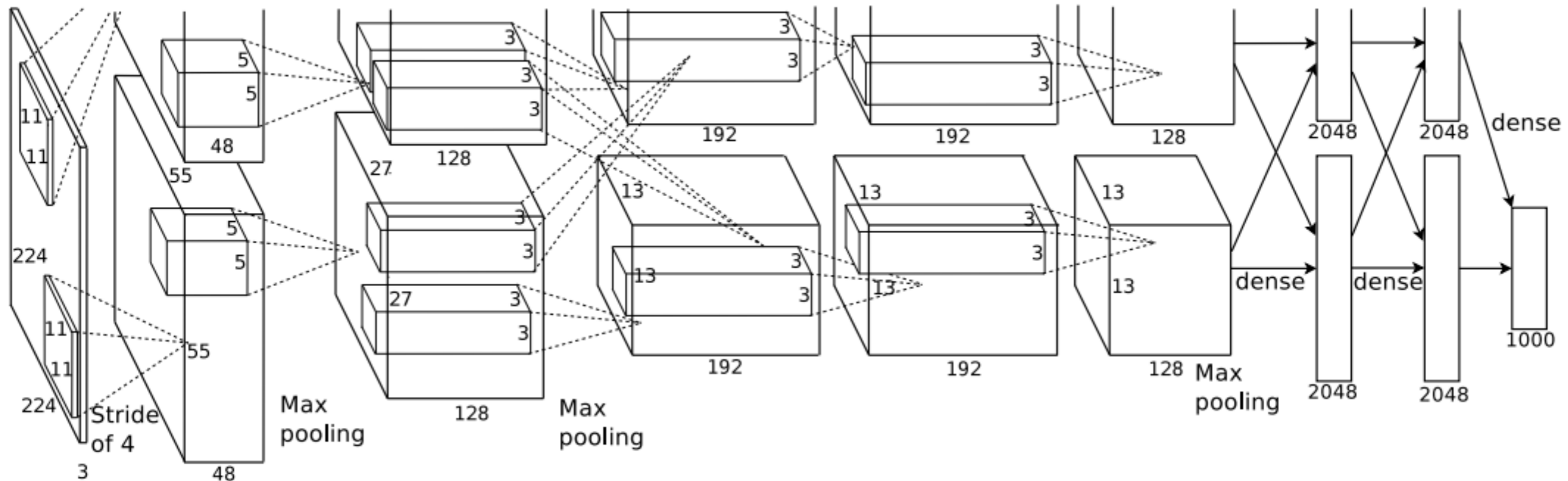


# Génération d'images par apprentissage profond



GIF-4105/7105 Photographie Algorithmique, Hiver 2019  
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

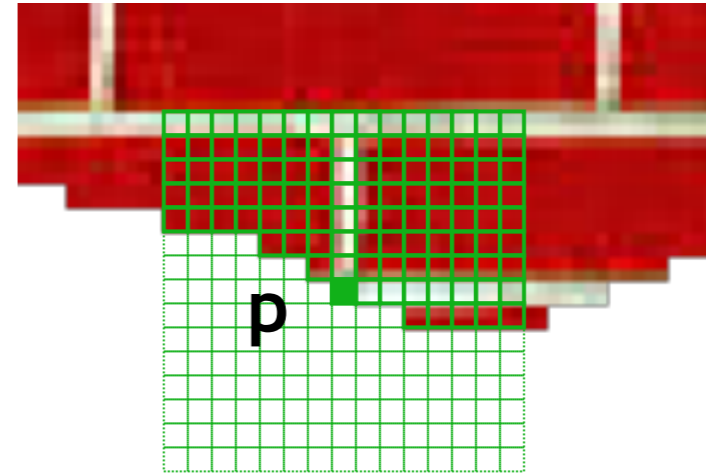
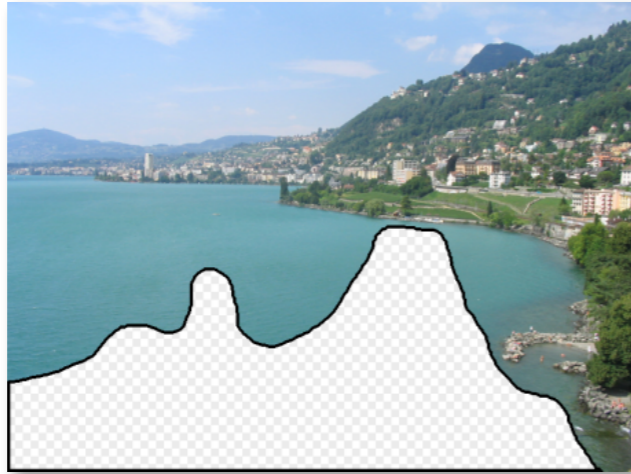
# «Digital Dark Matter»

[Perona 2010]

