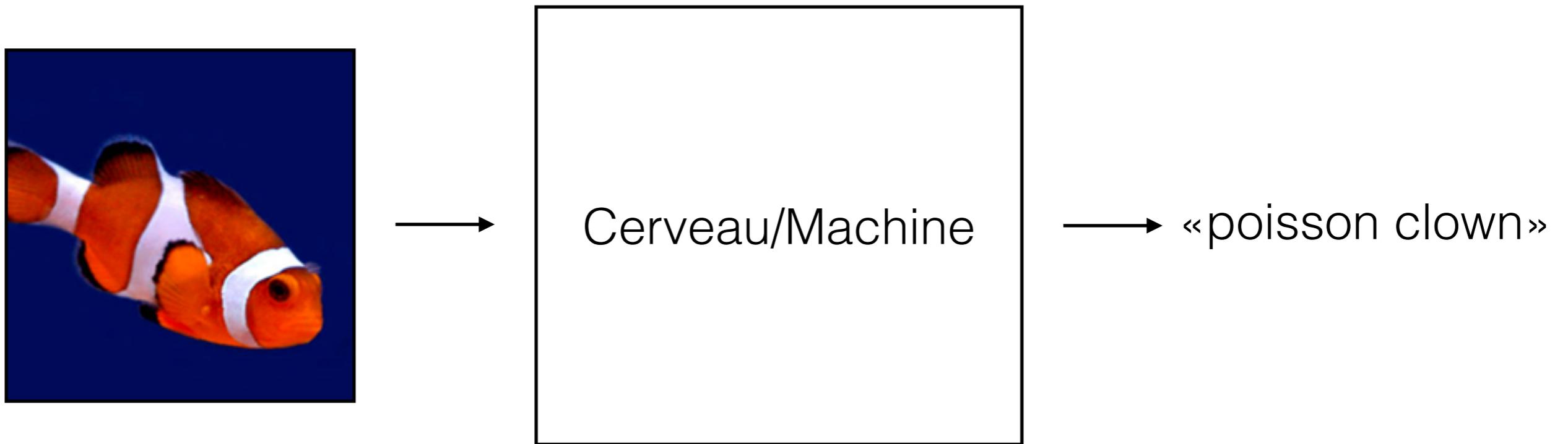
V2/V4



V1/V2

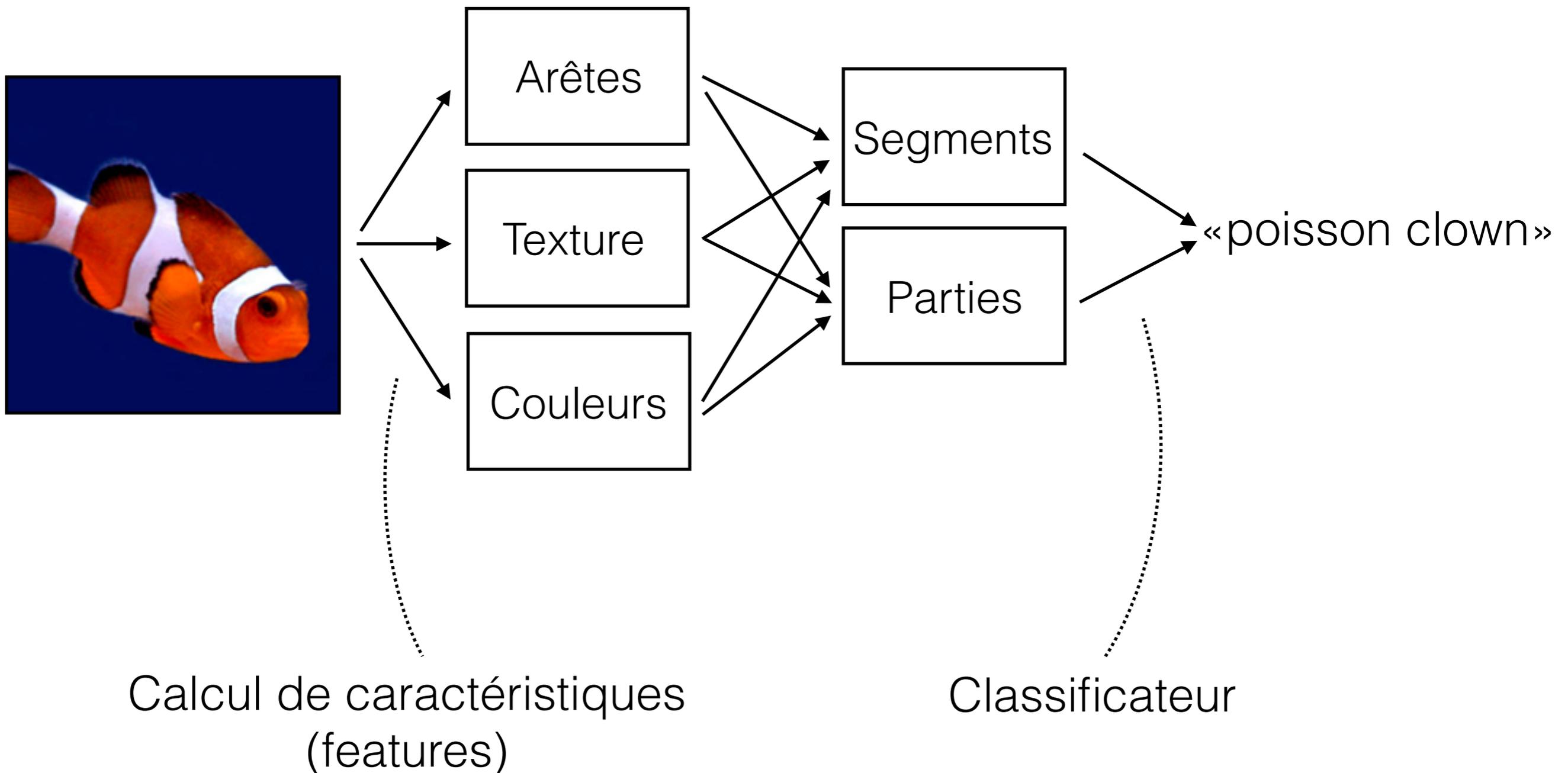


Idée de base



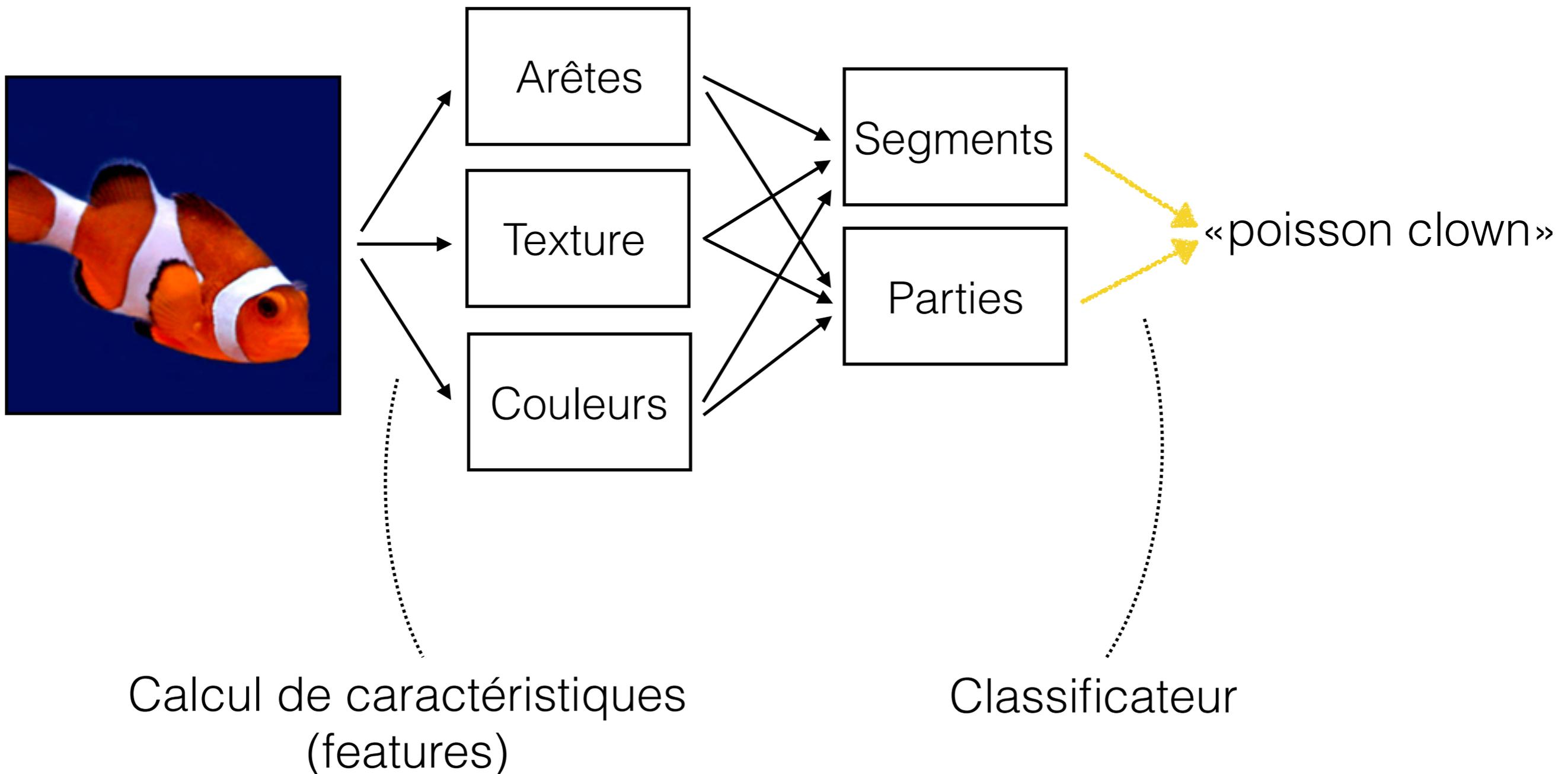
Hiérarchie d'unités *simples*

Reconnaissance d'objets: approche «traditionnelle»



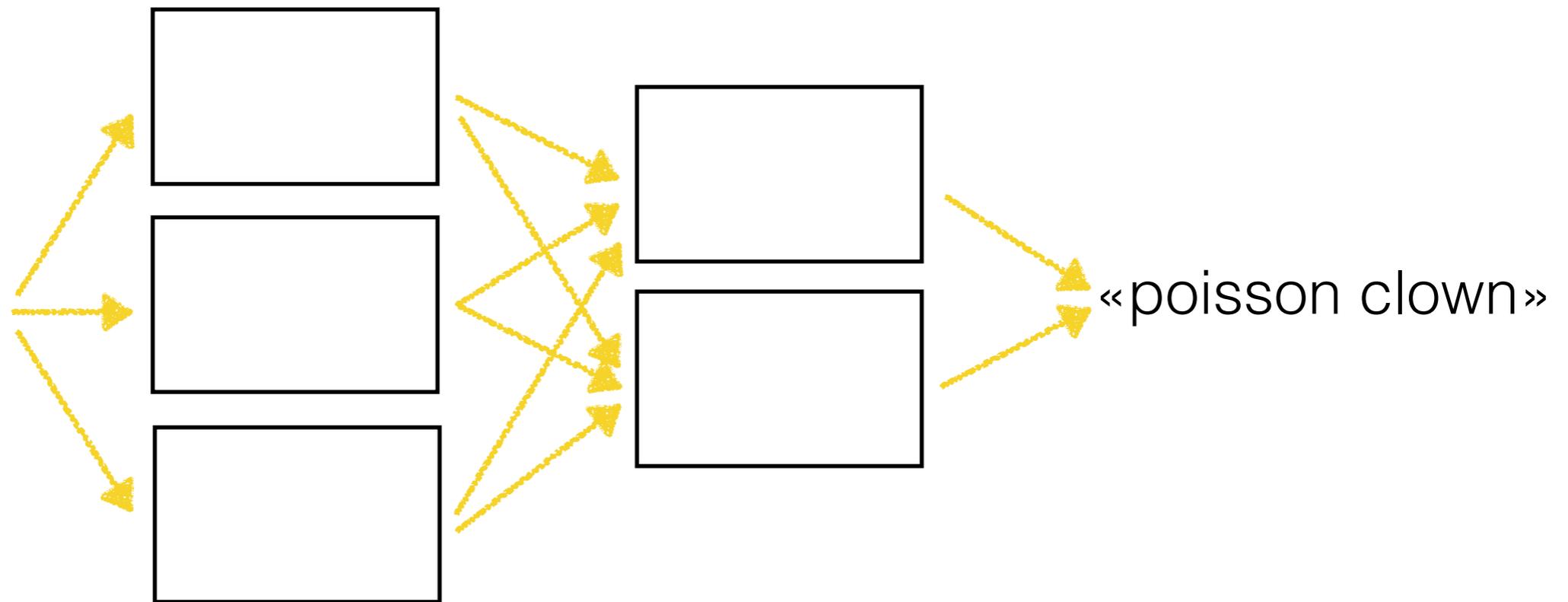
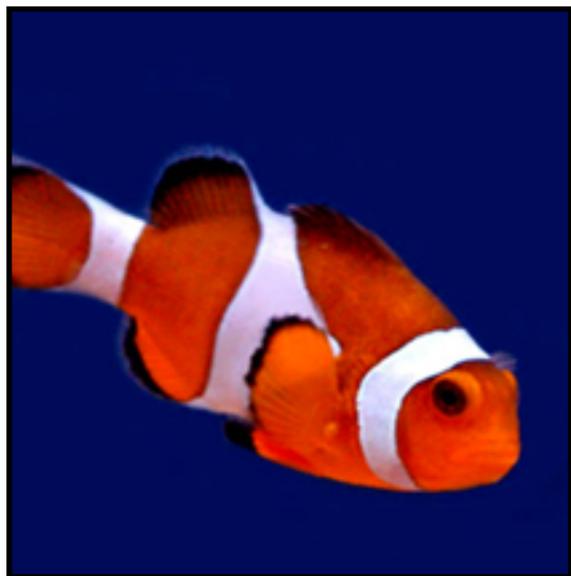
Reconnaissance d'objets: approche «traditionnelle»

Appris automatiquement



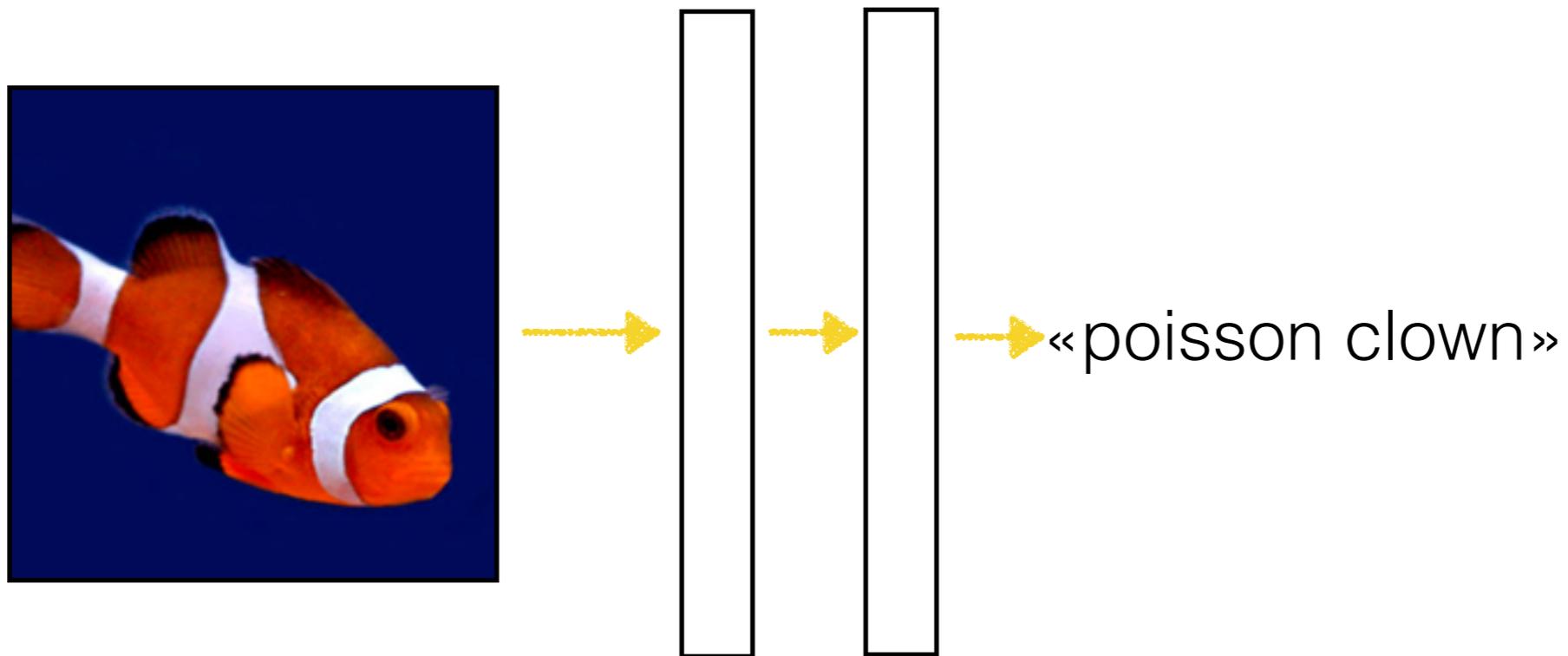
Réseau de neurones

Appris automatiquement



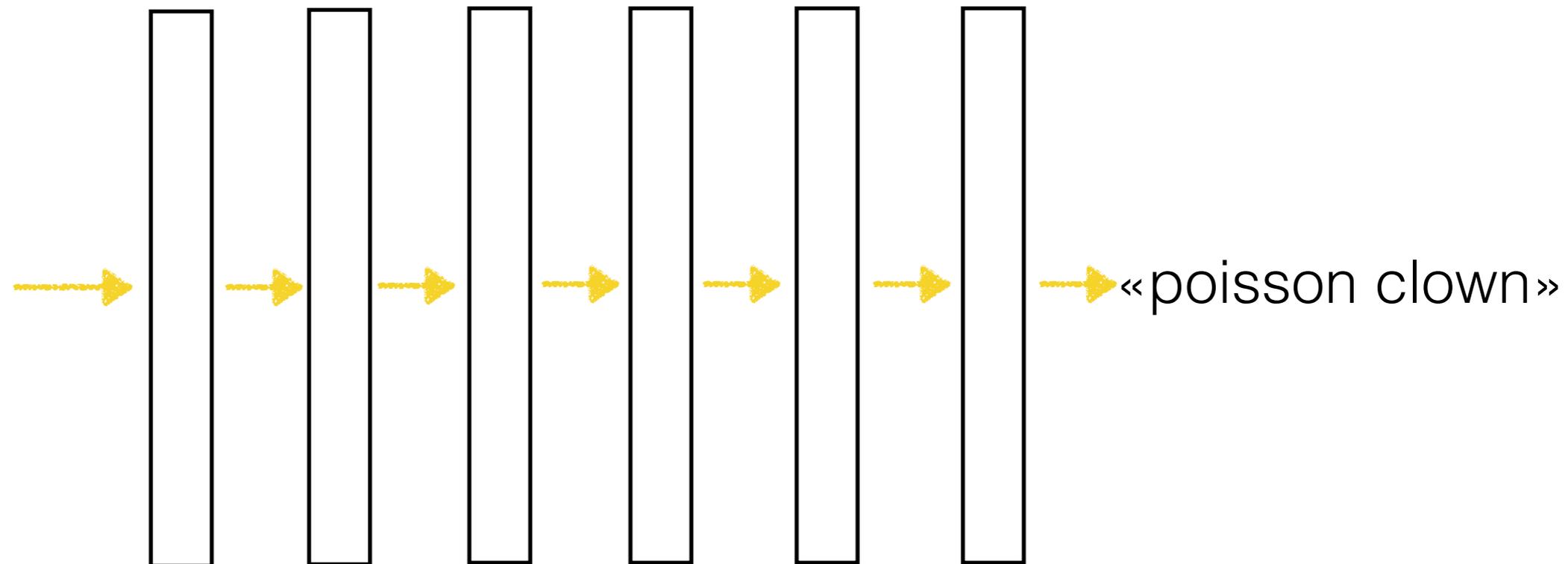
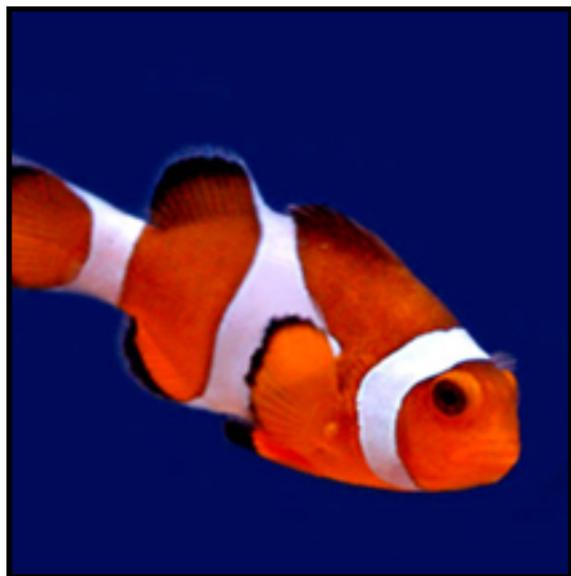
Réseau de neurones

Appris automatiquement



Réseau de neurones *profond*

Appris automatiquement

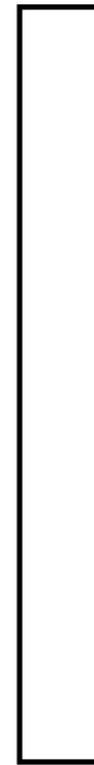


Calculs dans un réseau de neurones

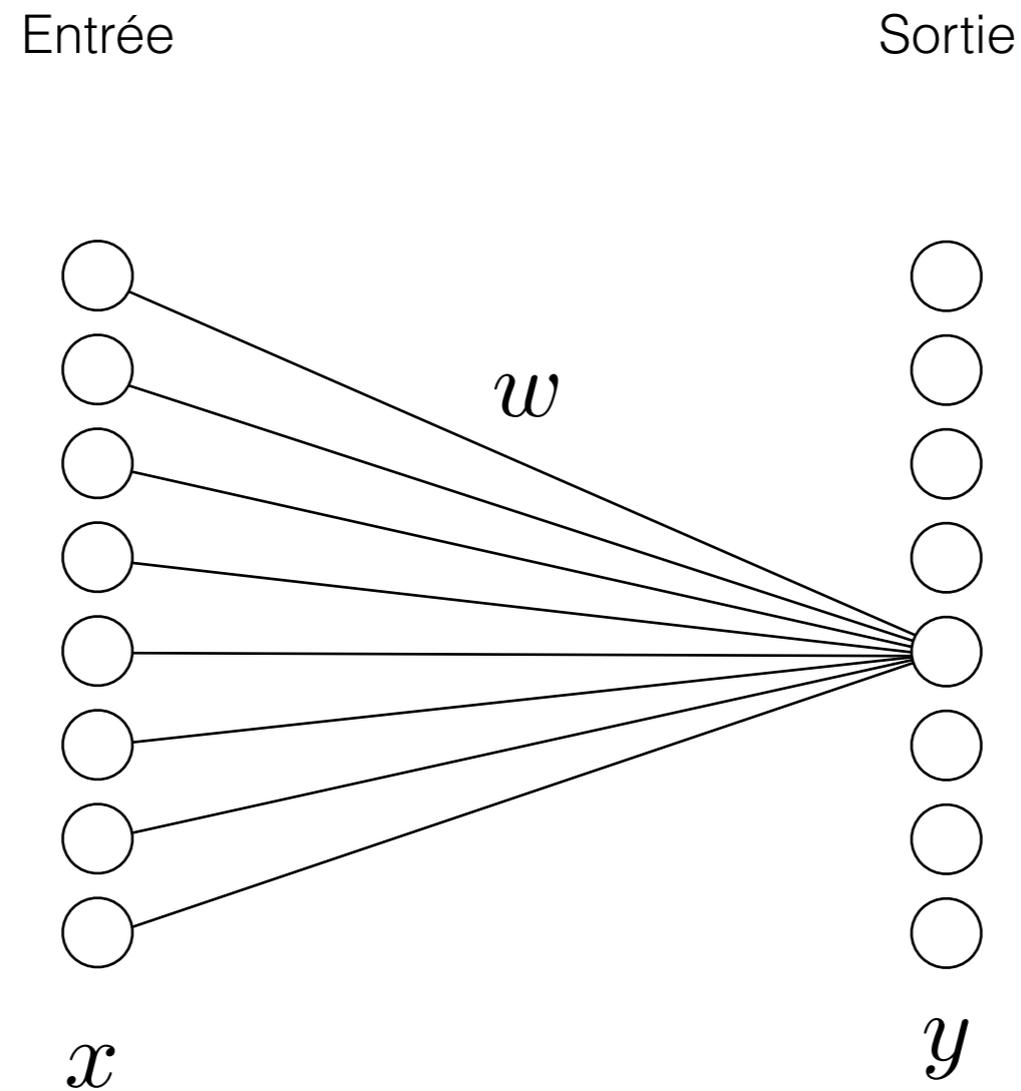
Que font ces flèches?
Quels calculs sont effectués?

Entrée

Sortie

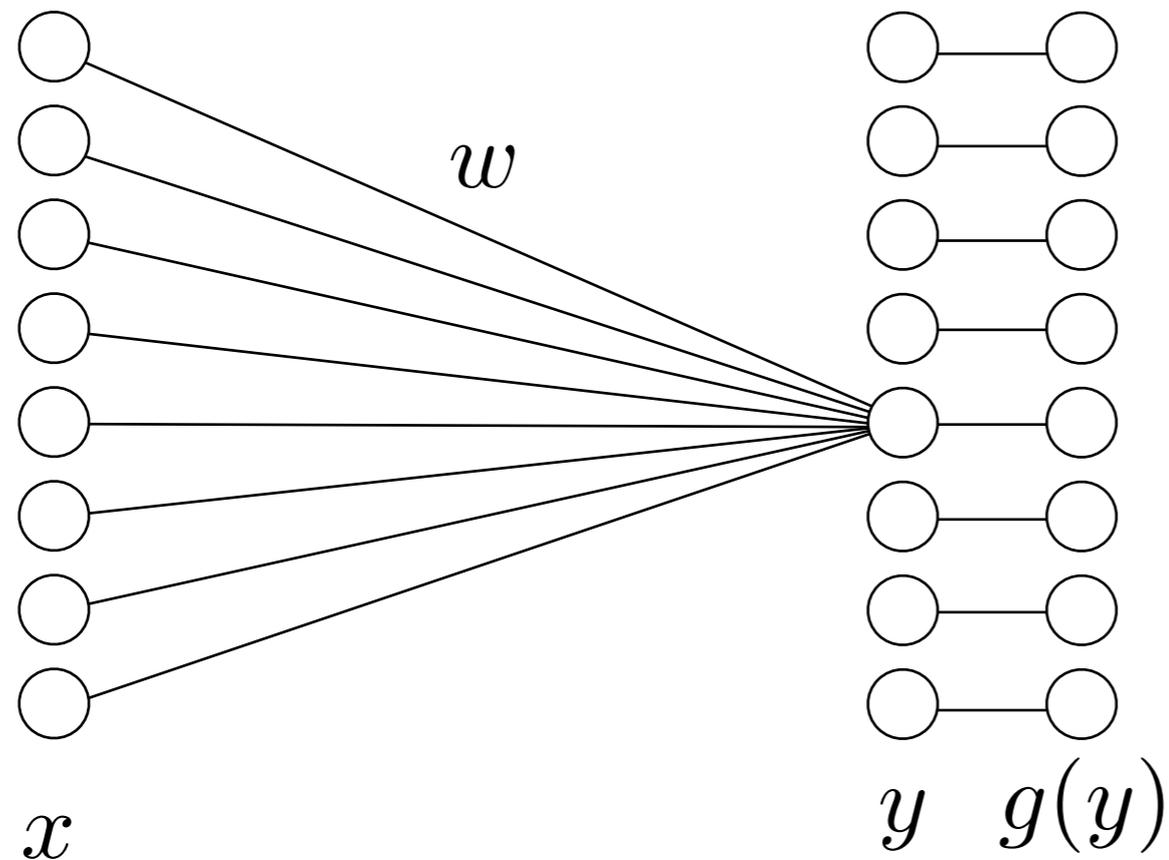


Calculs dans un réseau de neurones

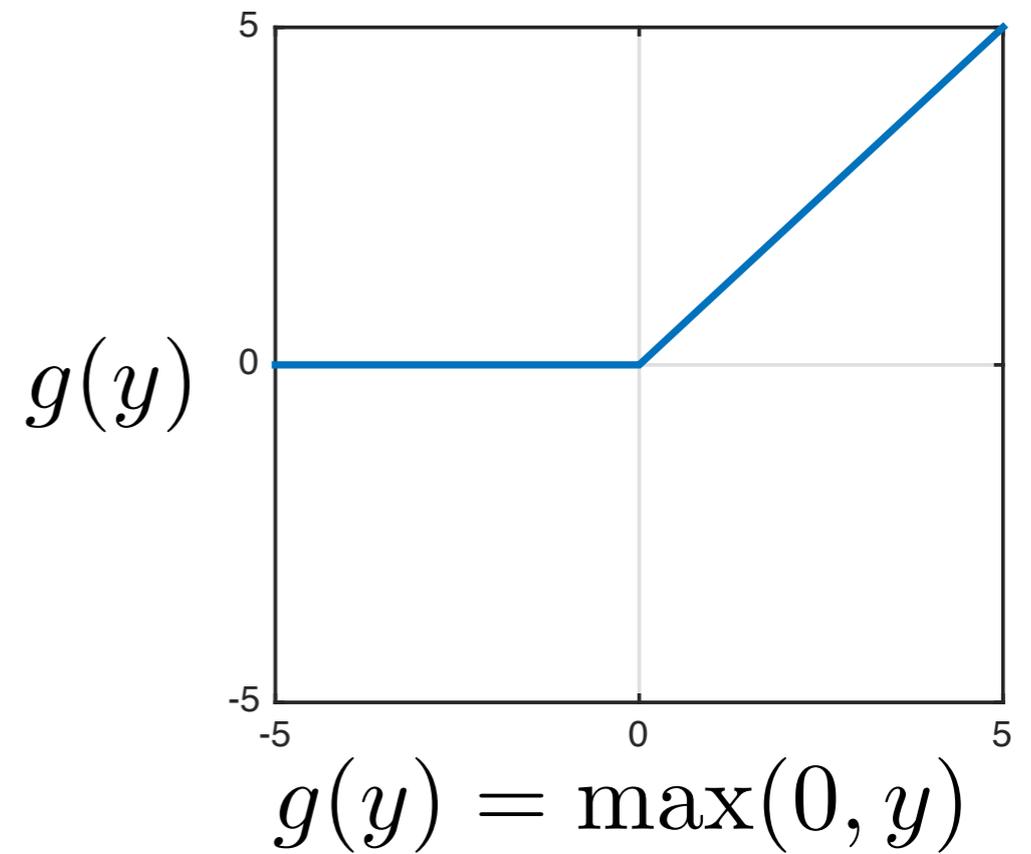


$$y_j = \sum_i w_{ij} x_i$$

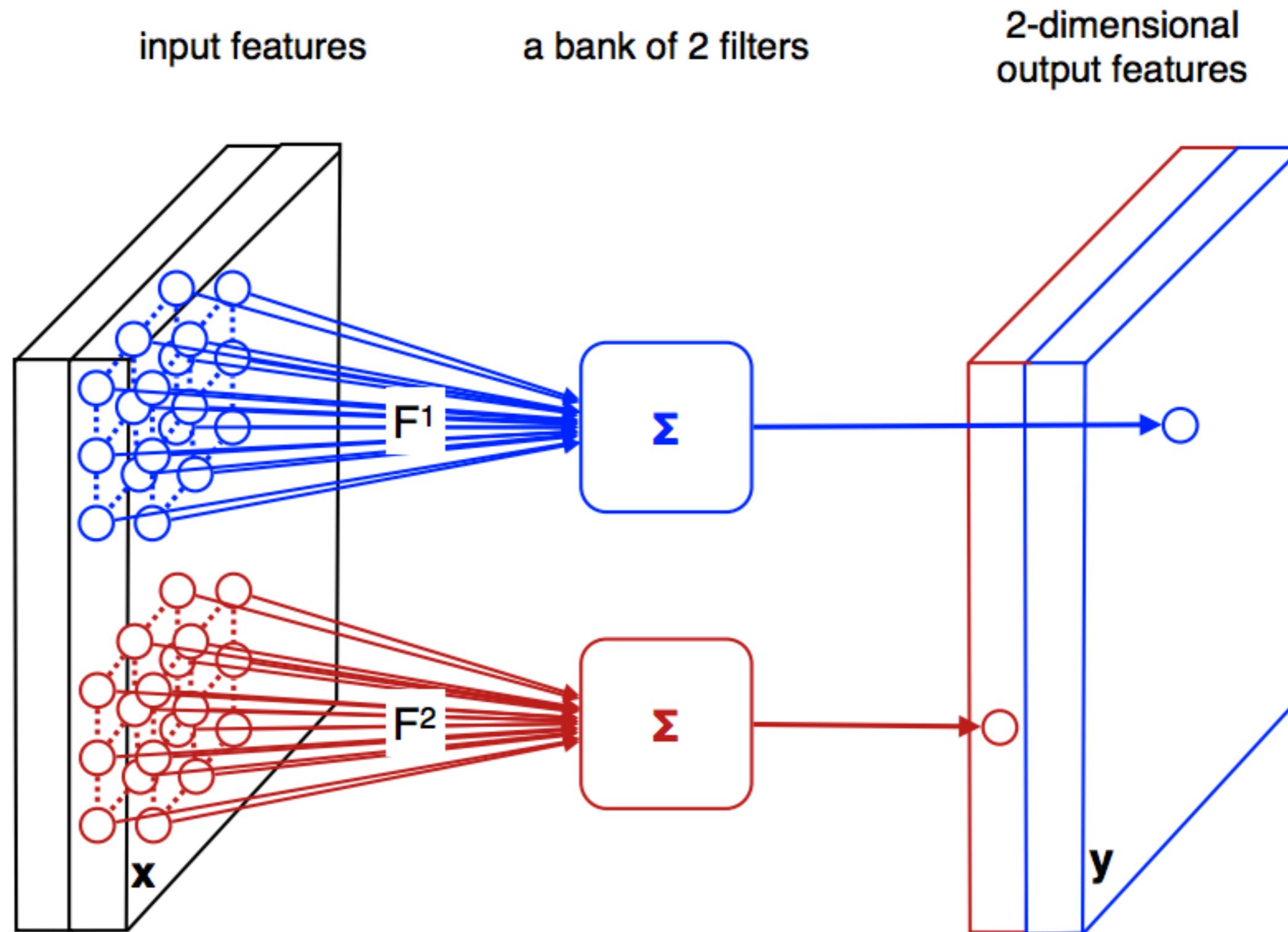
Calculs dans un réseau de neurones



«Rectified linear unit» (ReLU)



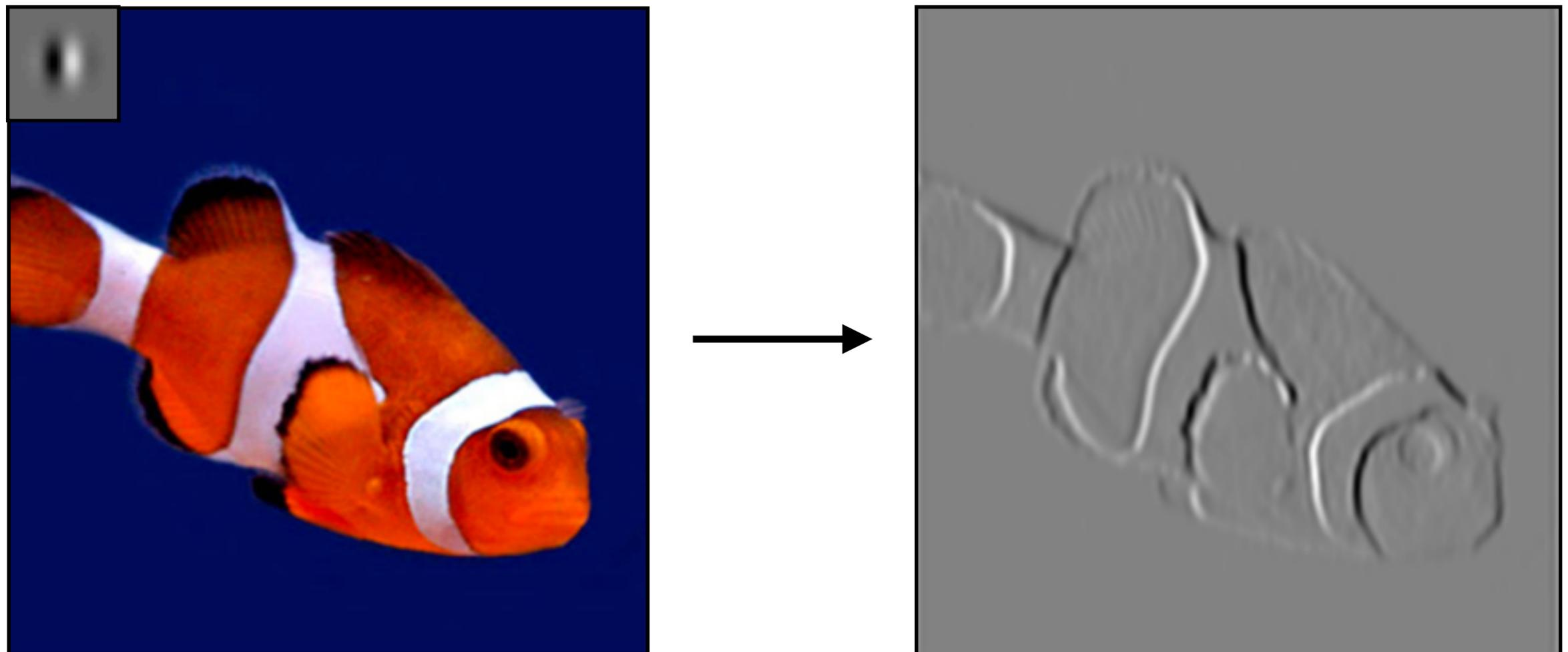
Réseaux de neurones à *convolution* (CNN)



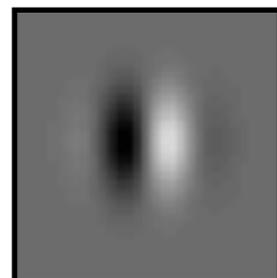
Réseaux de neurones à convolution (CNN)

Convolution

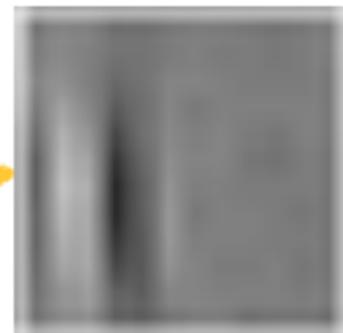
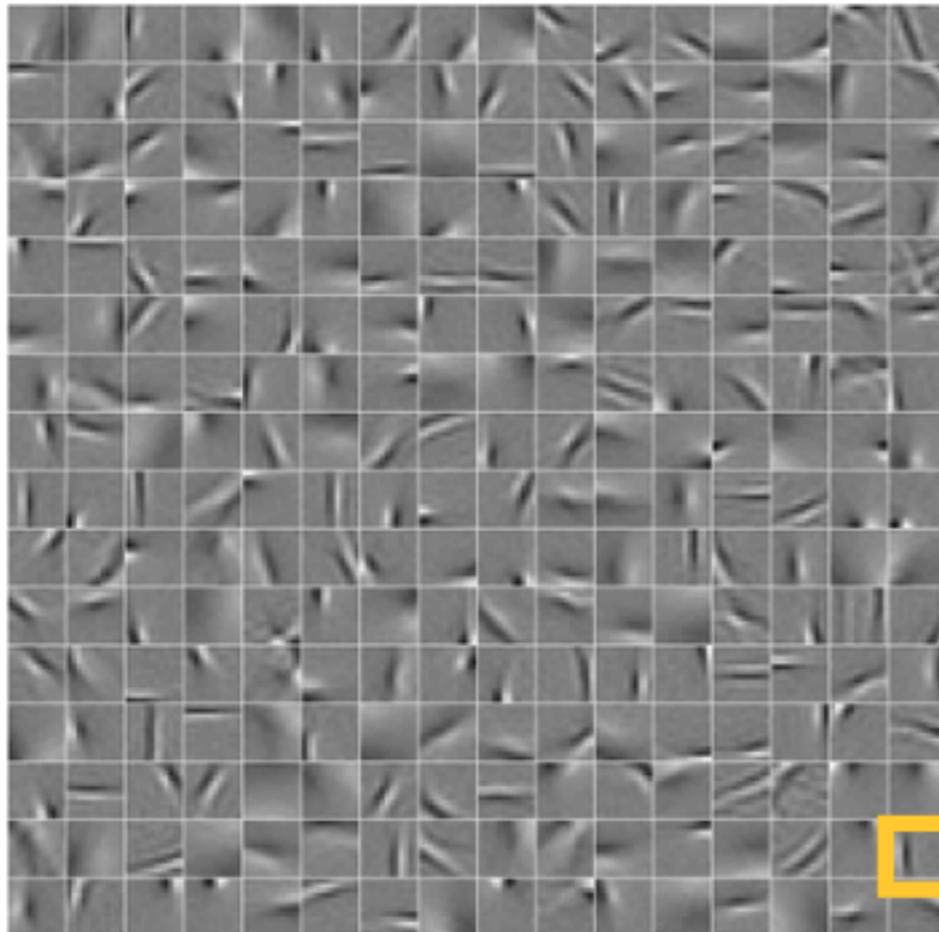
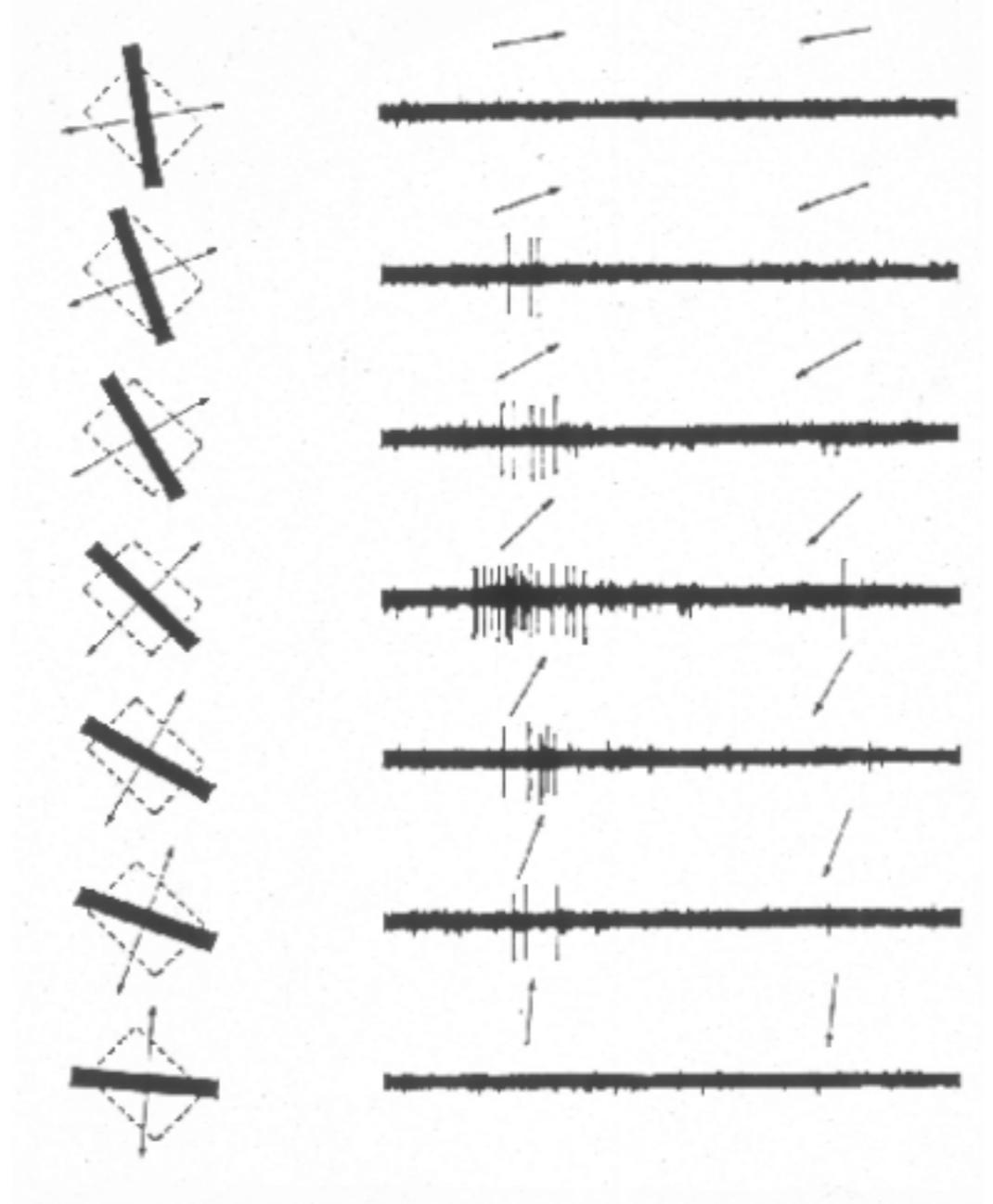
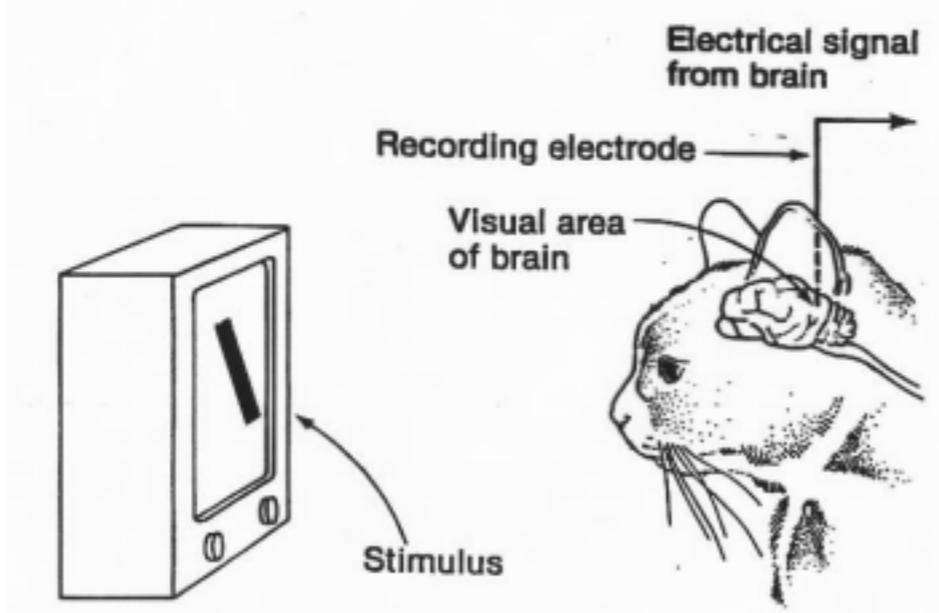
Avantage: les paramètres sont *partagés*
(toutes les fenêtres de l'image sont traitées de la même façon)



filtre

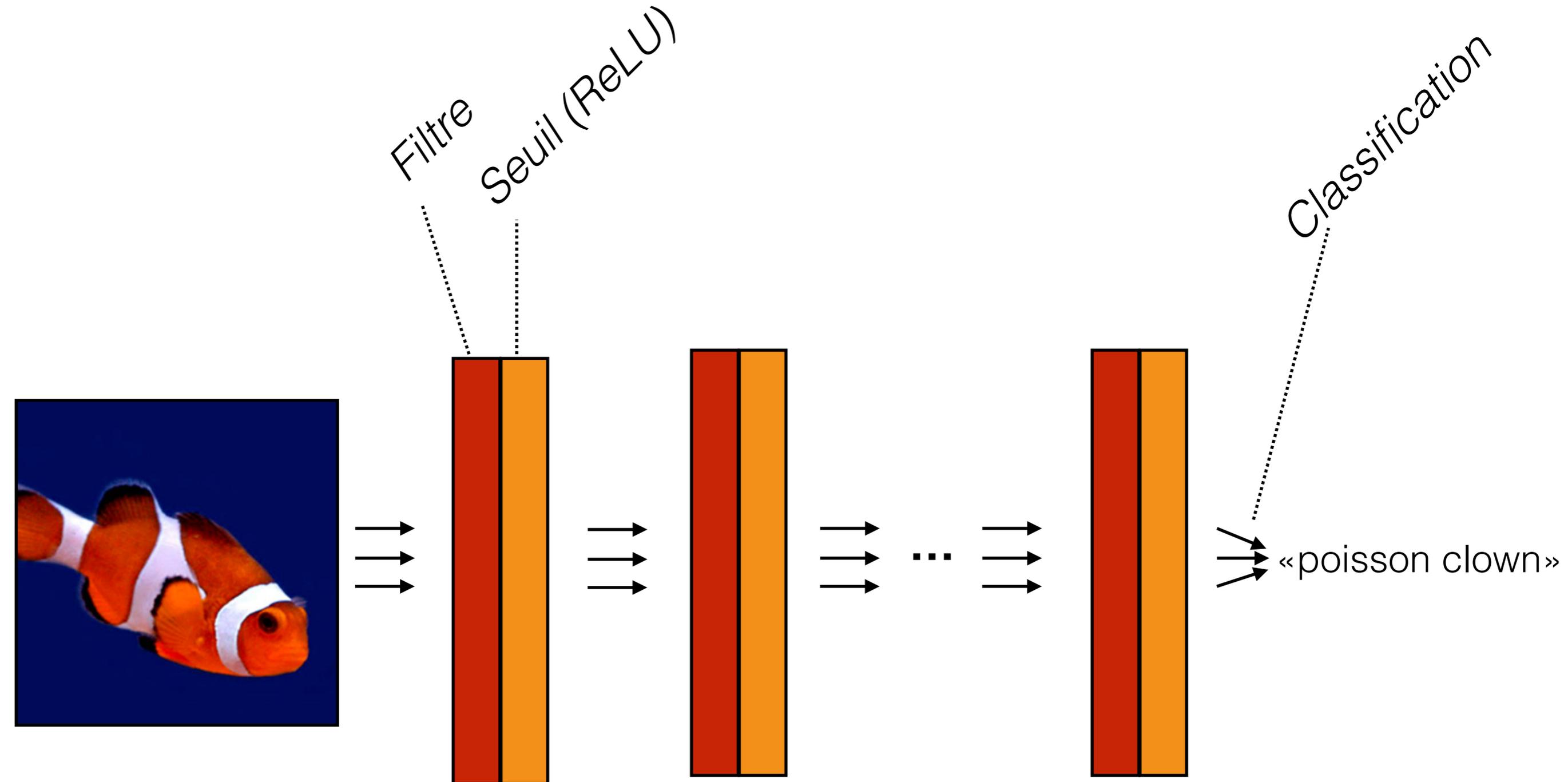


[Hubel and Wiesel 59]



oriented filter

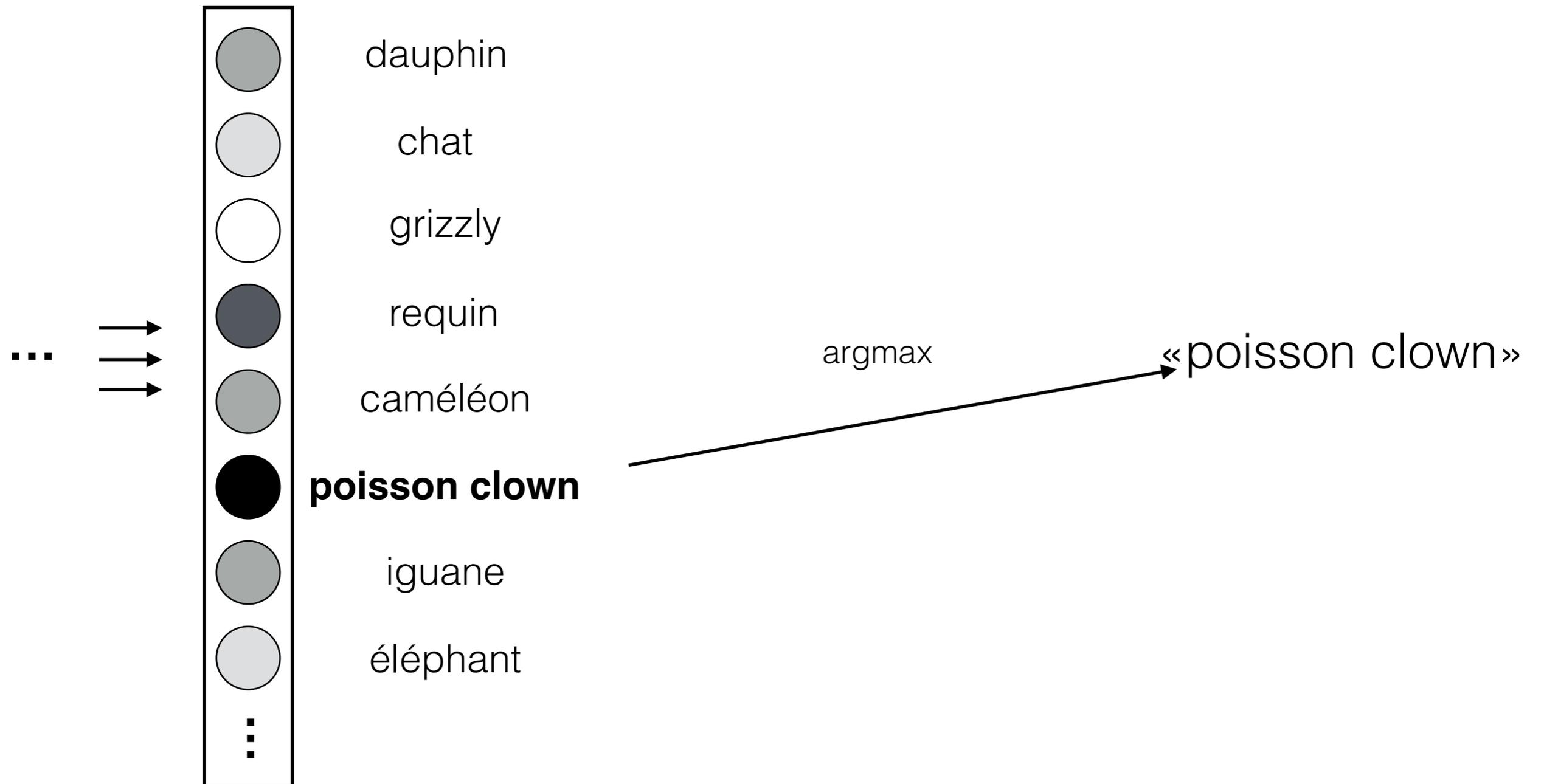
Calculs dans un réseau de neurones



$$f(\mathbf{x}) = f_L(\dots f_2(f_1(\mathbf{x})))$$

Calculs dans un réseau de neurones

Dernière couche
(classification)





Classification units



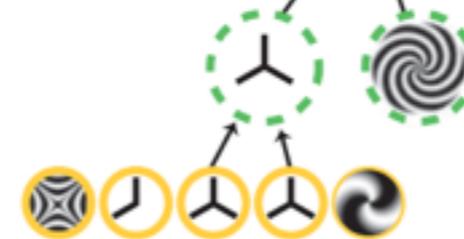
PIT/AIT



V4/PIT



V2/V4

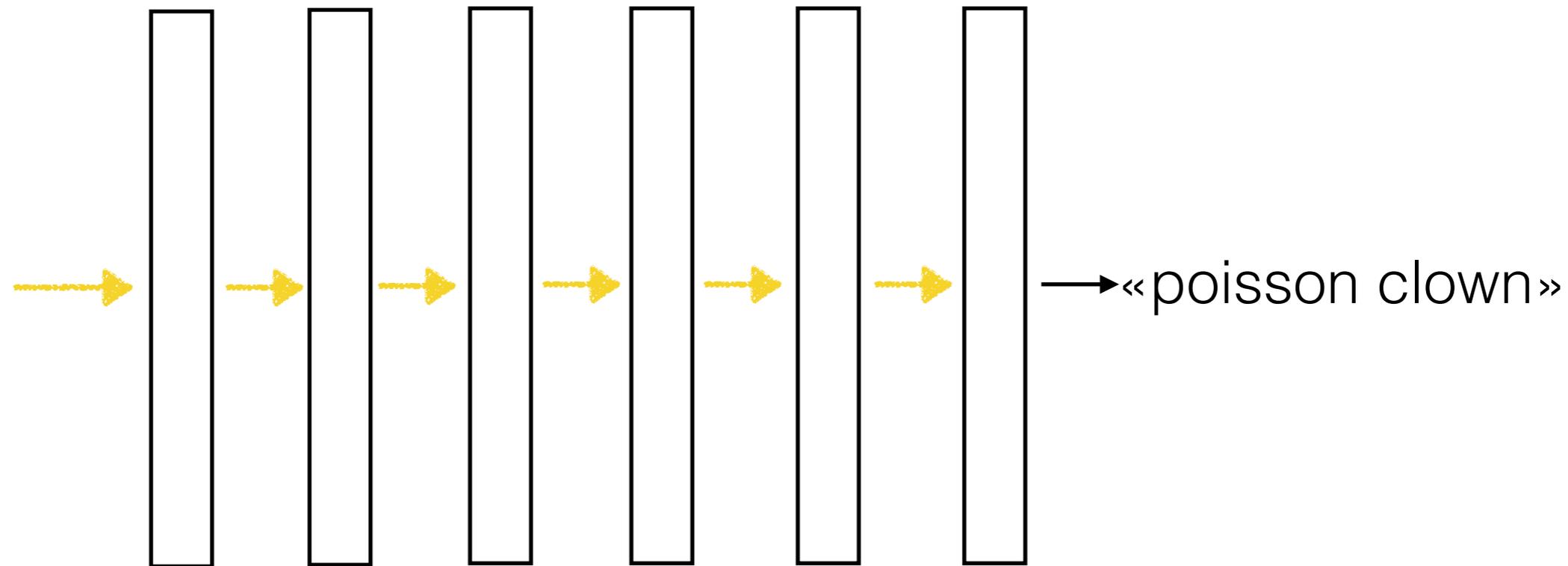
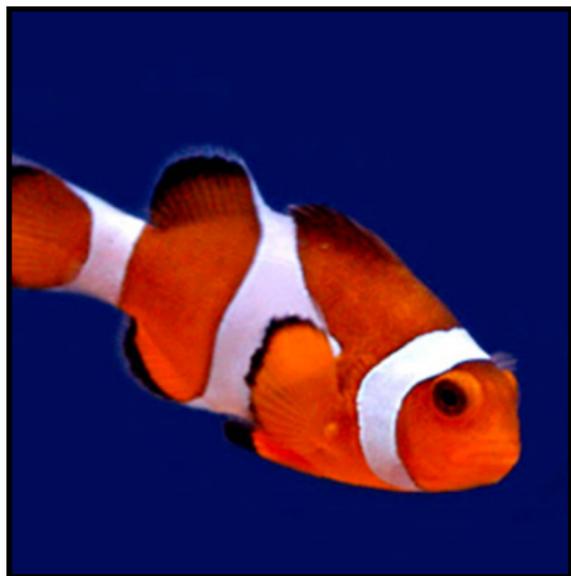


V1/V2



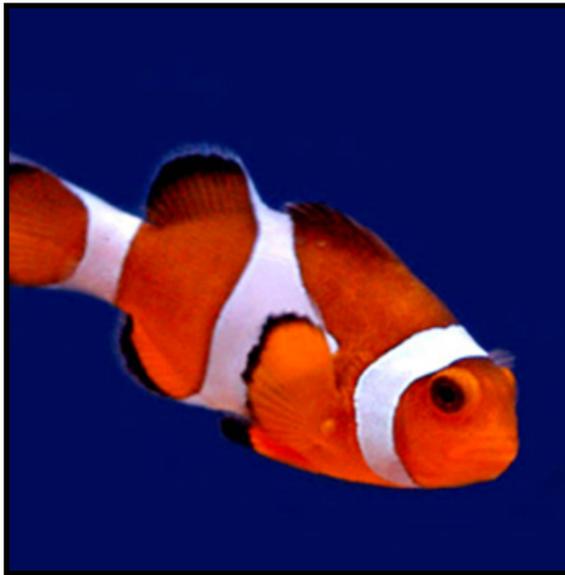
Apprentissage par réseaux profonds

Appris automatiquement



Idée: on modifie les poids (dans un CNN, les filtres!) jusqu'à temps que la sortie corresponde à ce qu'on veut

Apprentissage par réseaux profonds



→ «poisson clown»



→ «grizzly»



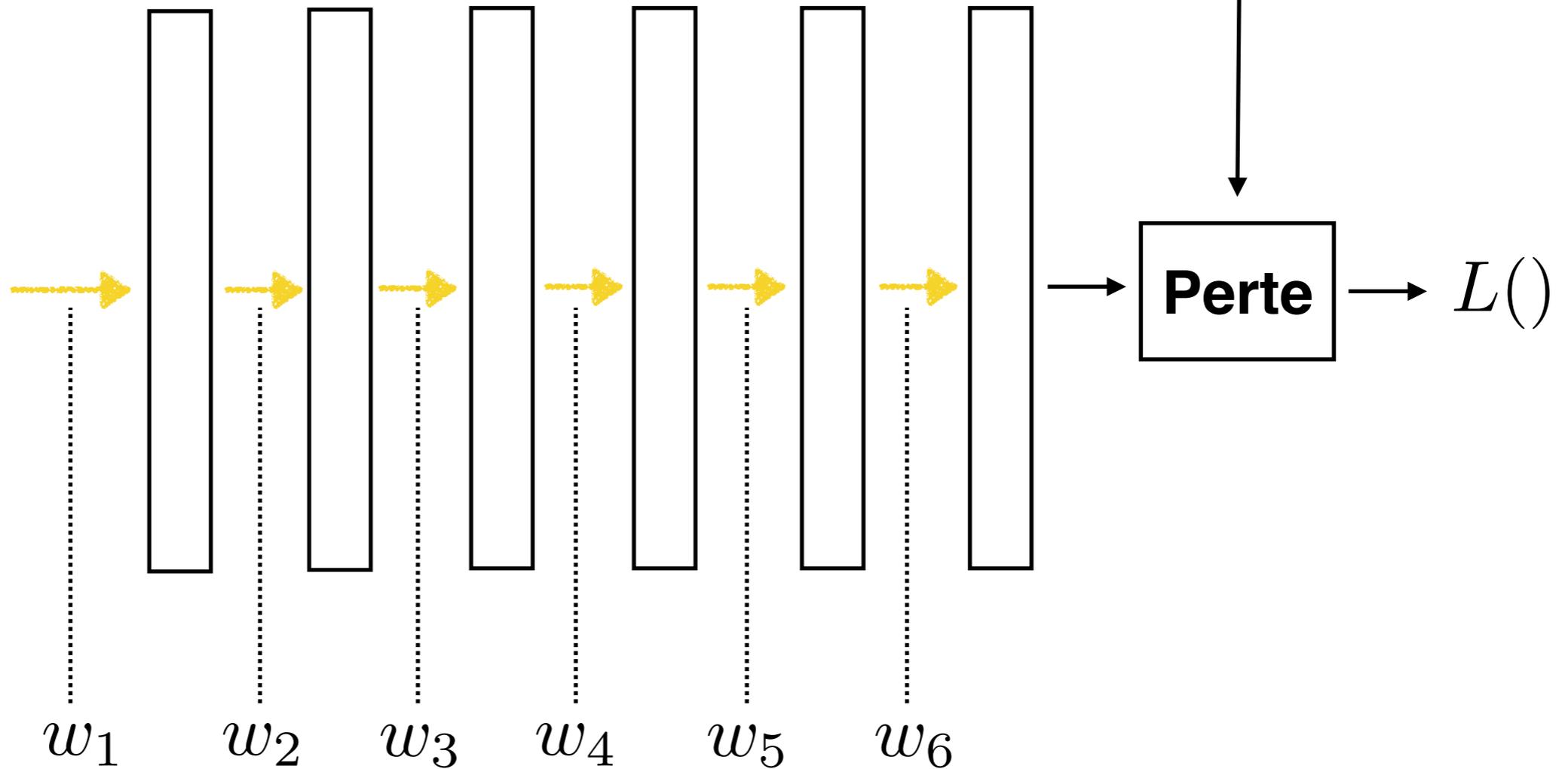
→ «caméléon»

Nous voudrions entraîner le réseau à associer chaque image à la bonne étiquette

Apprentissage par réseaux profonds

Appris automatiquement

«poisson clown»

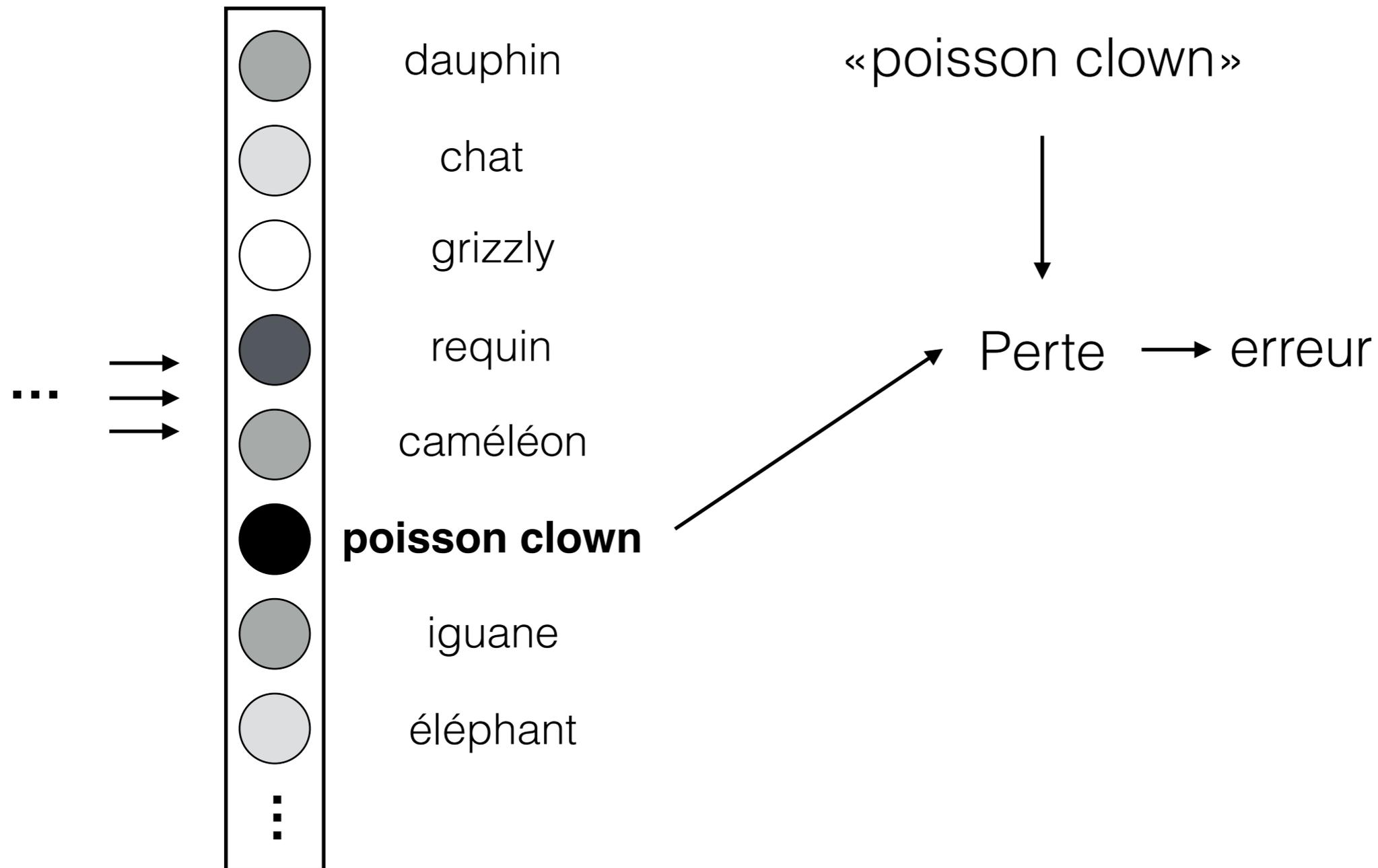


$$\underset{\mathbf{w}}{\operatorname{argmin}} L(w_1, \dots, w_6)$$

Fonction de perte (*loss function*)

Sortie du réseau

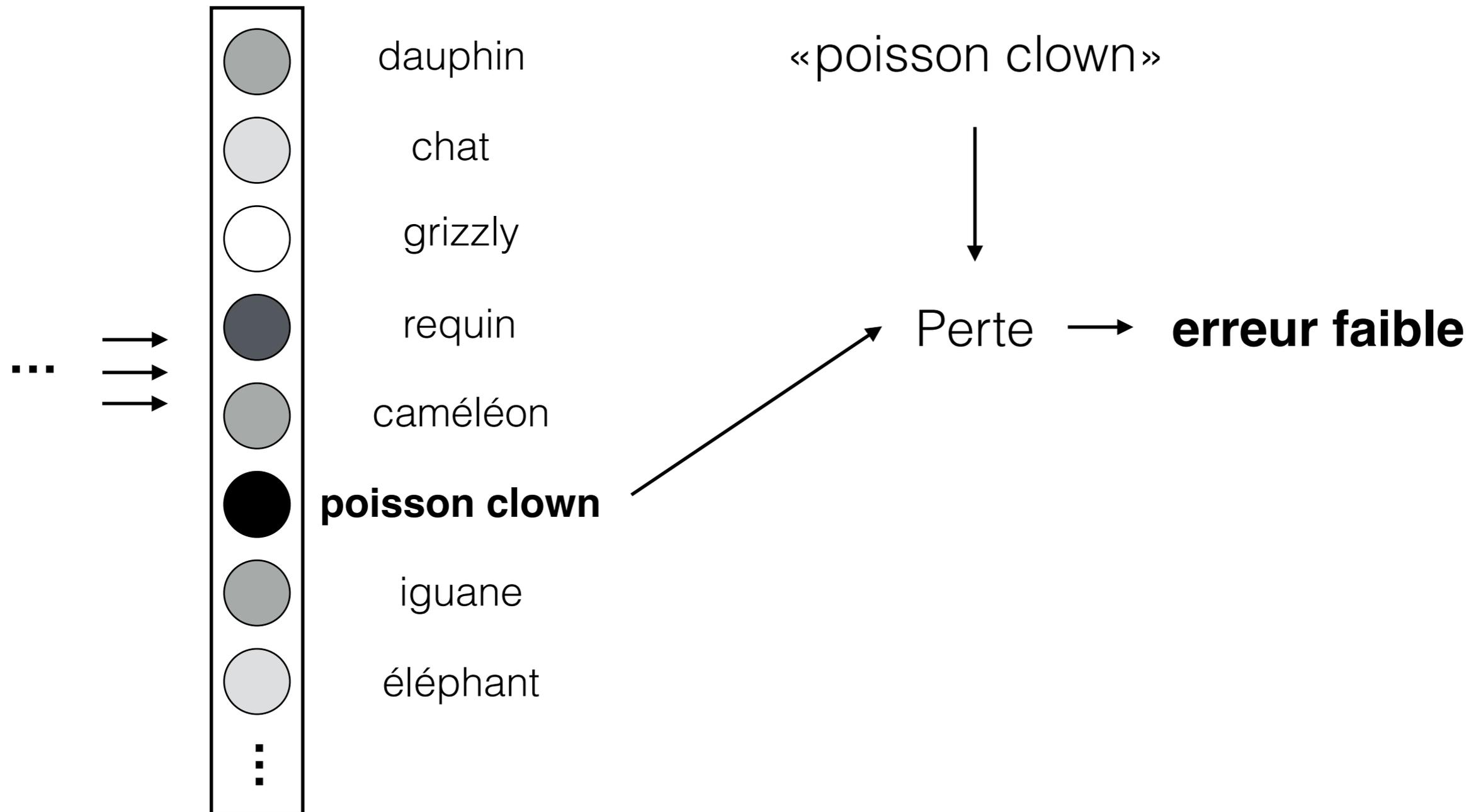
«Vraie» étiquette



Fonction de perte (*loss function*)

Sortie du réseau

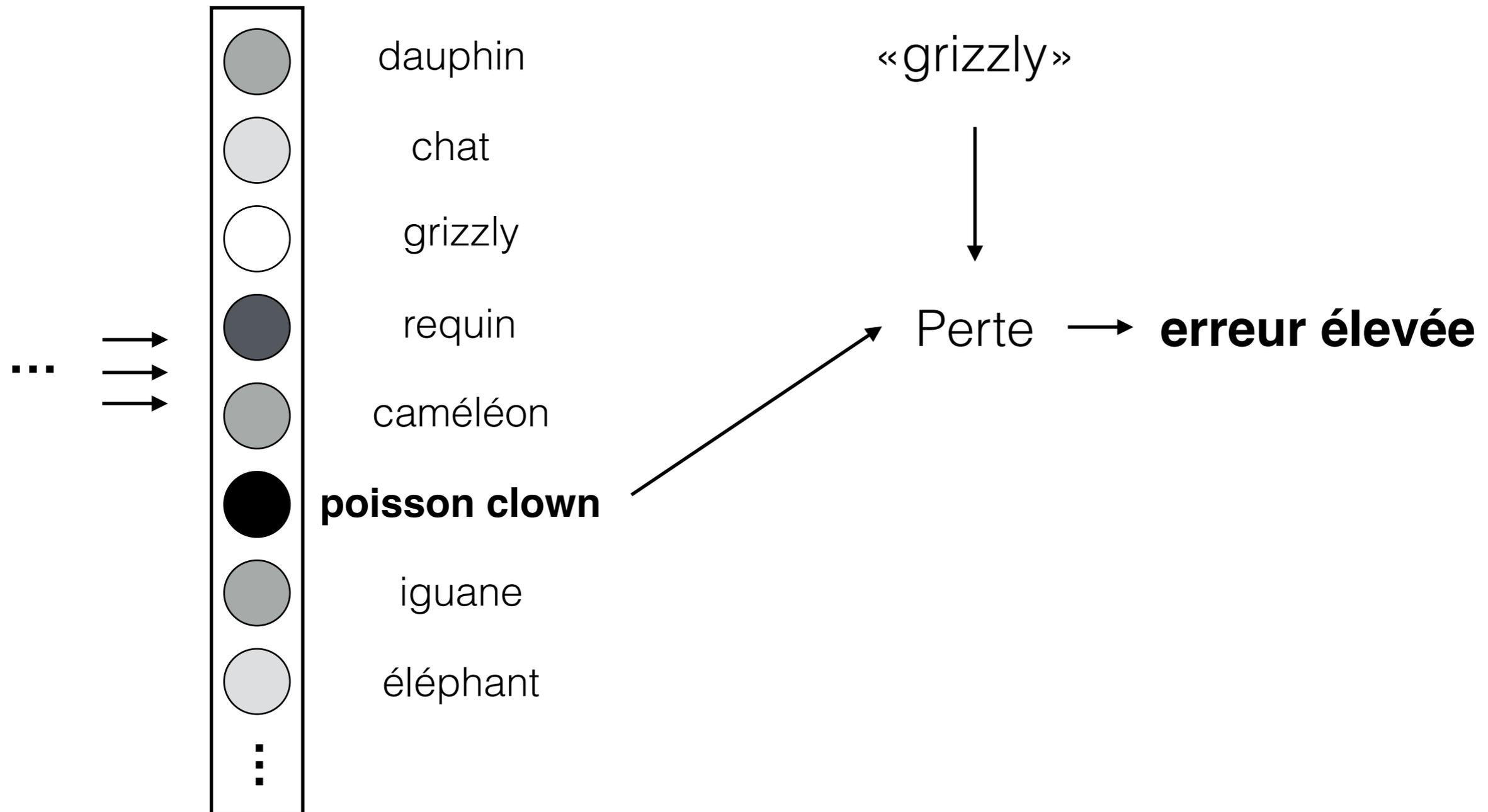
«Vraie» étiquette



Fonction de perte (*loss function*)

Sortie du réseau

«Vraie» étiquette



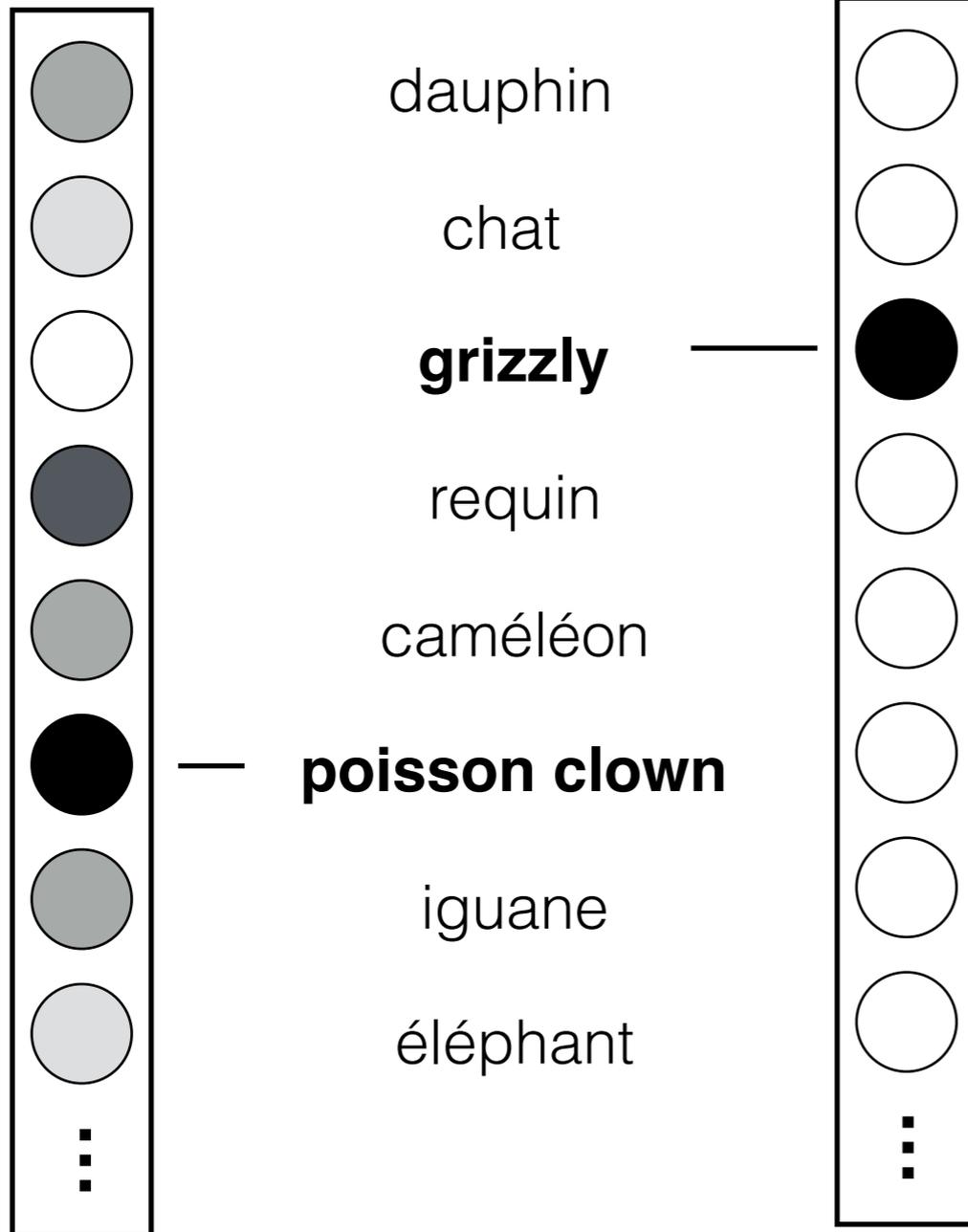
Fonction de perte pour classification

Sortie du réseau

«Vraie» étiquette

$\hat{\mathbf{z}}$

\mathbf{z}



Probabilité des données observées sous le modèle

$$H(\hat{\mathbf{z}}, \mathbf{z}) = - \sum_c \hat{\mathbf{z}}_c \log \mathbf{z}_c$$

*Apprend un modèle
de type $p(c|\mathbf{x})$*

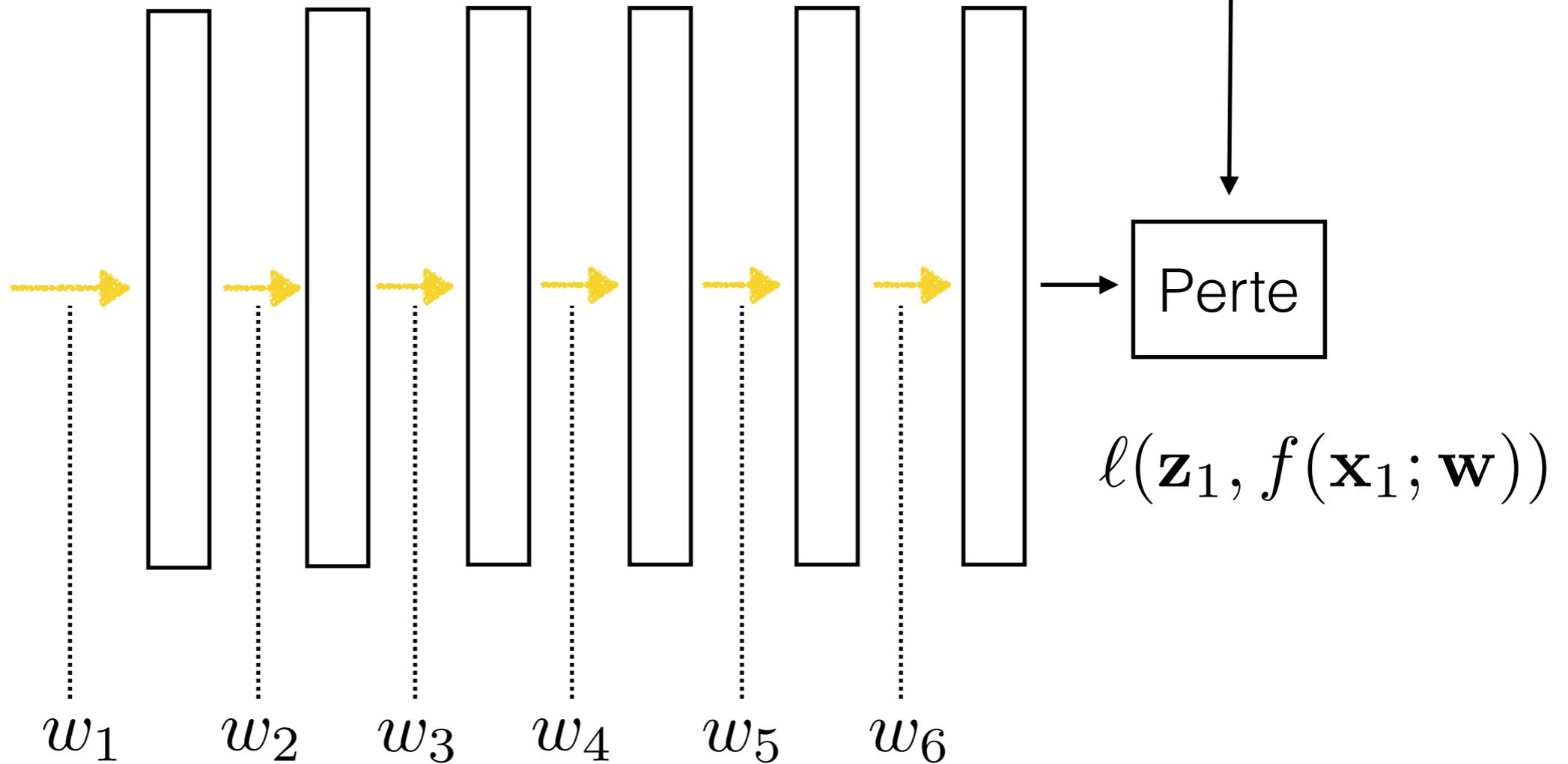
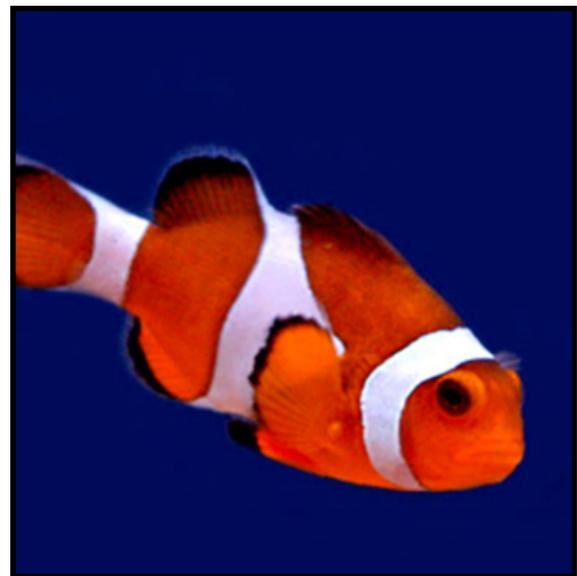
Apprentissage par réseaux profonds

Appris automatiquement

\mathbf{z}_1

«poisson clown»

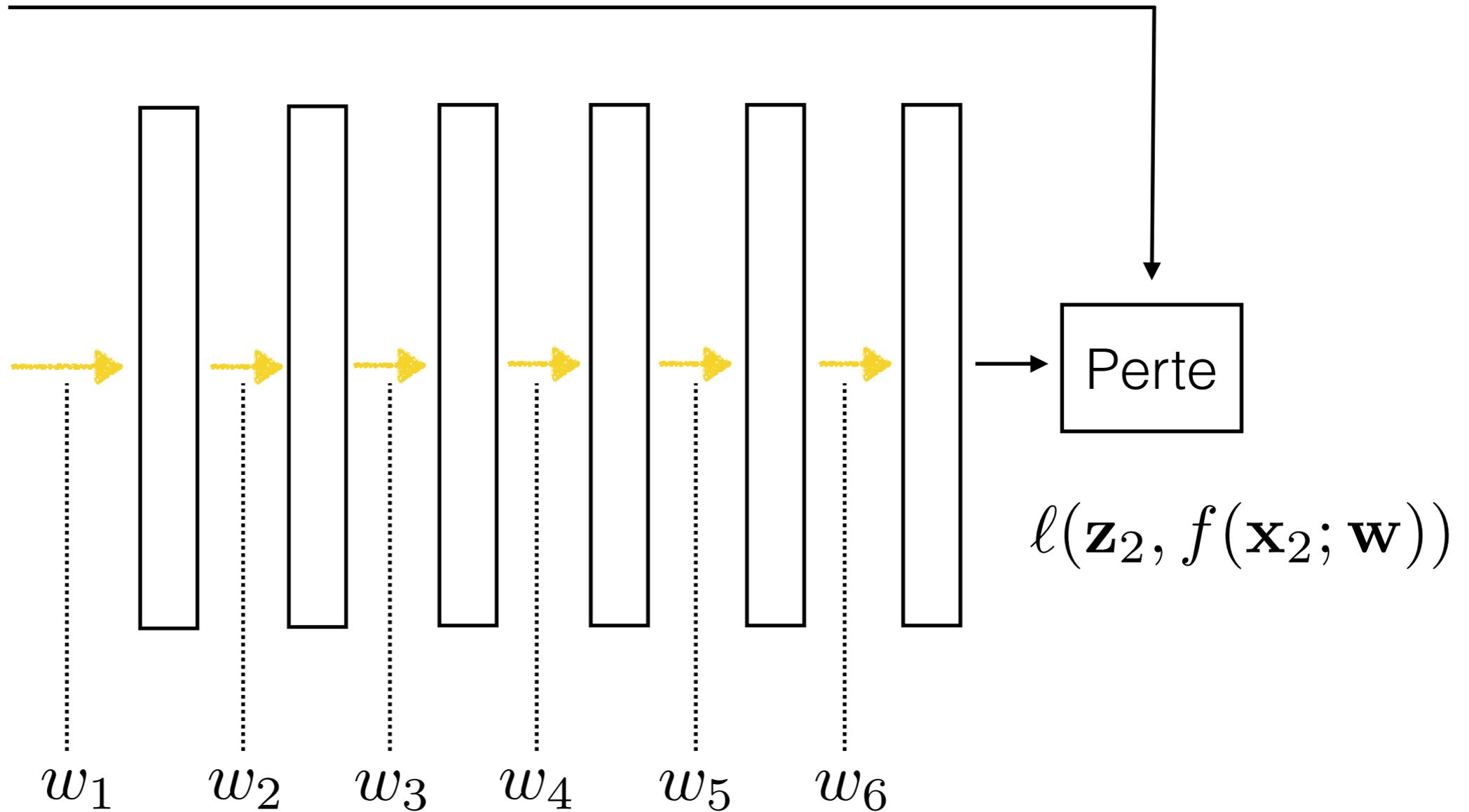
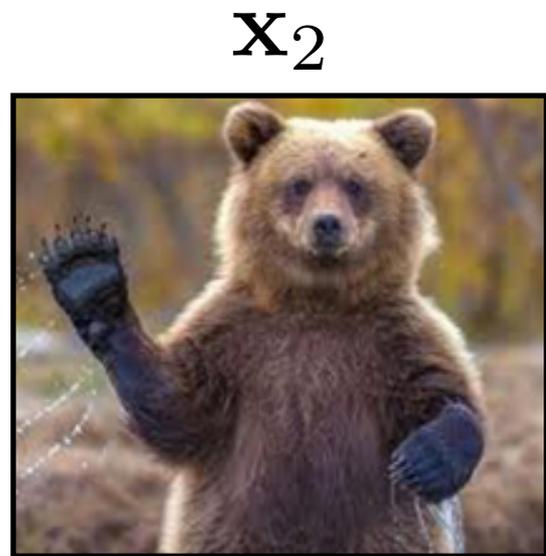
\mathbf{x}_1



Apprentissage par réseaux profonds

Appris automatiquement

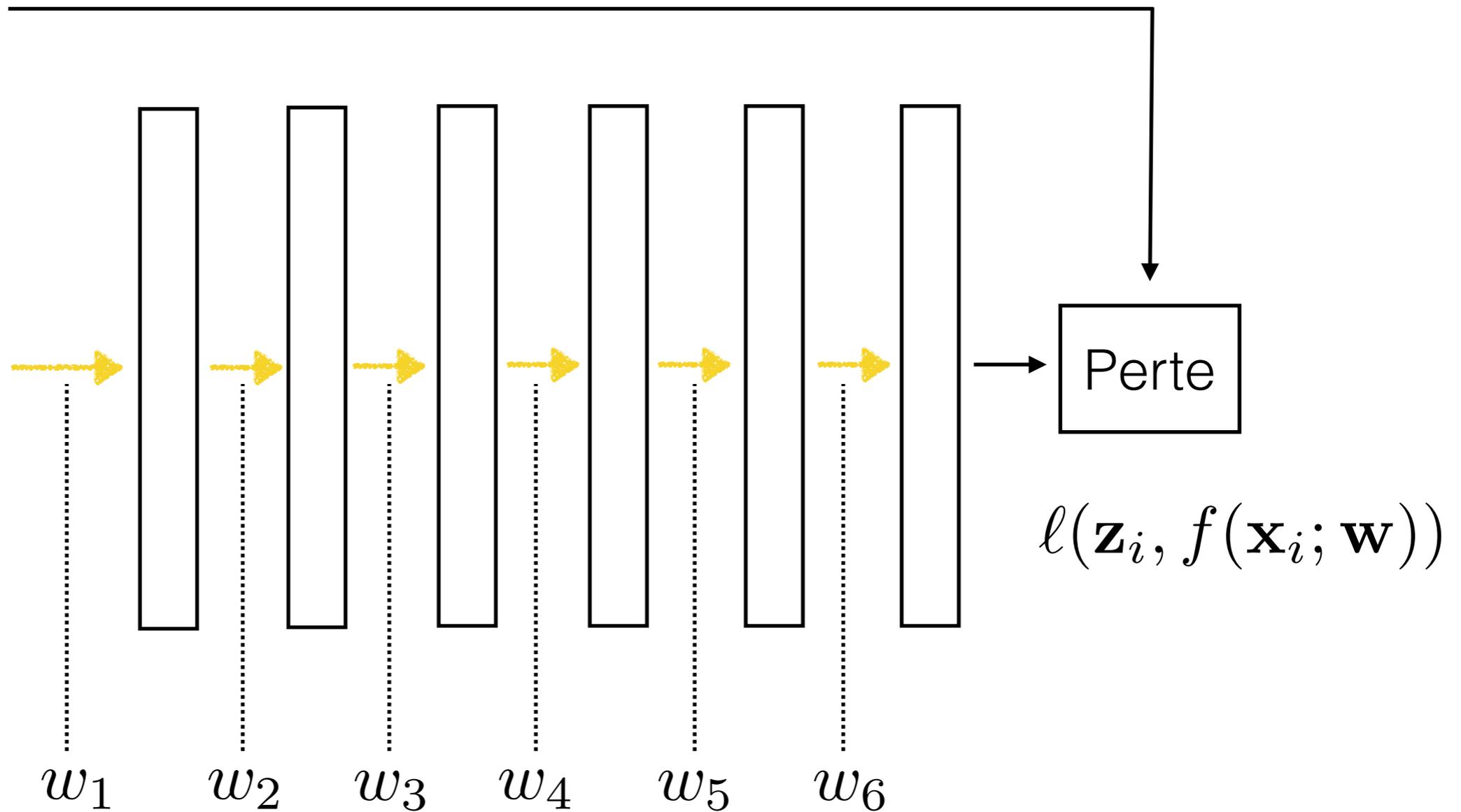
\mathbf{z}_2
«grizzly»



Apprentissage par réseaux profonds

Appris automatiquement

\mathbf{z}_i
«caméléon»



$$\operatorname{argmin}_{\mathbf{w}} \sum_i l(\mathbf{z}_i, f(\mathbf{x}_i; \mathbf{w}))$$

Descente du gradient

$$\operatorname{argmin}_{\mathbf{w}} \sum_i \ell(\mathbf{z}_i, f(\mathbf{x}_i; \mathbf{w})) = L(\mathbf{w})$$

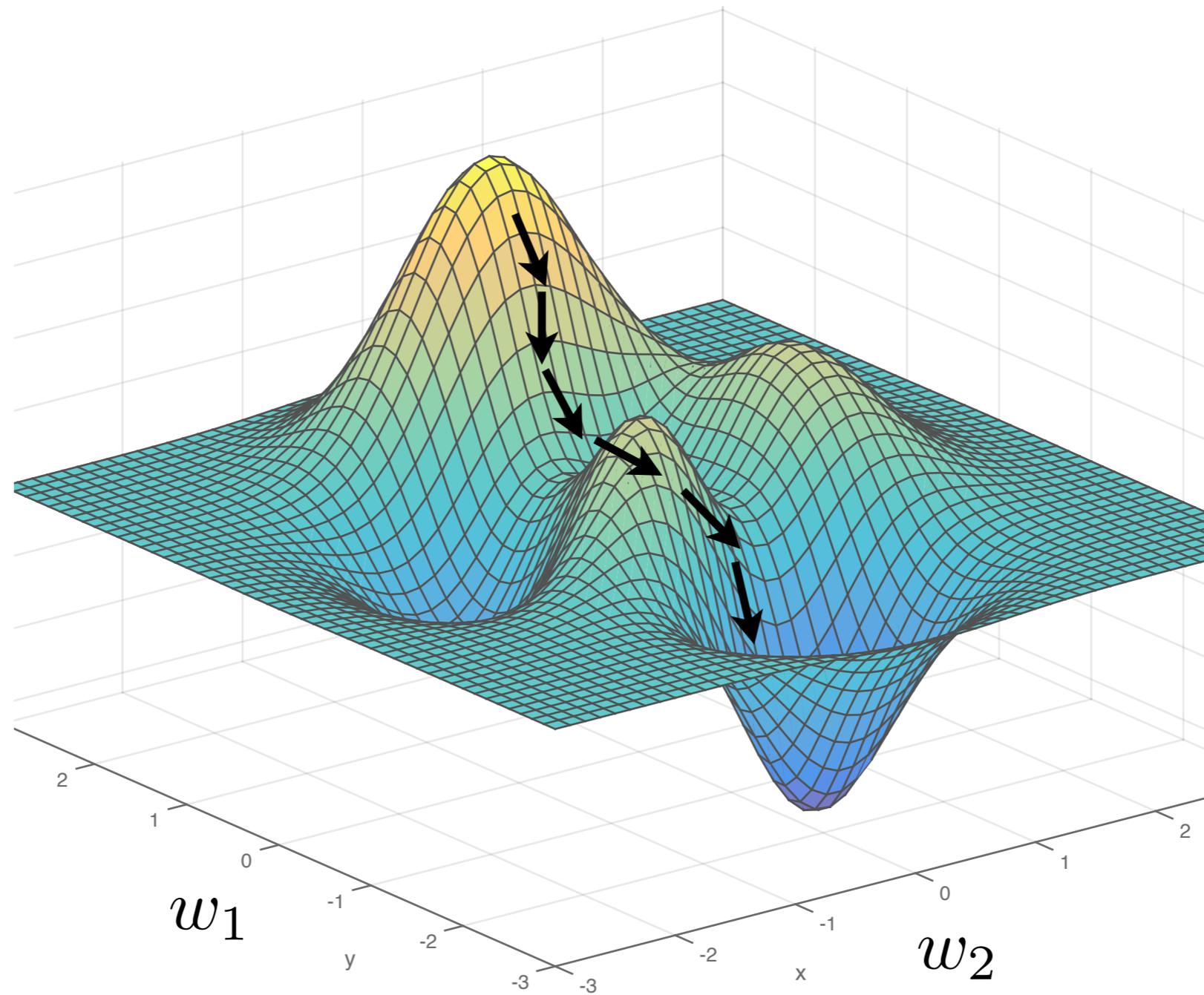
Une itération de descente du gradient

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta_t \frac{\partial L(\mathbf{w}^t)}{\partial \mathbf{w}}$$

taux d'apprentissage

Descente du gradient

$L(\mathbf{w})$



$$p(c|\mathbf{x})$$

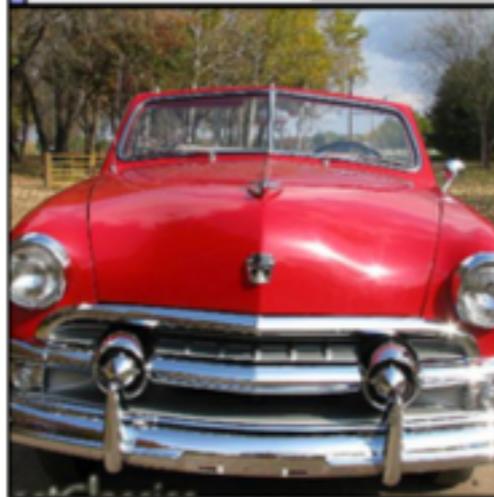


mite

container ship

motor scooter

leopard



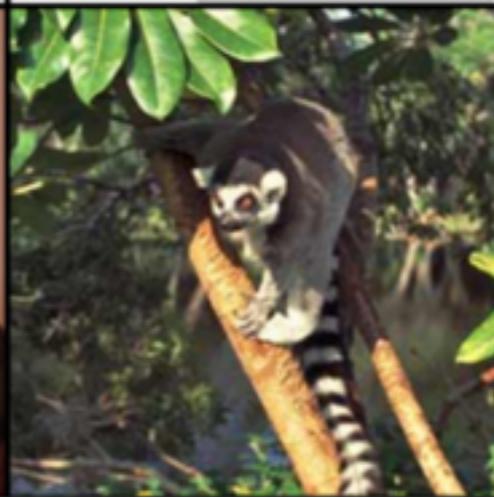
grille



mushroom



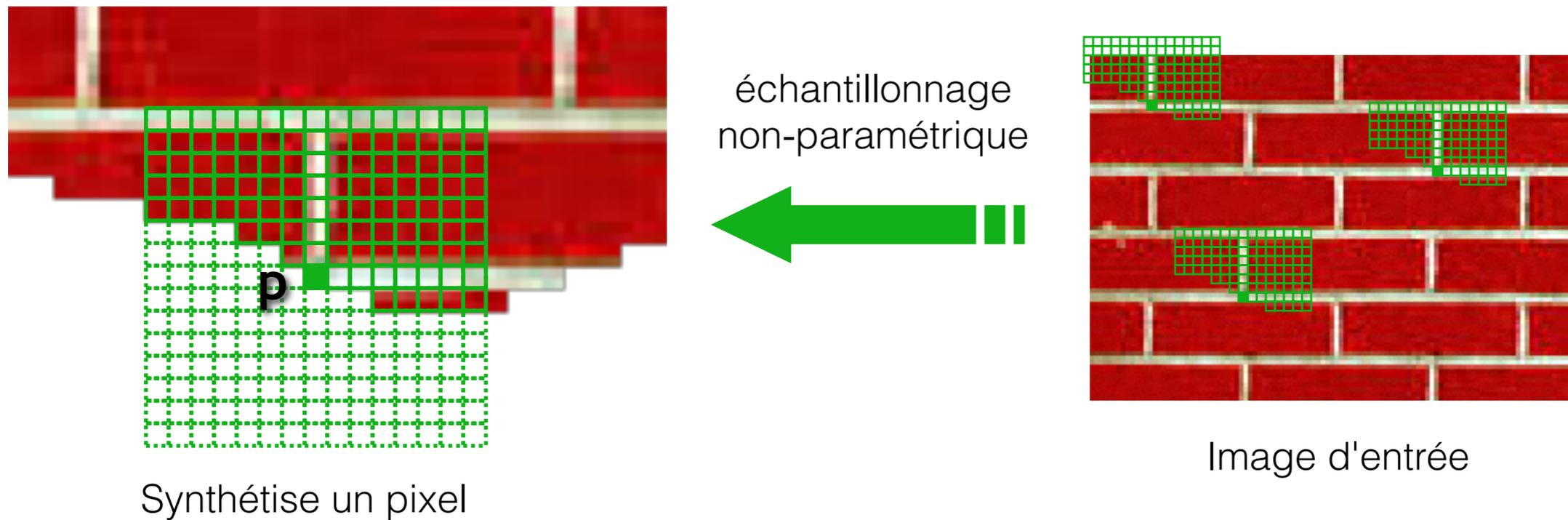
cherry



Madagascar cat

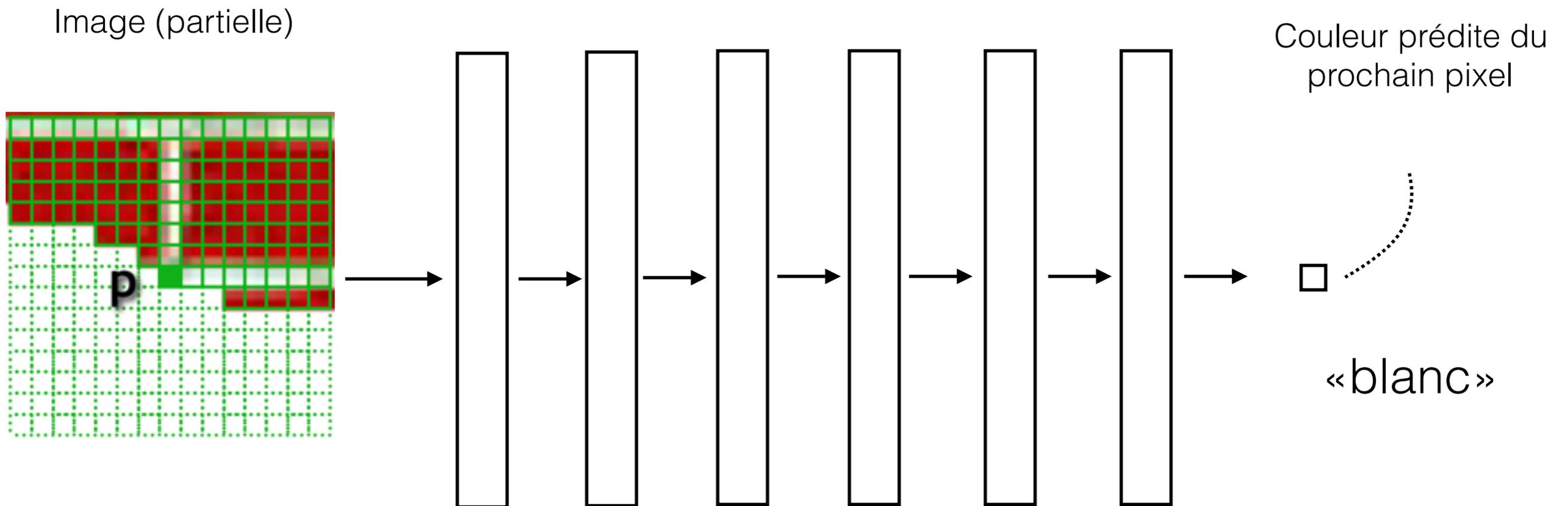


Synthèse de texture: rappel

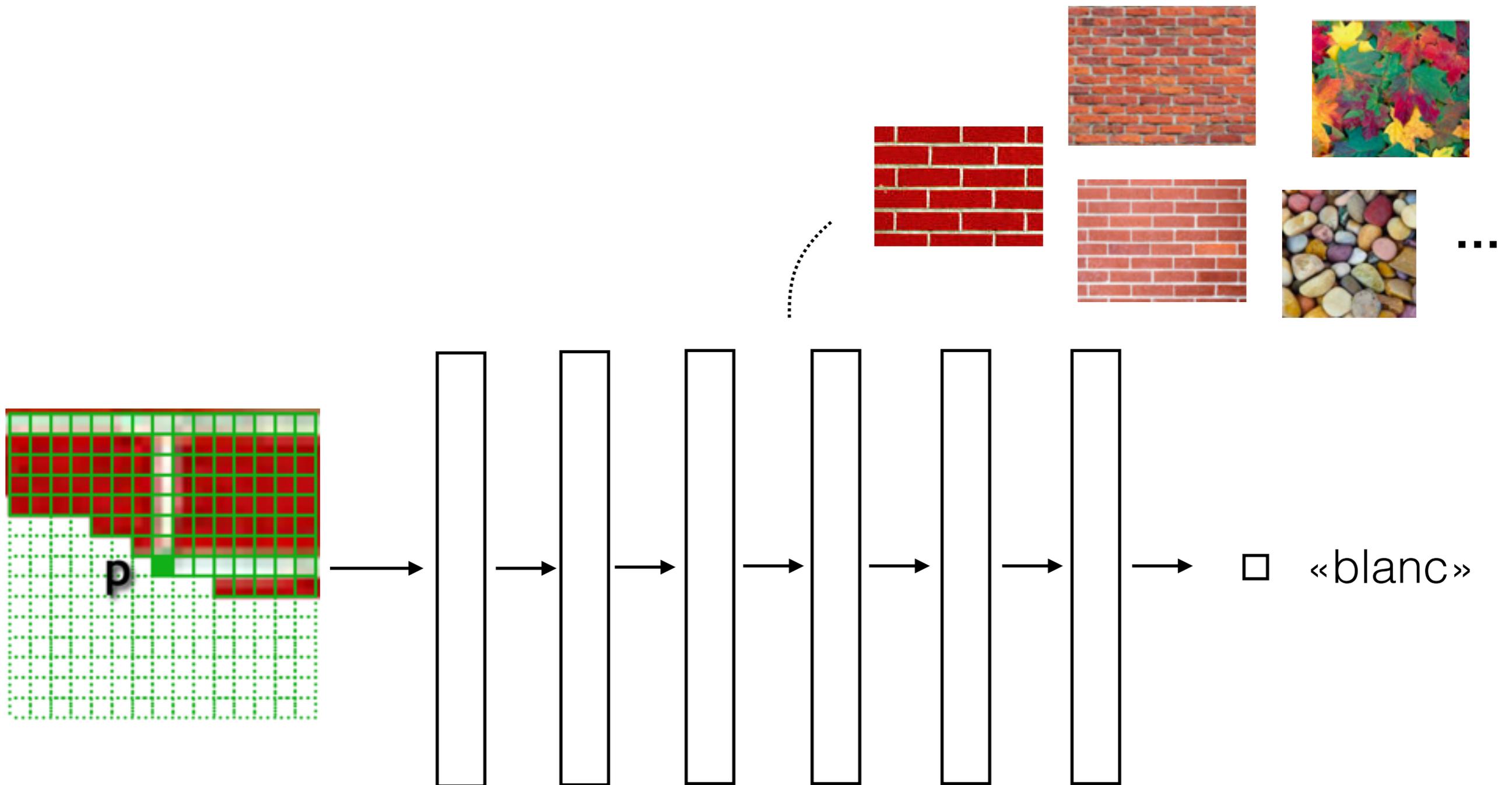


Modélise $P(p|N(p))$

Synthèse de texture par réseau profond

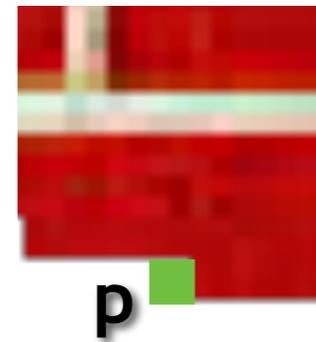
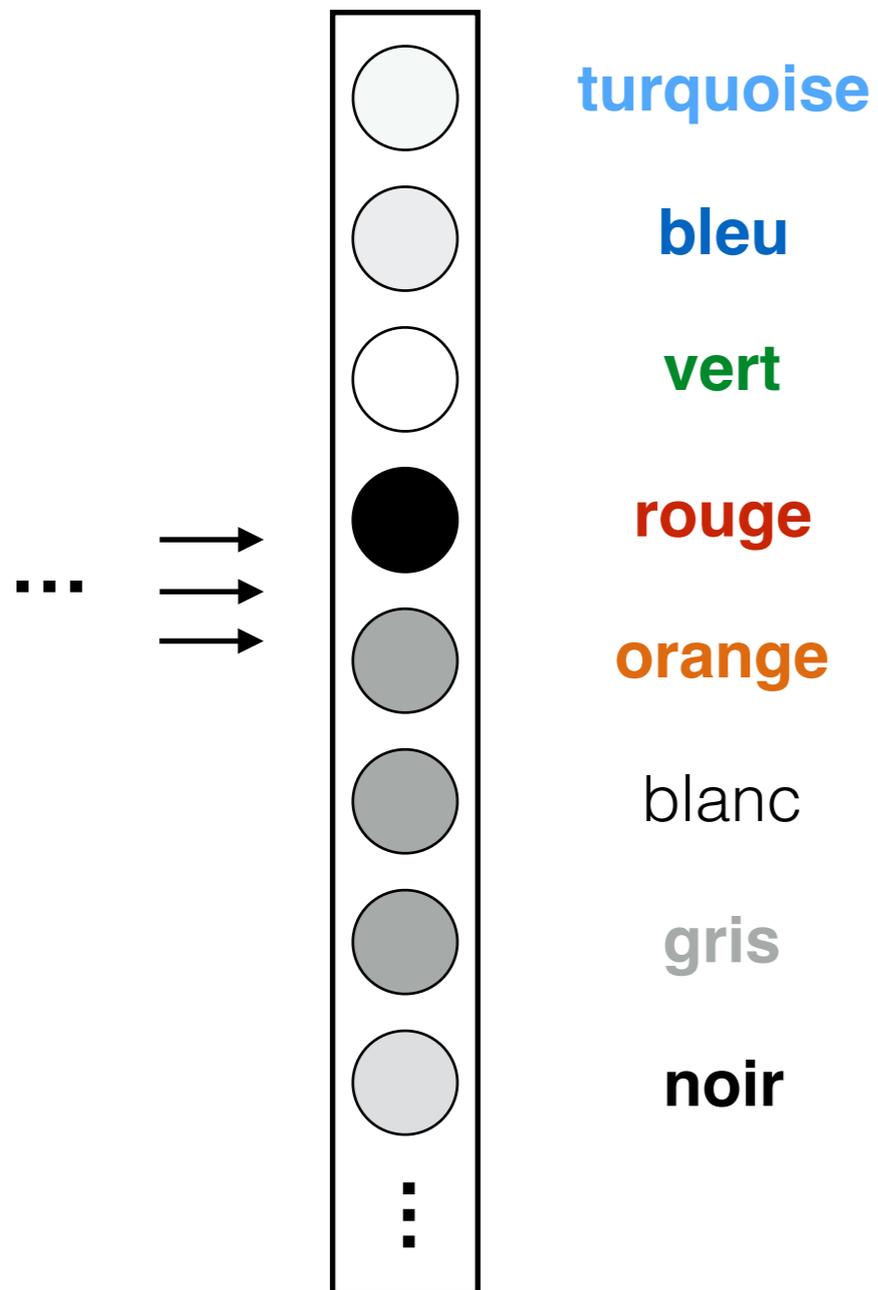


Synthèse de texture par réseau profond



Échantillonnage

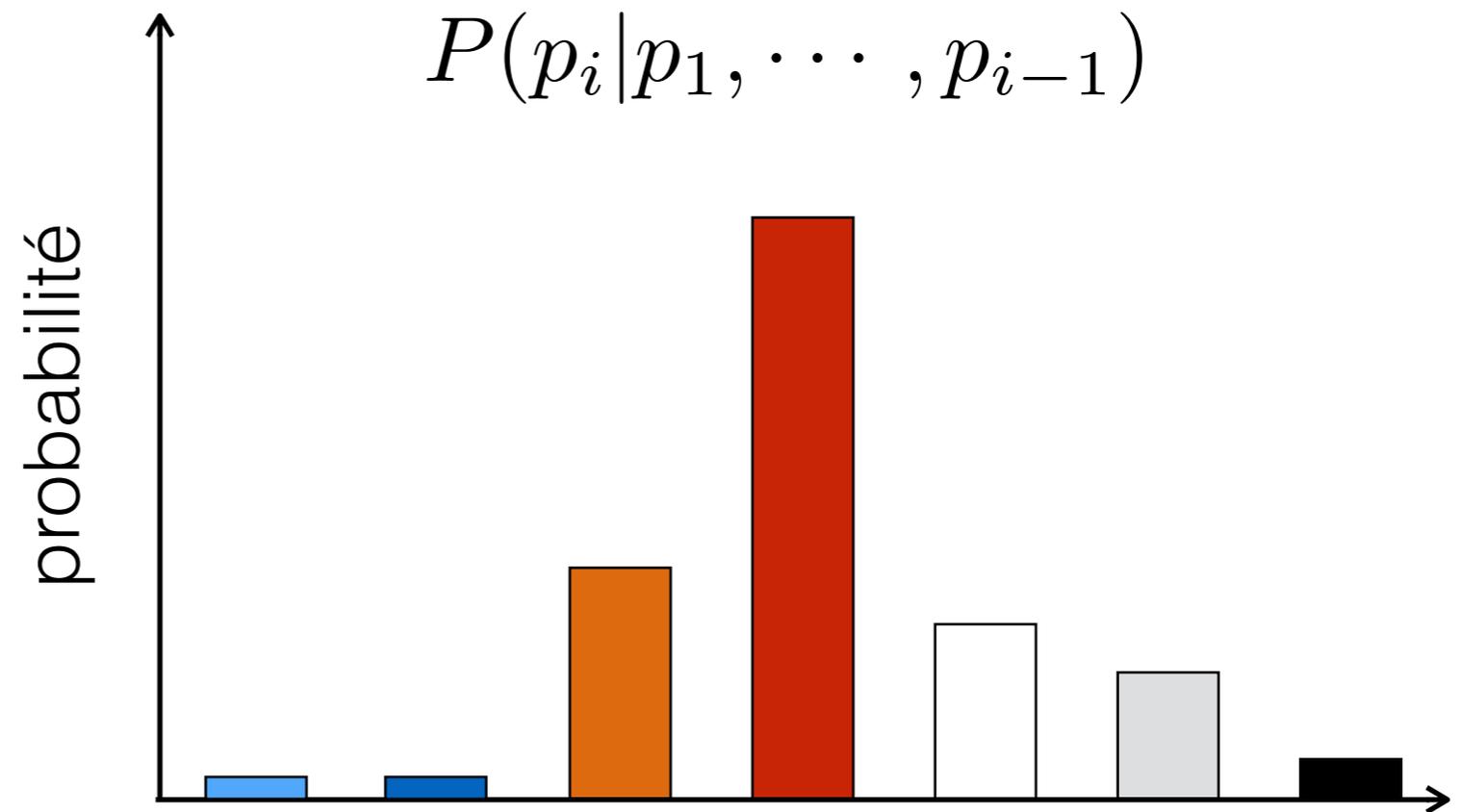
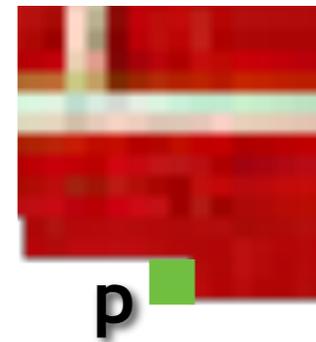
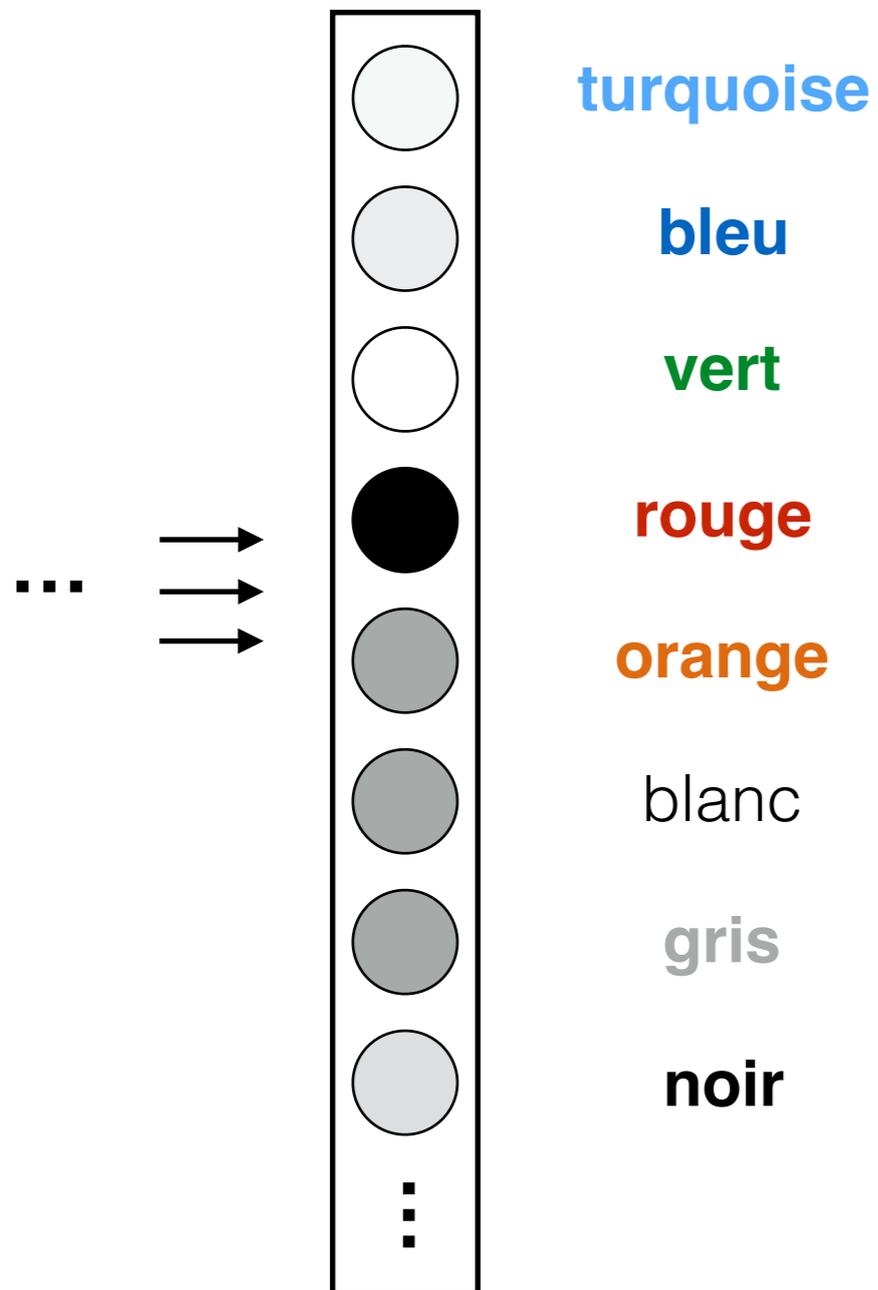
Sortie du réseau



$$P(p_i | p_1, \dots, p_{i-1})$$

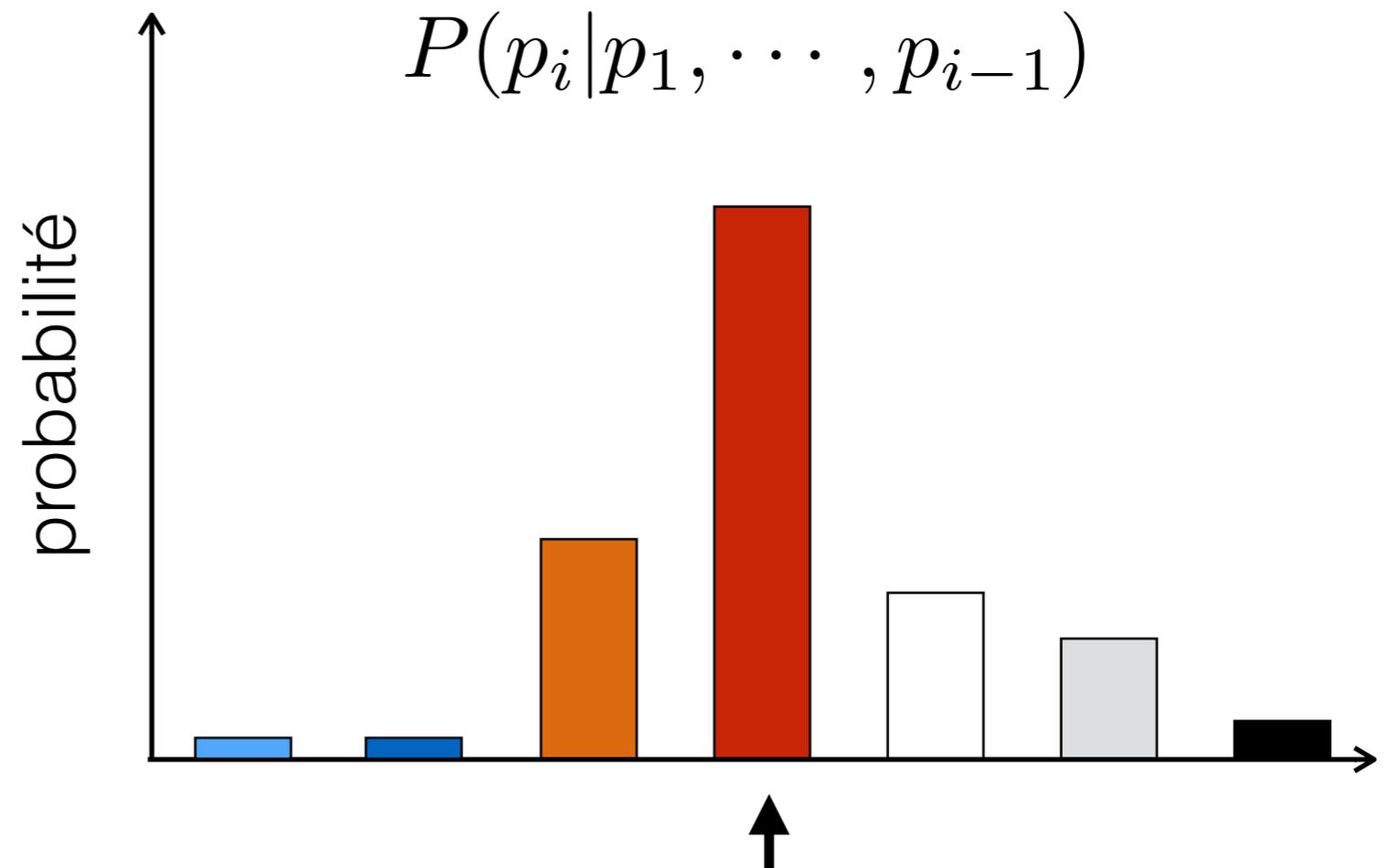
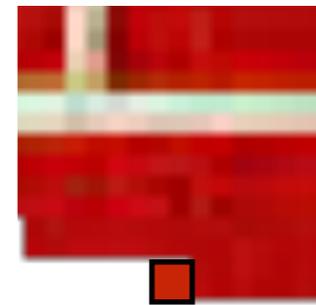
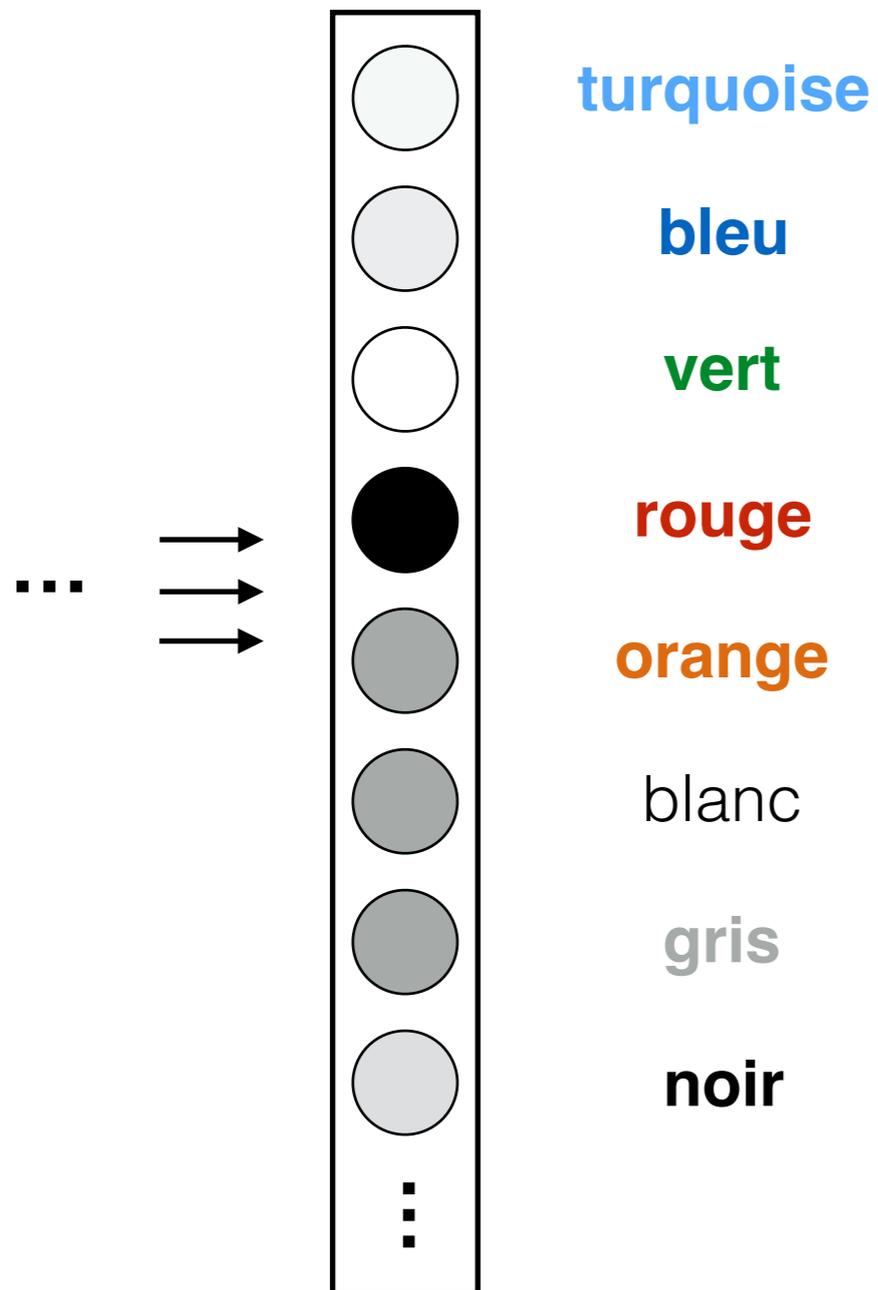
Échantillonnage

Sortie du réseau



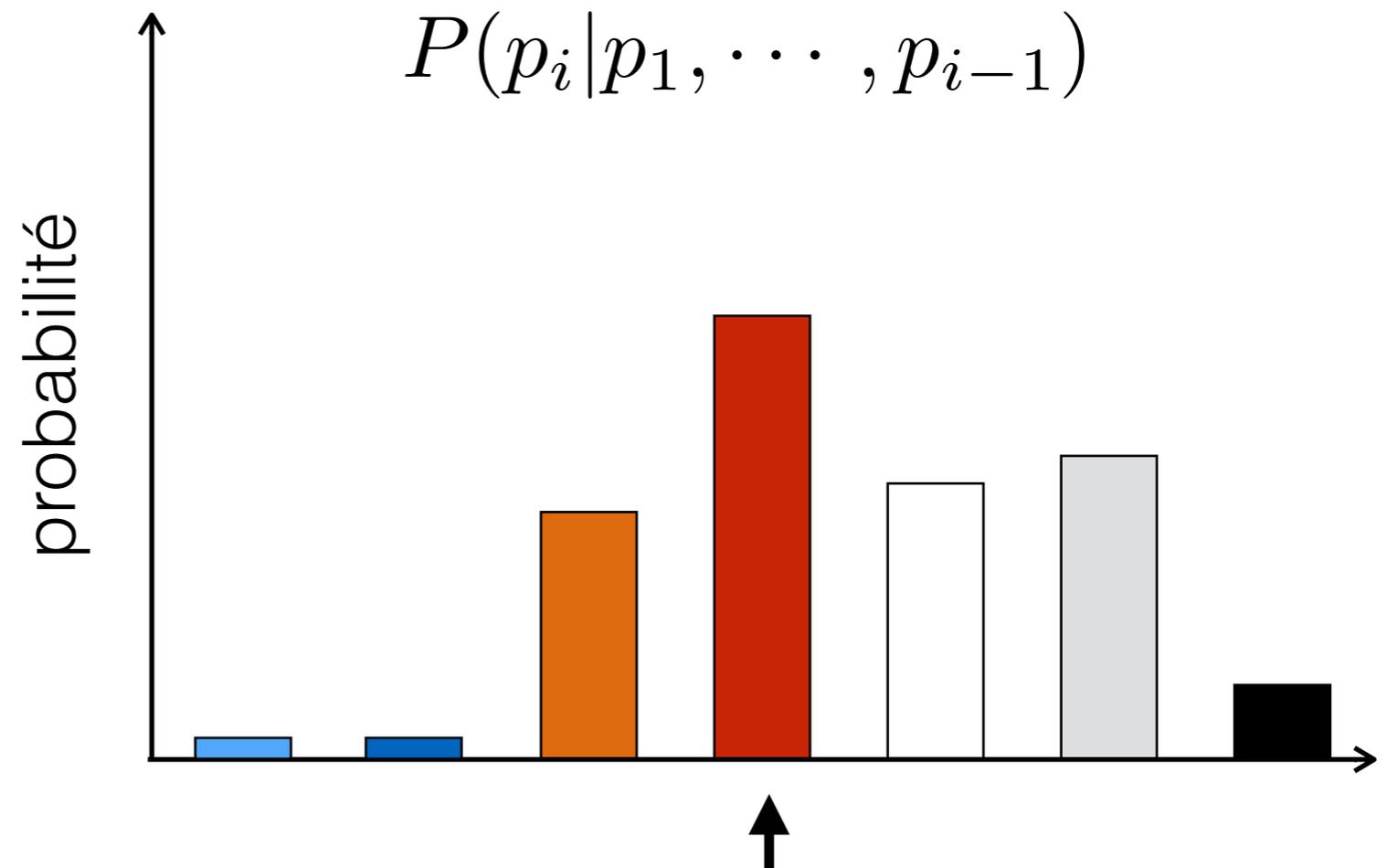
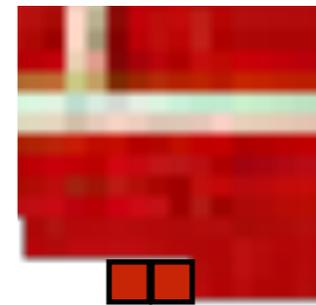
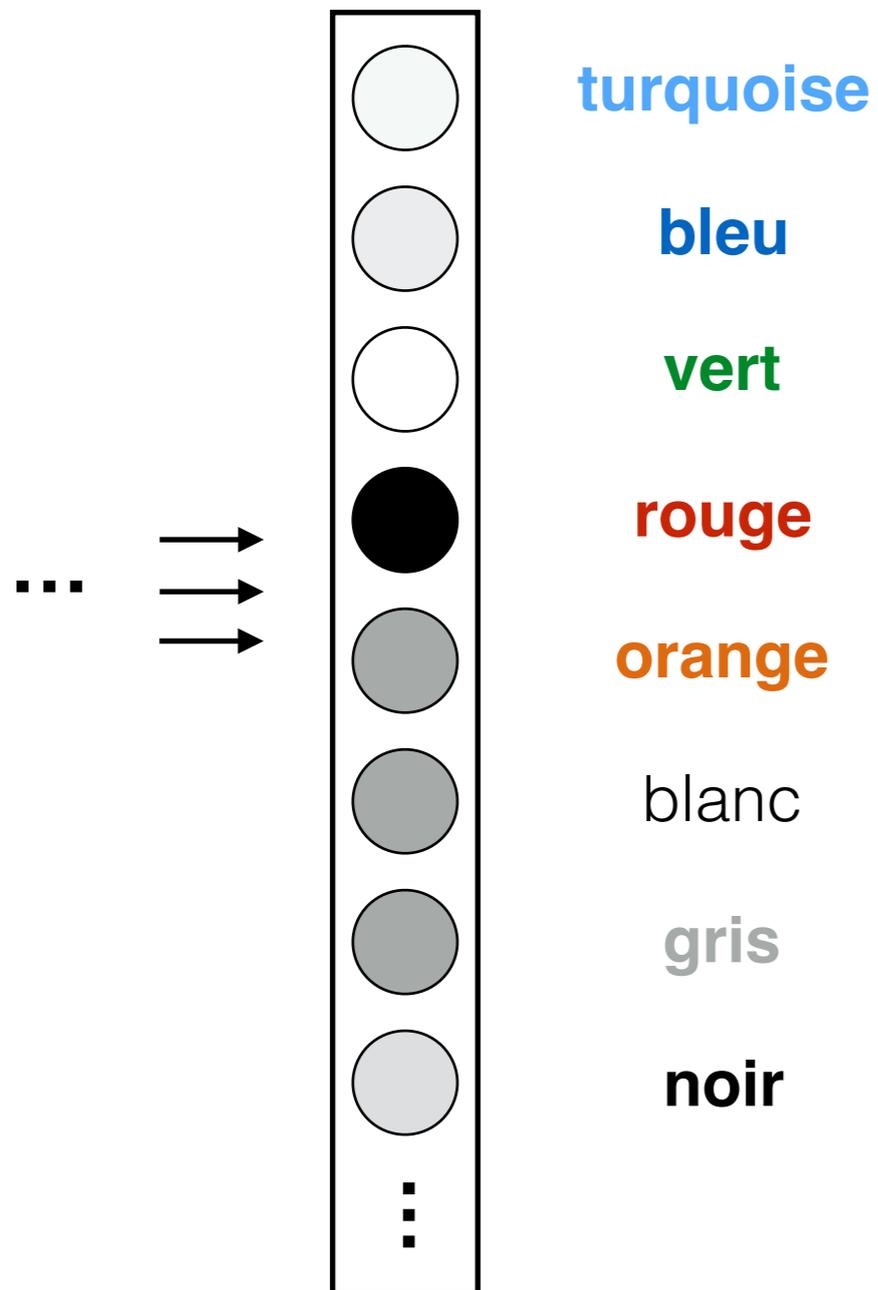
Échantillonnage

Sortie du réseau



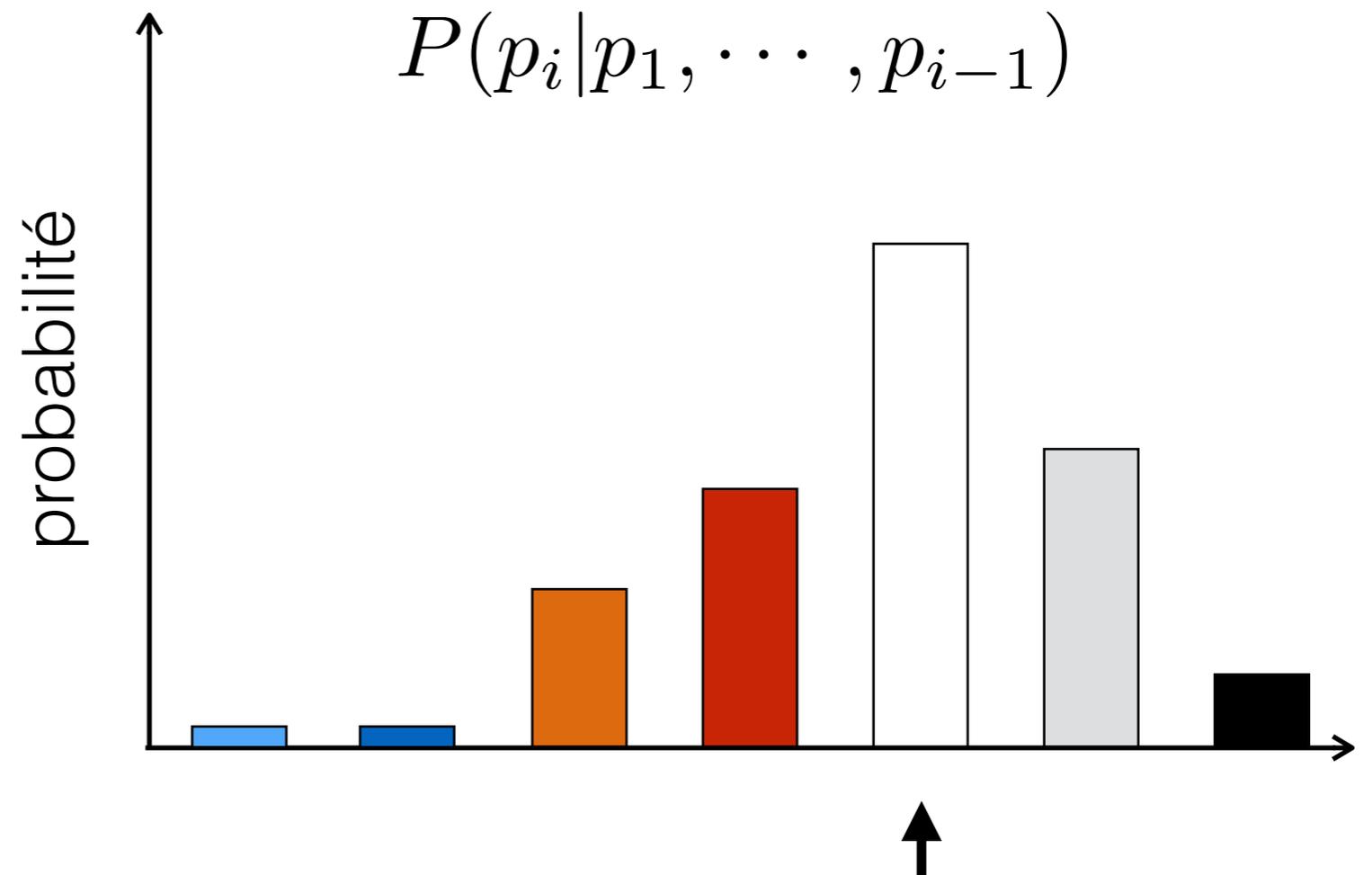
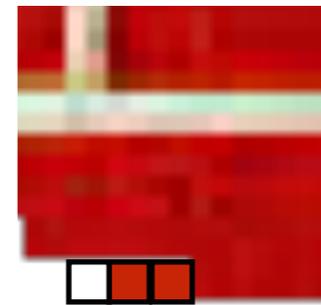
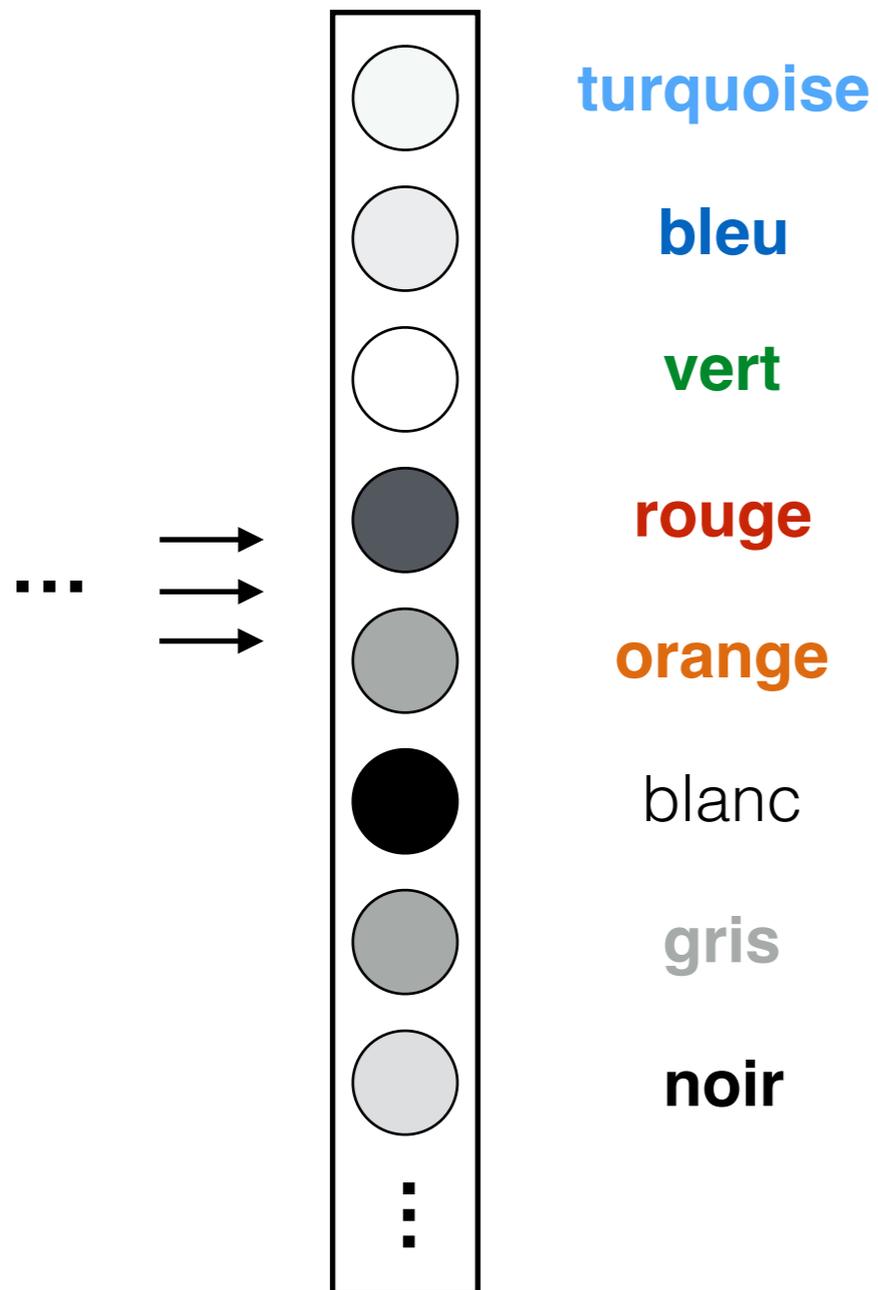
Échantillonnage

Sortie du réseau



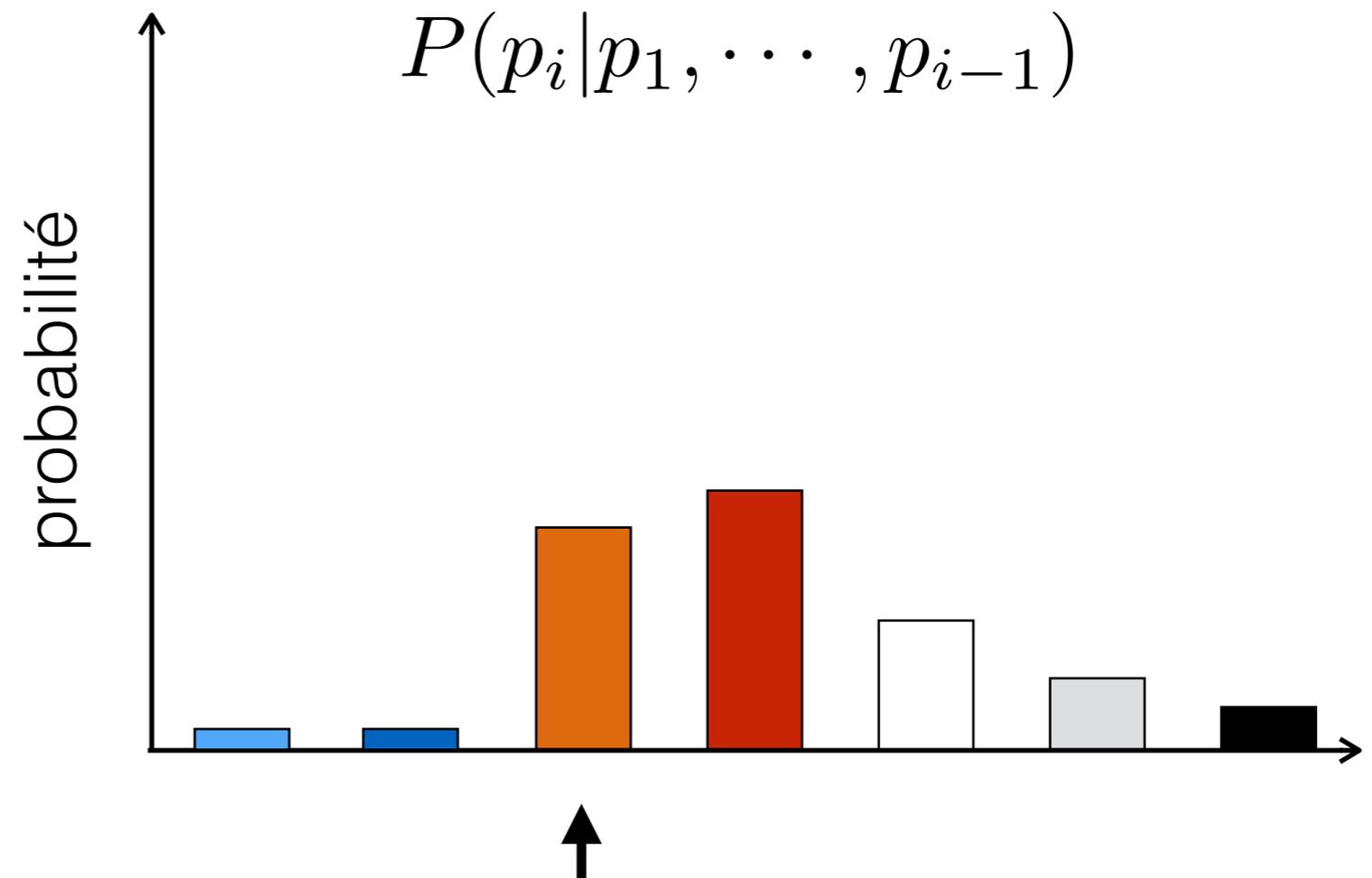
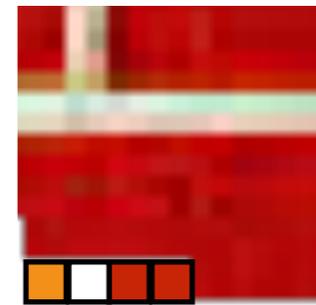
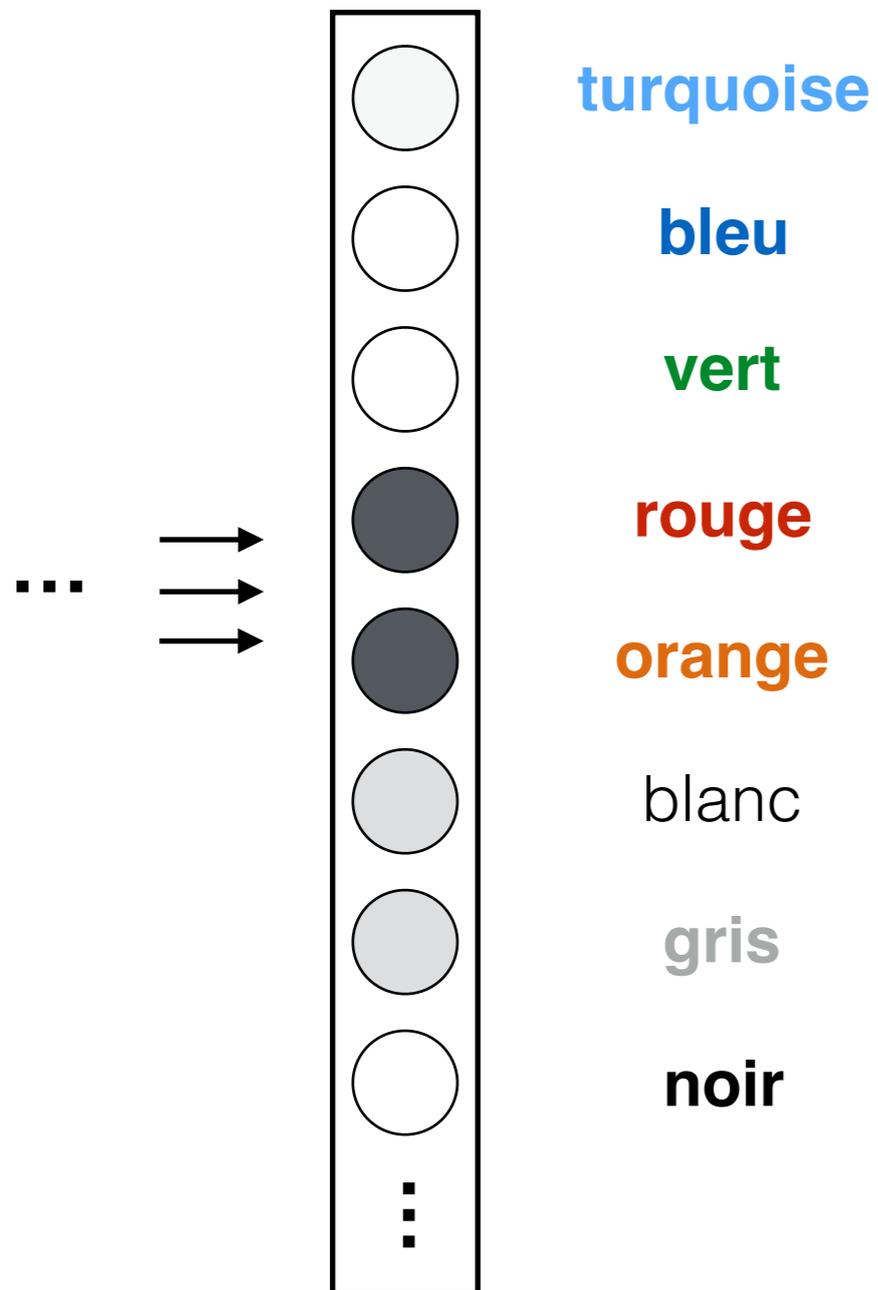
Échantillonnage

Sortie du réseau



Échantillonnage

Sortie du réseau

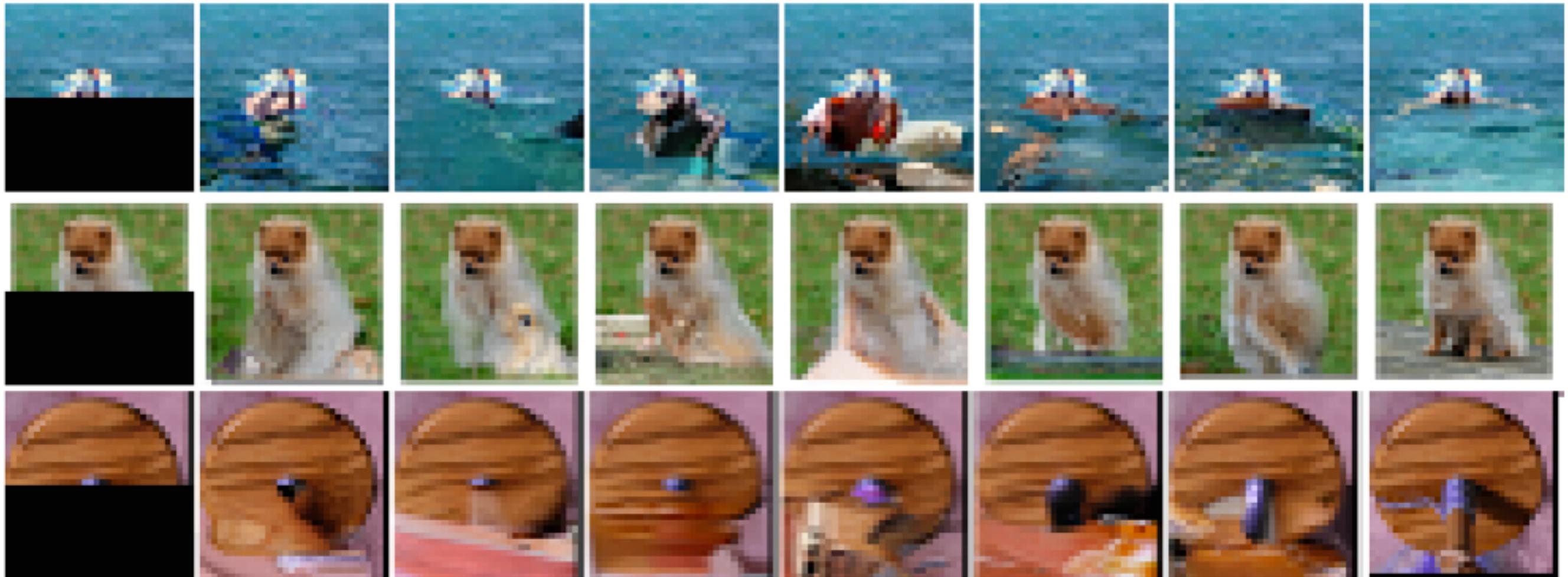


Compléter une image

occlusion

résultats

image
originale





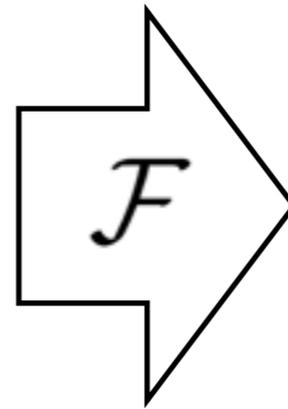
Ansel Adams, Yosemite Valley Bridge



[Zhang et al. 2016]



Image d'entrée: canal L seulement



Information en couleurs: canaux ab (de Lab)

$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$

$$\hat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times 2}$$

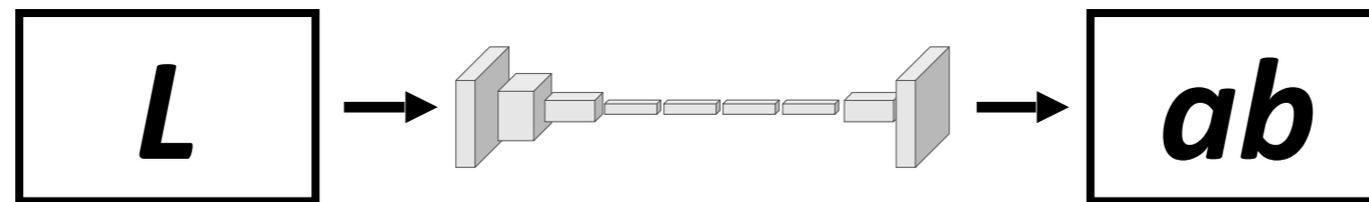
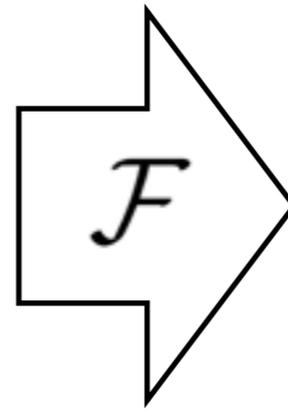




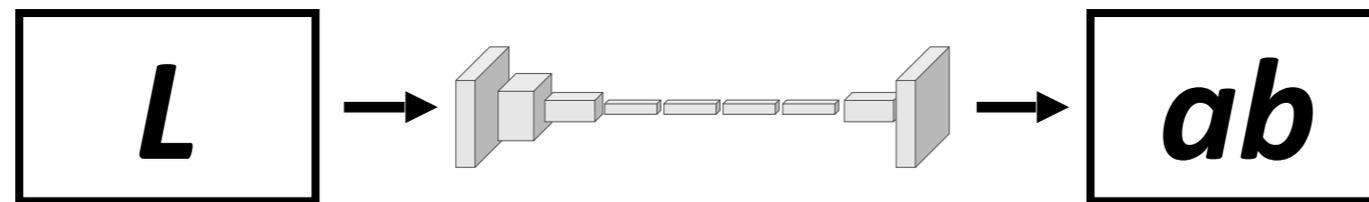
Image: canal L (de Lab)



Concatenation (L,ab)

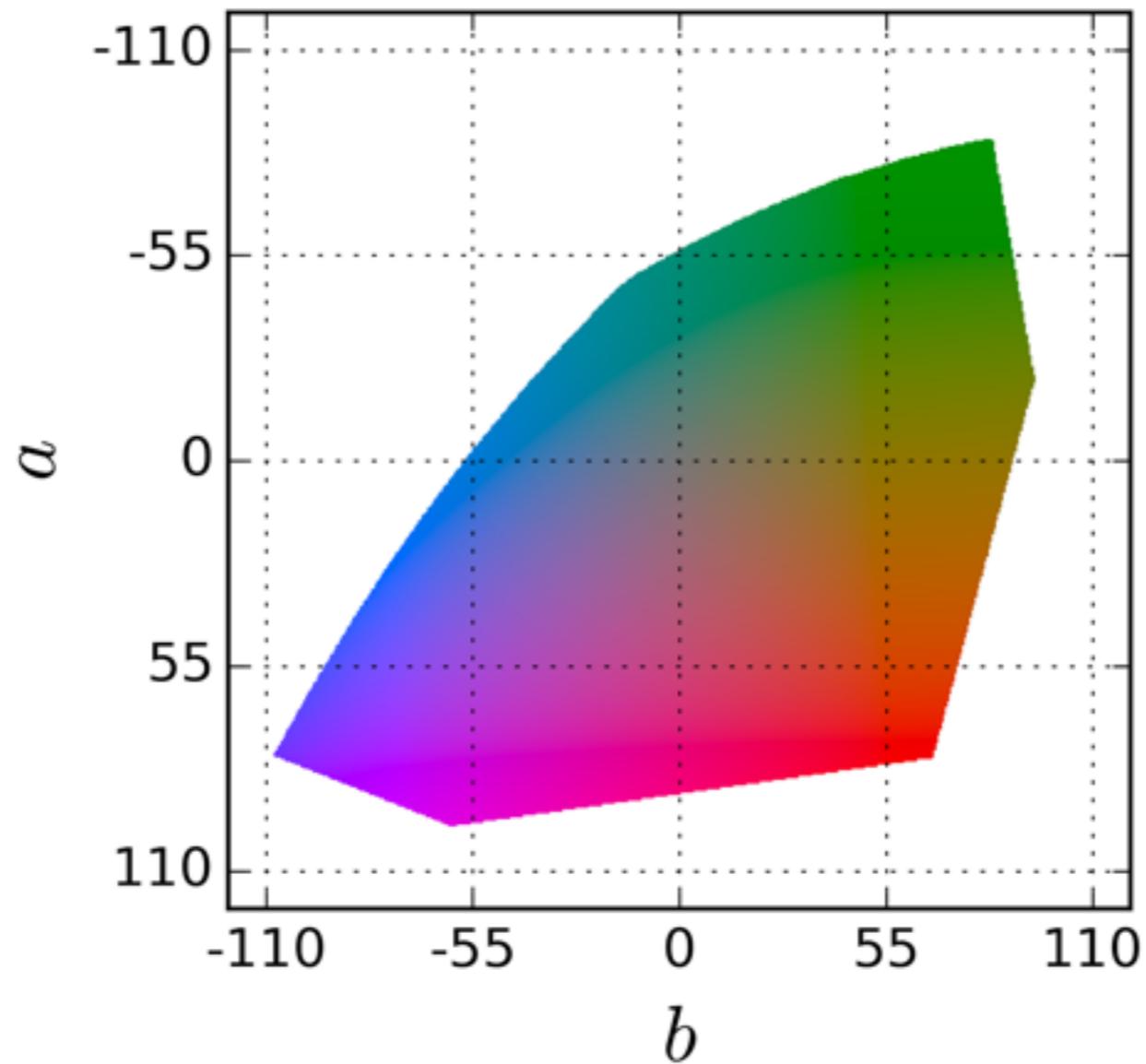
$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$

$$(\mathbf{X}, \hat{\mathbf{Y}})$$



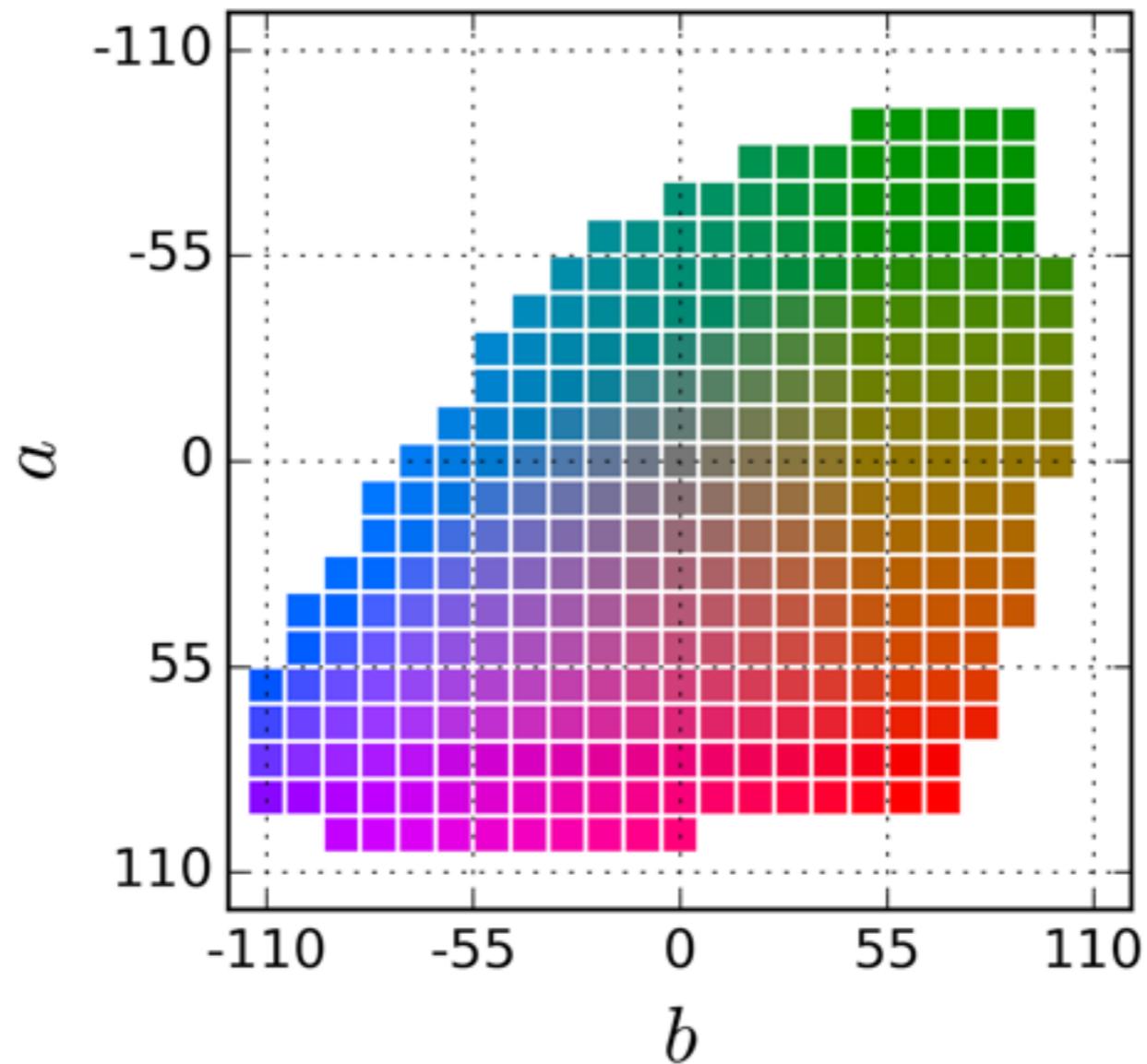
Meilleure fonction de perte

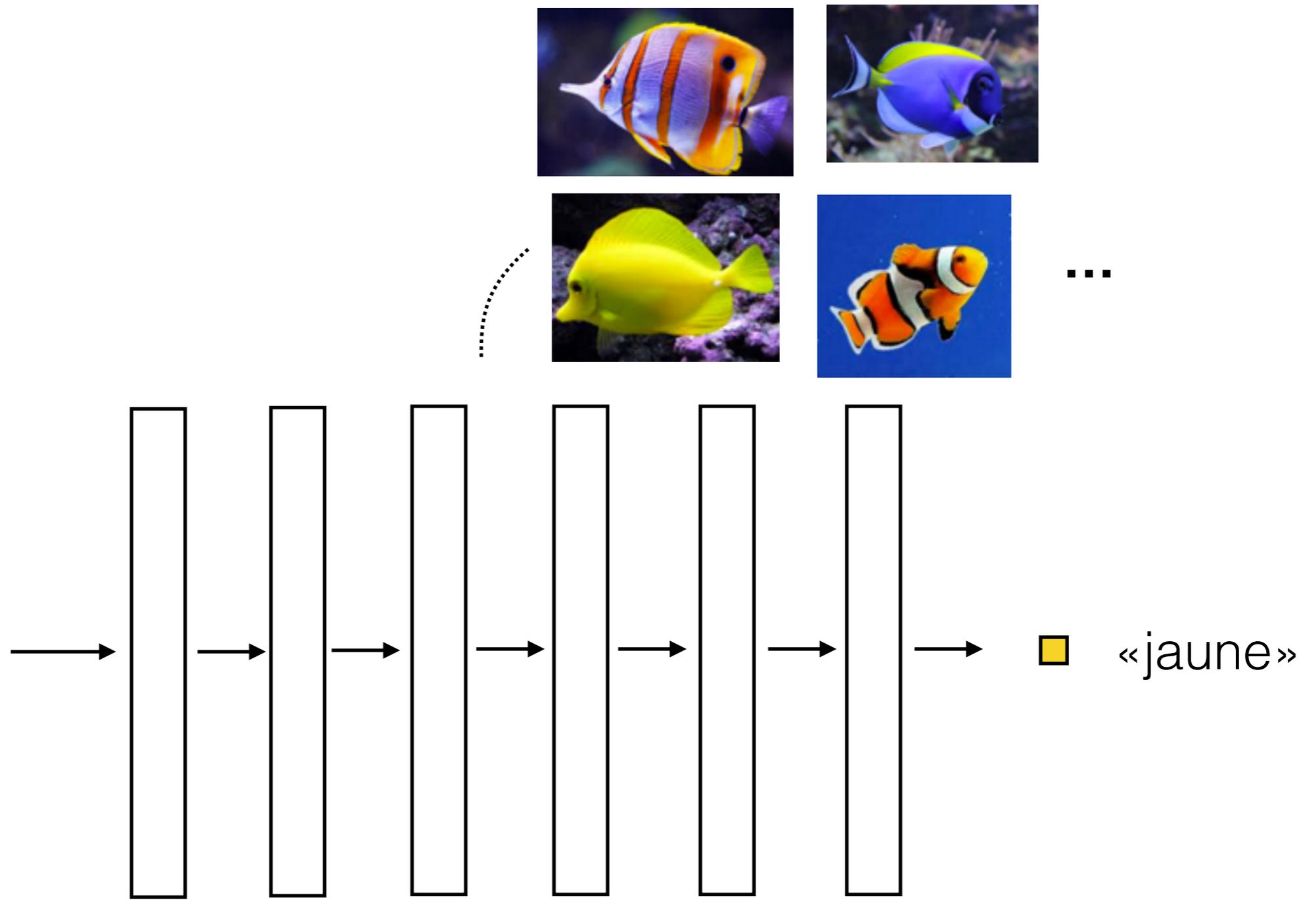
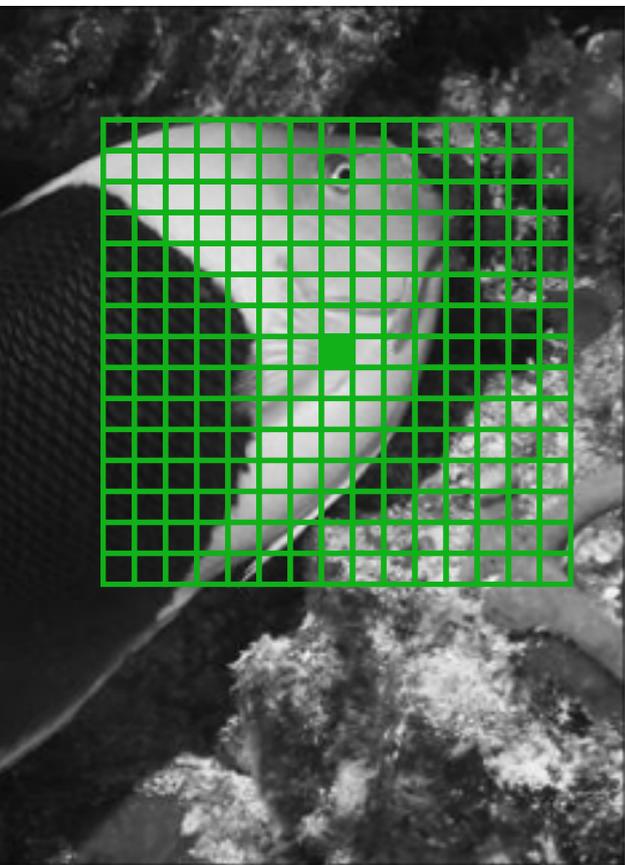
Espace (continue) des couleurs ab

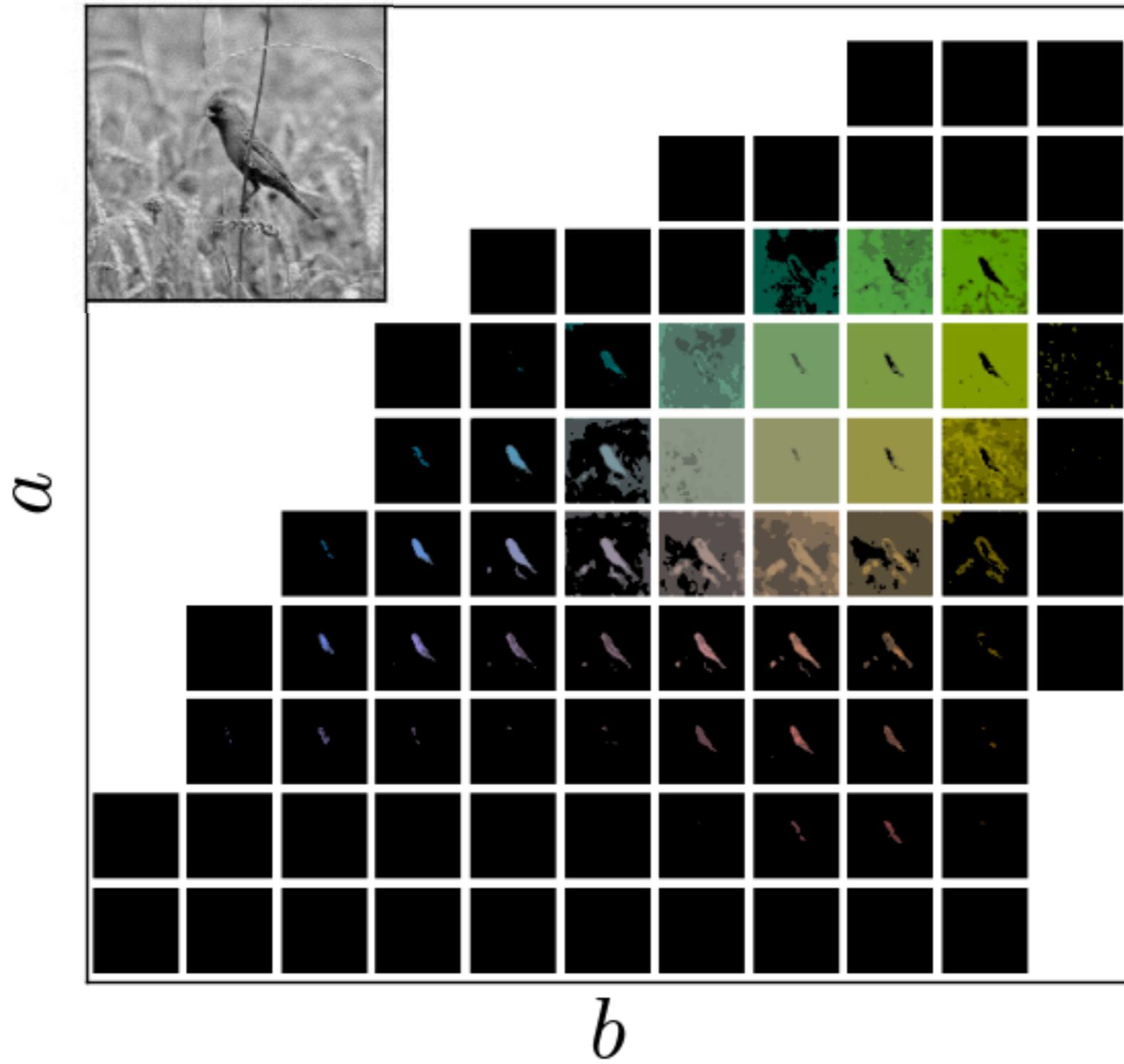


Meilleure fonction de perte

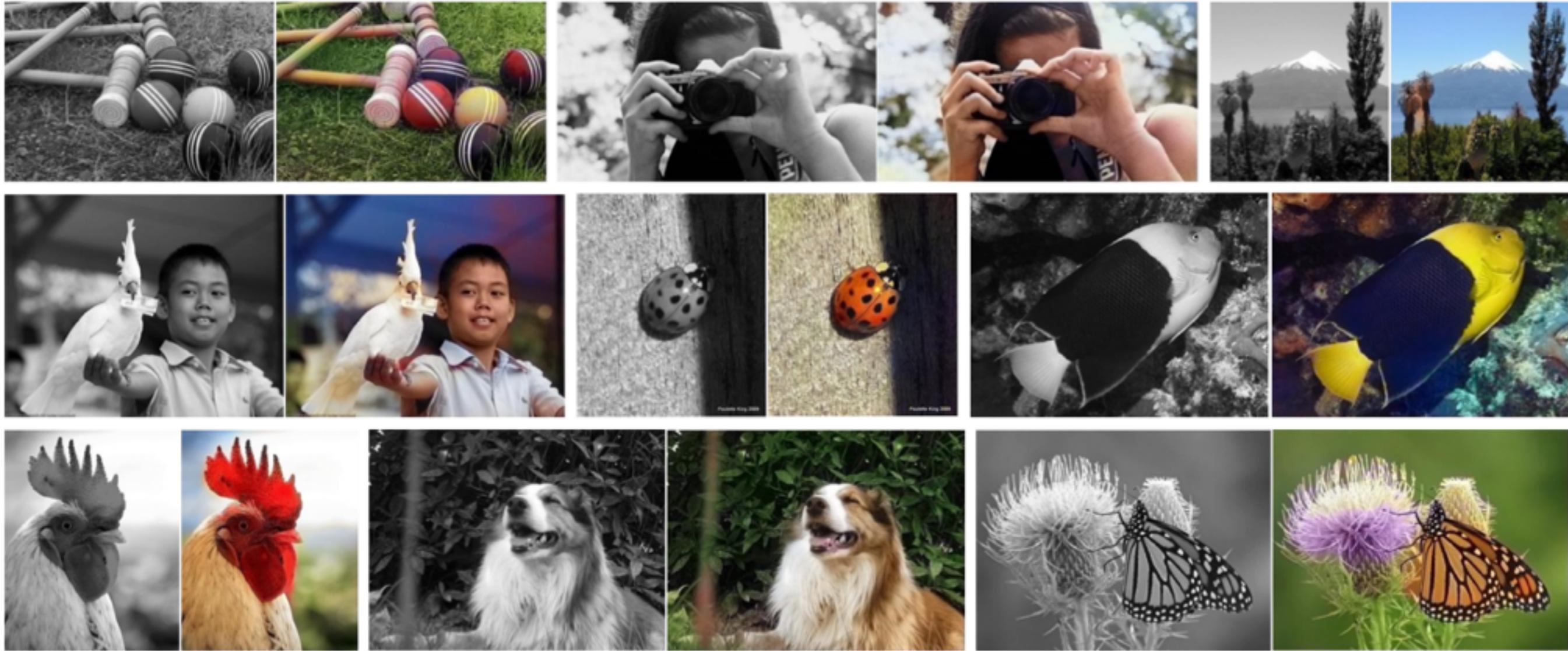
Discrétisation de l'espace de couleurs ab







Bons résultats



Moins bons résultats



Biais





Reddit /u/SherySantucci



Reddit ColorizeBot

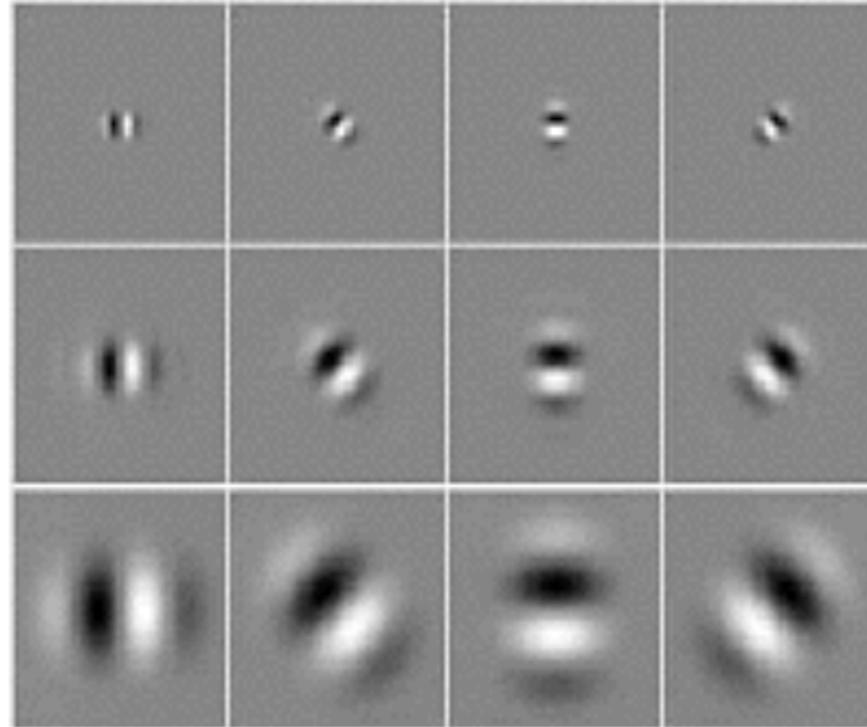


Photo de Reddit /u/Timteroo,
Murale de Eduardo Kobra
95



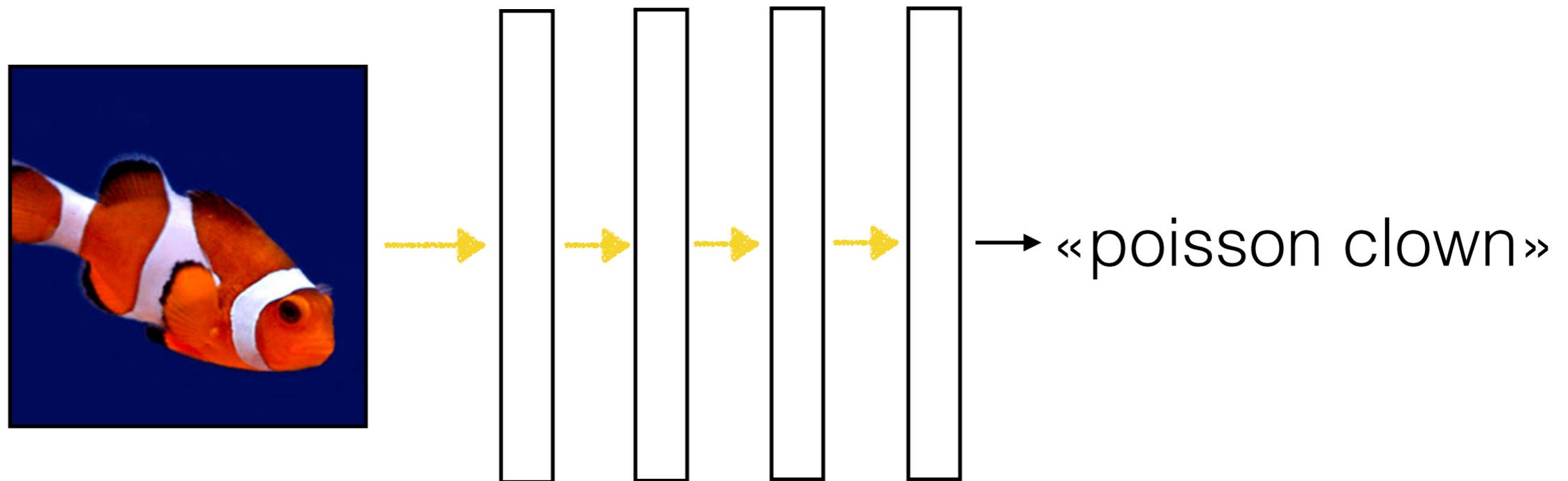
Reddit ColorizeBot

Représenter les textures



Pourquoi *ces* caractéristiques exactement?

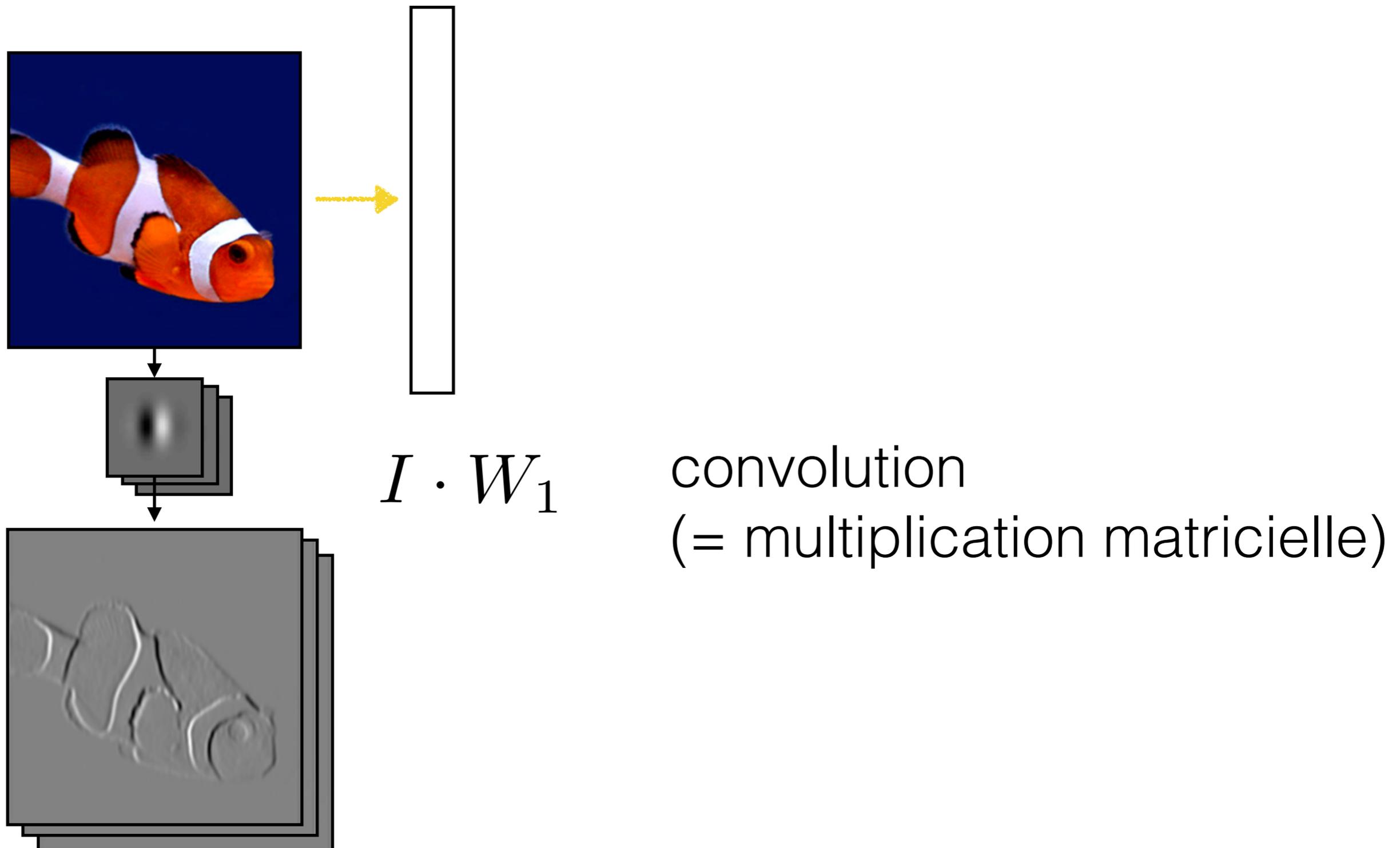
Choix de la représentation



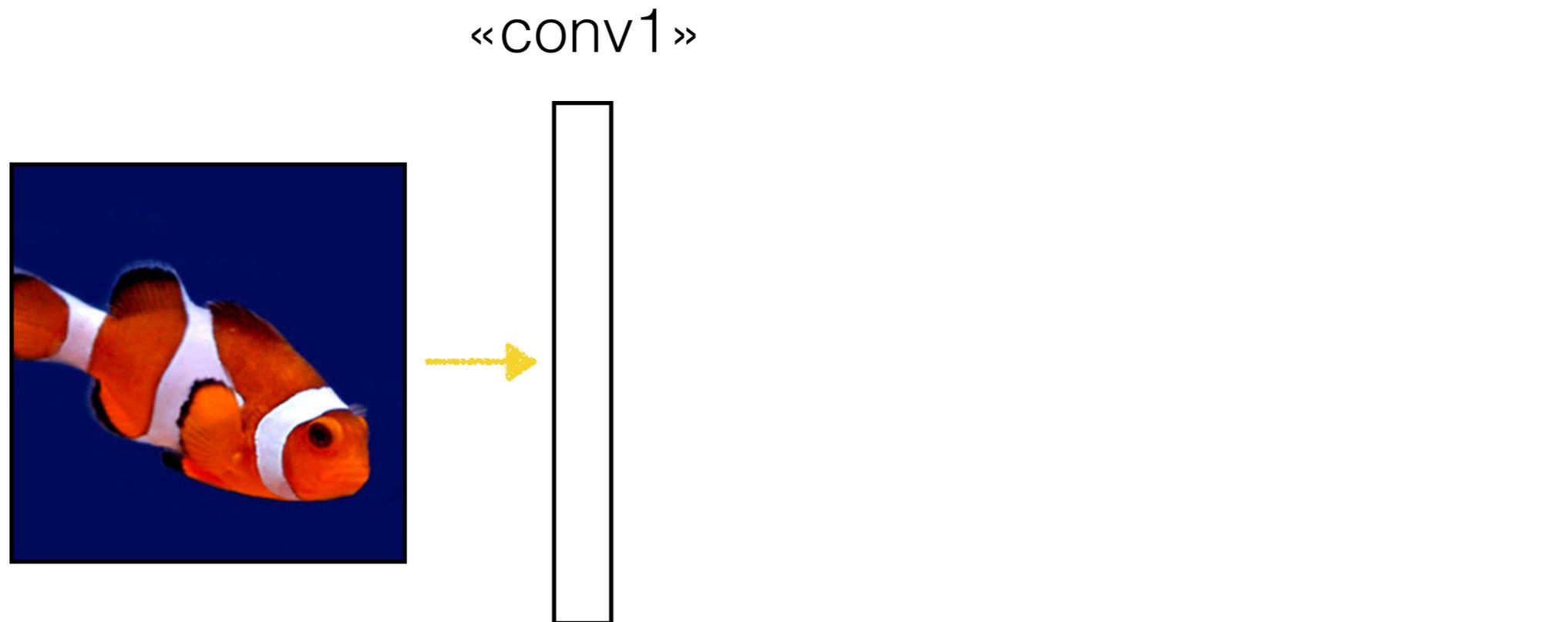
Au lieu de banques de filtres pré-déterminés, utilisons les caractéristiques apprises par un réseau de neurones!

Un réseau de neurone entraîné à reconnaître les objets devient un calculateur de caractéristiques très robustes.

Extraire les caractéristiques d'une image



Extraire les caractéristiques d'une image

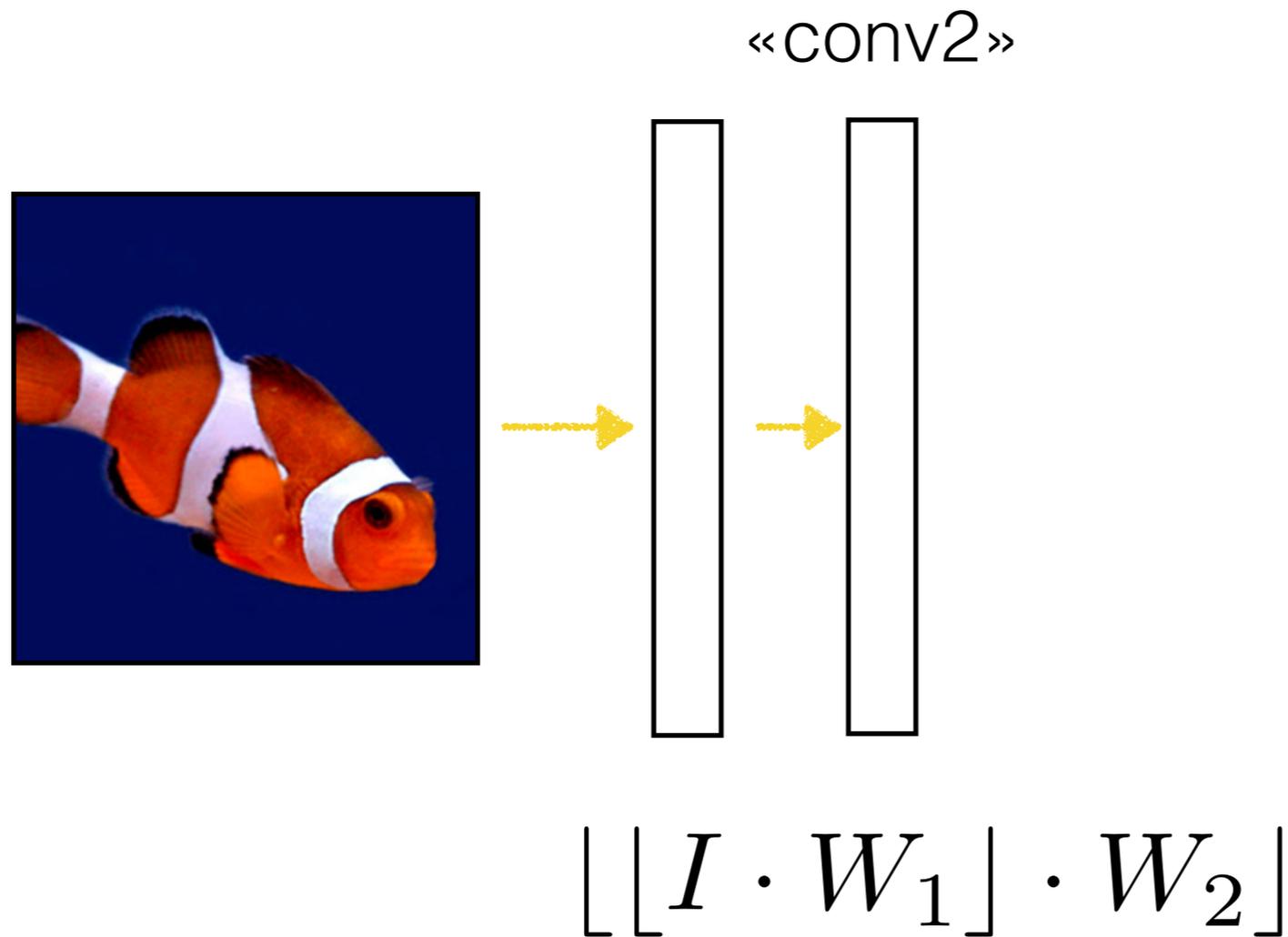


$[I \cdot W_1]$ convolution suivie d'une non-linéarité

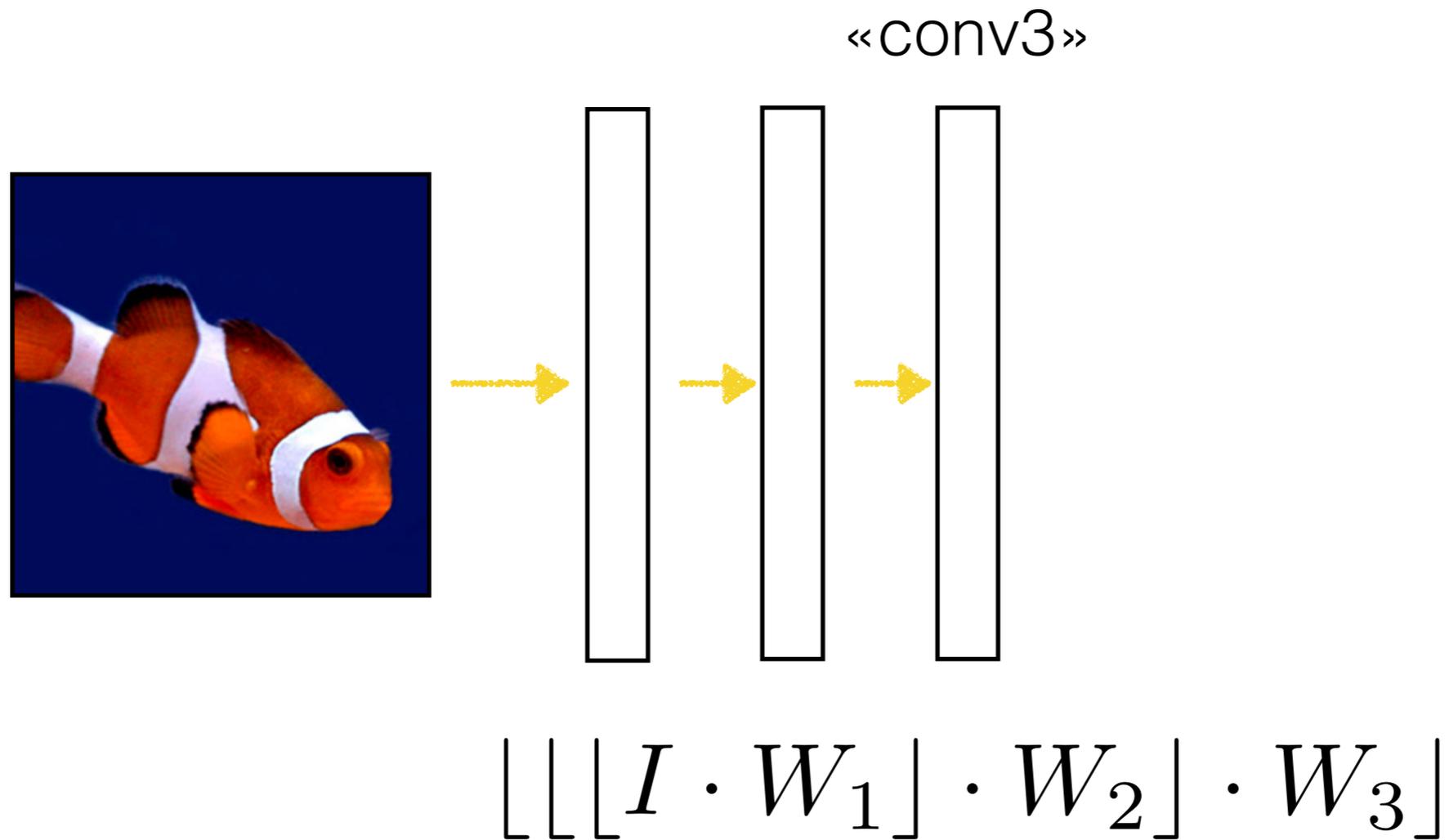
où $[\vec{x}]_i = \max(x_i, 0)$

i.e. valeurs négatives = 0

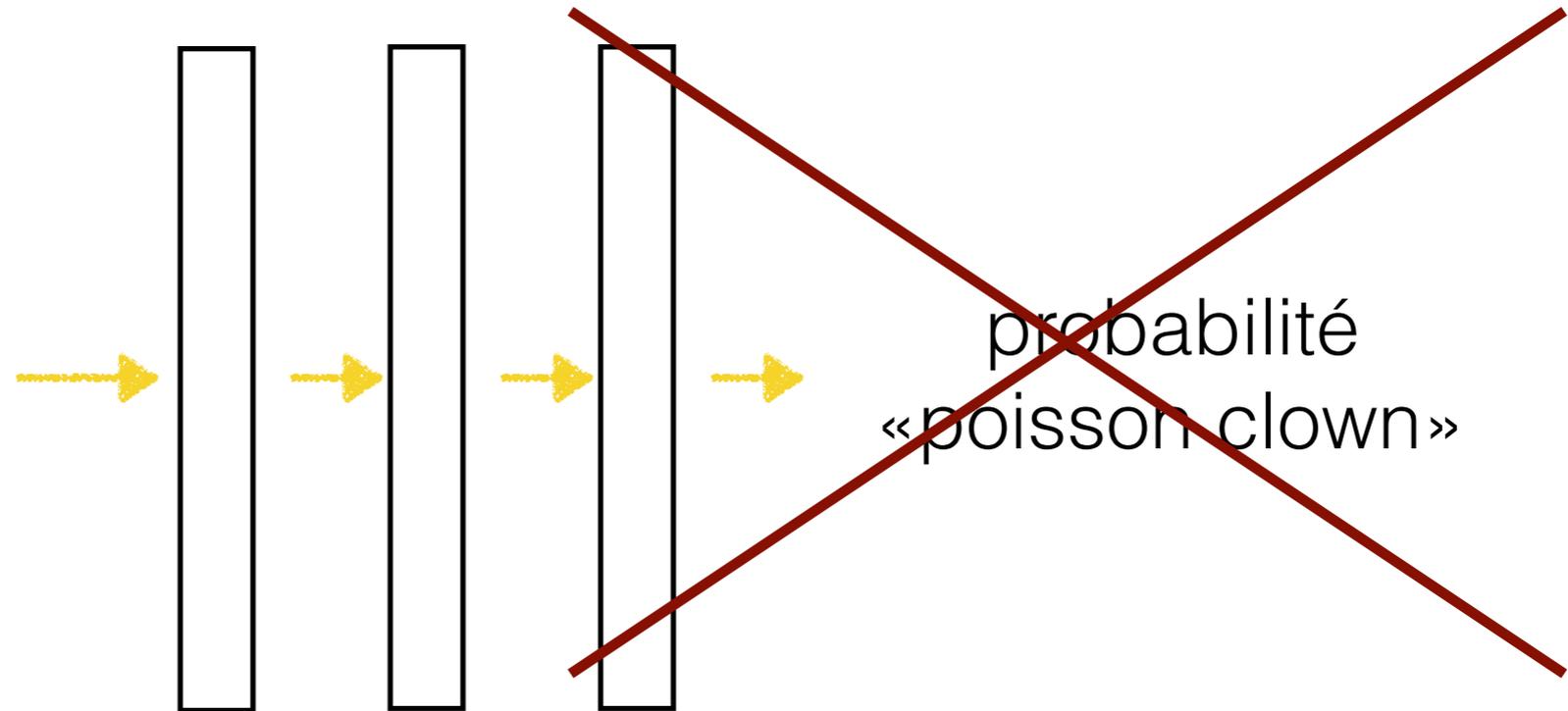
Extraire les caractéristiques d'une image



Extraire les caractéristiques d'une image

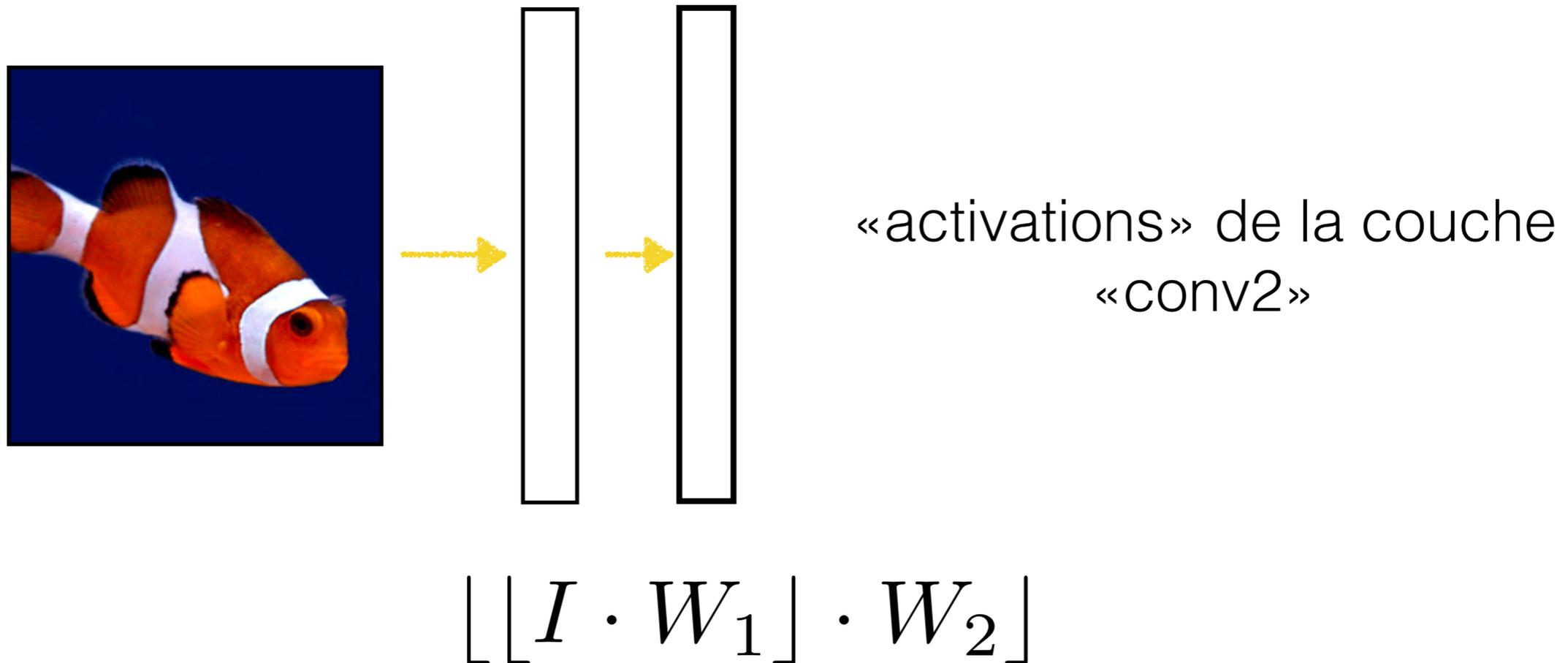


Extraire les caractéristiques d'une image

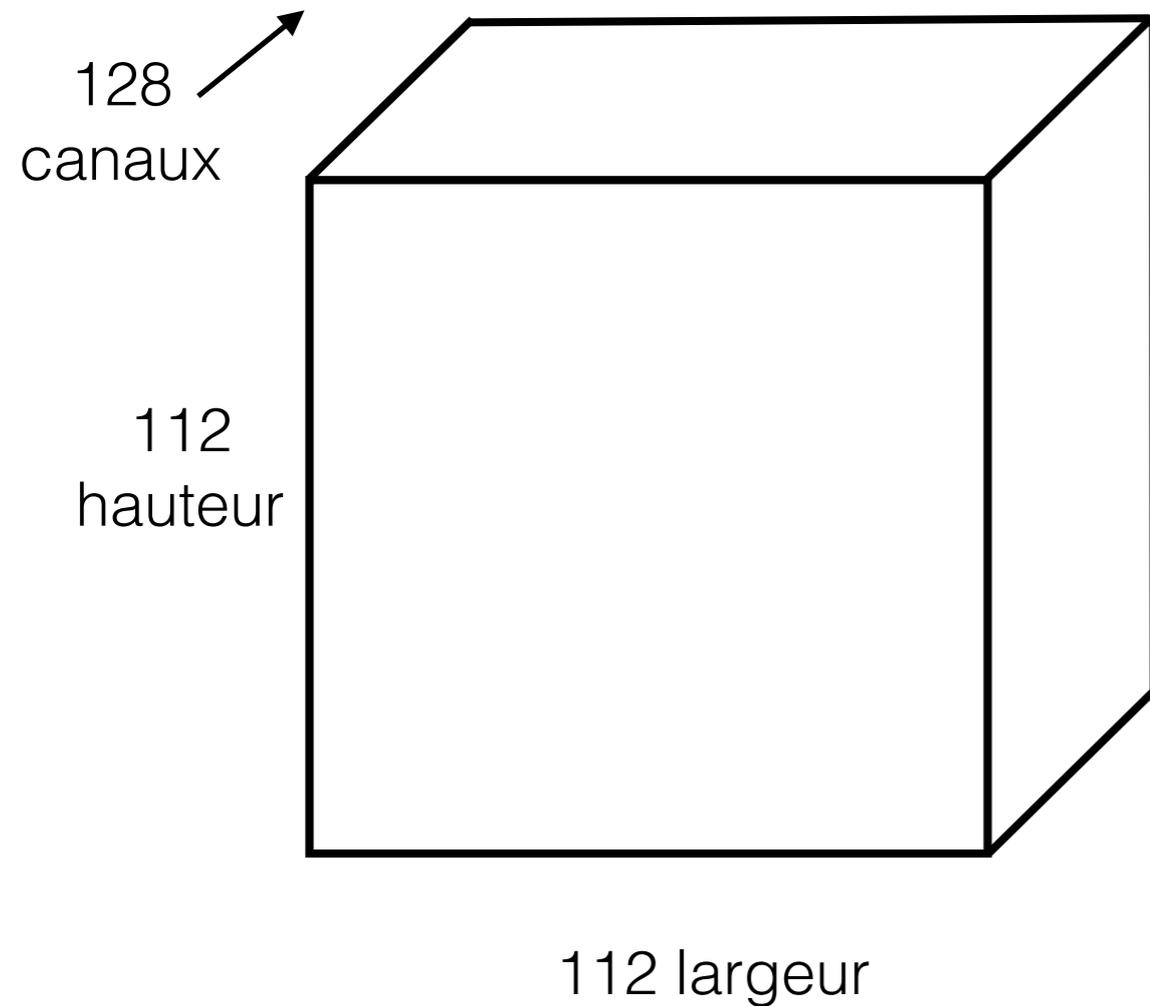


$$\left[\left[\left[I \cdot W_1 \right] \cdot W_2 \right] \cdot W_3 \right] \cdot w_{\text{«clown fish»}}$$

Extraire les caractéristiques d'une image



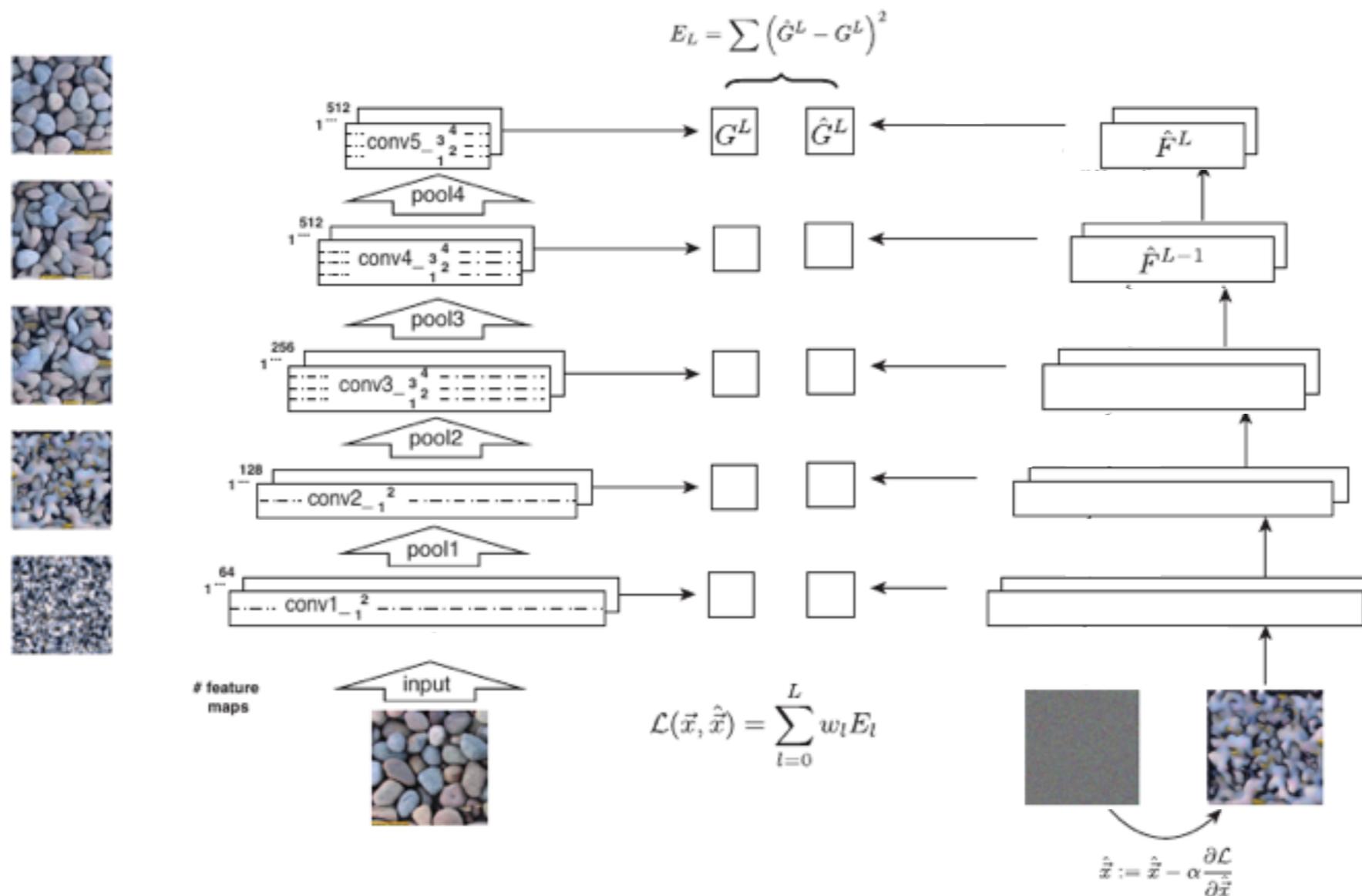
Extraire les caractéristiques d'une image



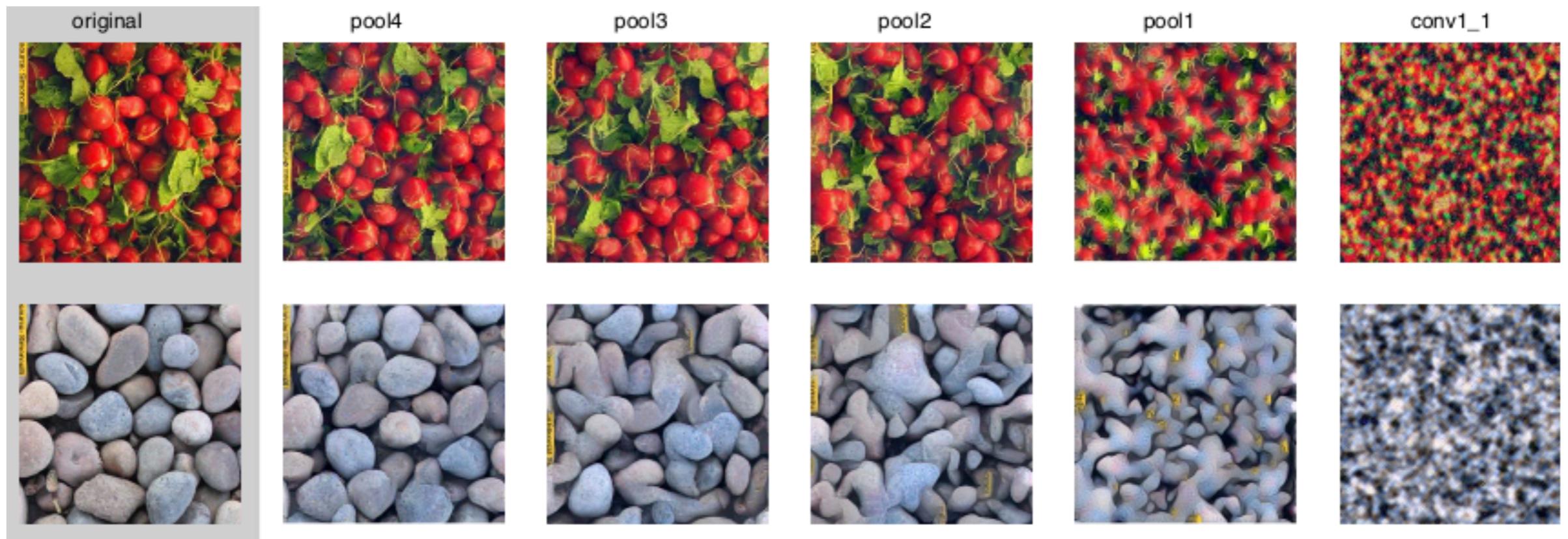
activations de «conv2»

e.g. $(112 \times 112) \times 128$ pour conv2
d'un modèle populaire (VGG19)

Synthèse de textures par CNN



Synthèse de textures par CNN



Haute → Basse

Statistiques de textures



Synthèse de textures par CNN

Échantillonner la texture originale



Synthèse de textures par CNN

Synthesis



Source

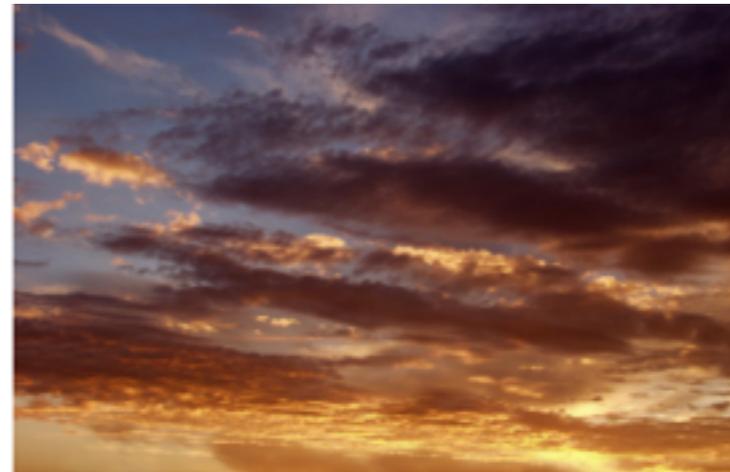
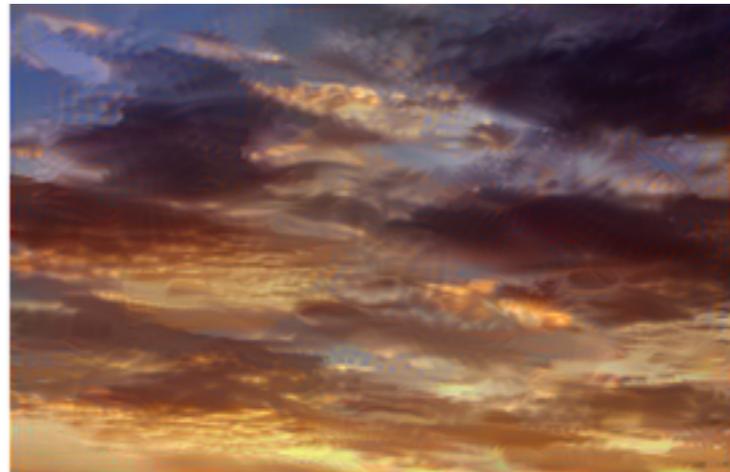


Synthèse de textures par CNN

Synthesis



Source



Capturer le style artistique

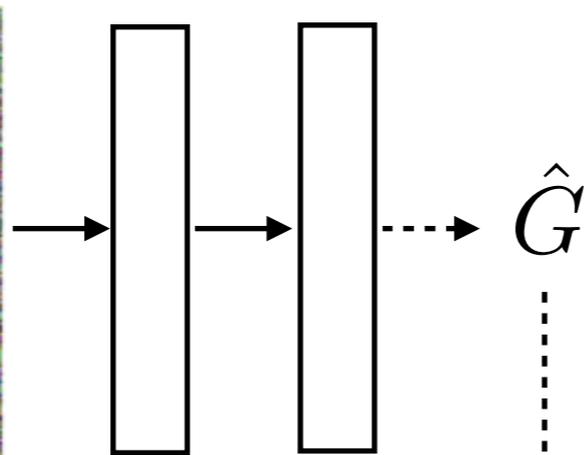
Comment transférer le style d'une peinture vers une photo?



Capturer le style artistique



Image de synthèse



\hat{G}



$$\sum_{i,j} (\hat{G}_{ij} - G_{ij})^2$$



Rajouter le contenu de la photo

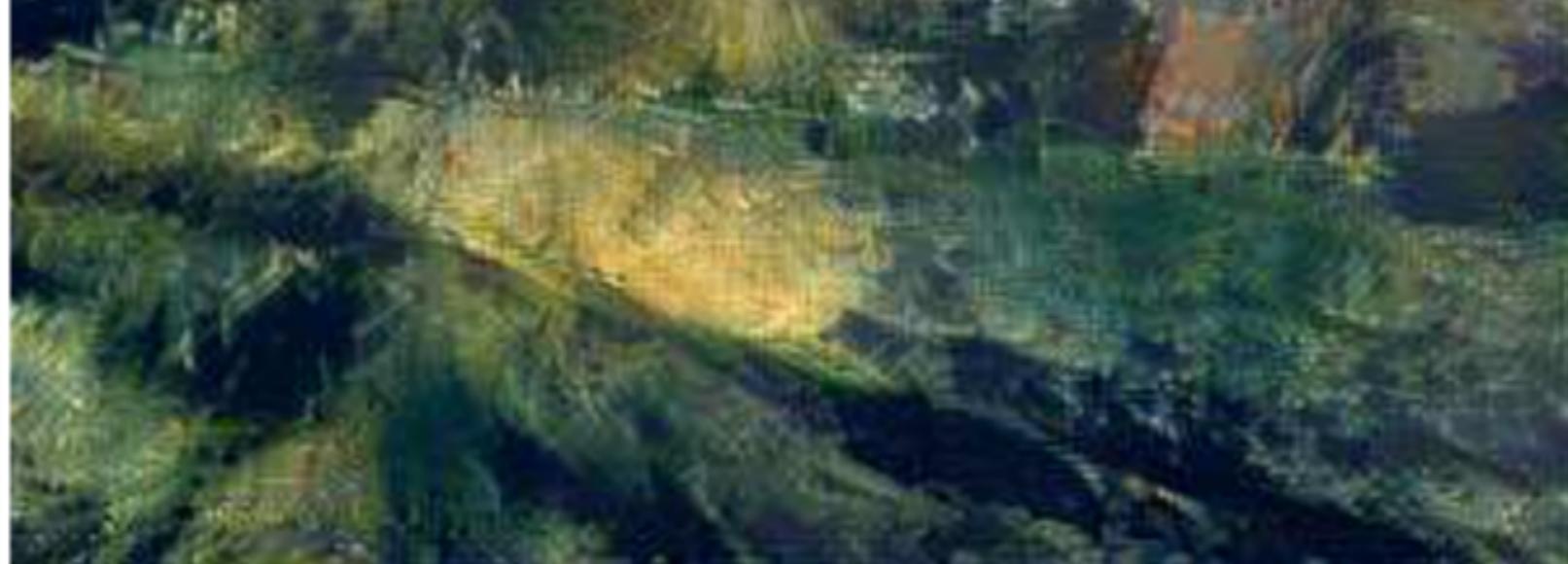
$$\sum_i \sum_{x,y} (c_i(x,y) - \hat{c}_i(x,y))^2$$

$$c_i(x,y)$$

$$\hat{c}_i(x,y)$$











Londres le jour



New York le soir

Londres le soir?

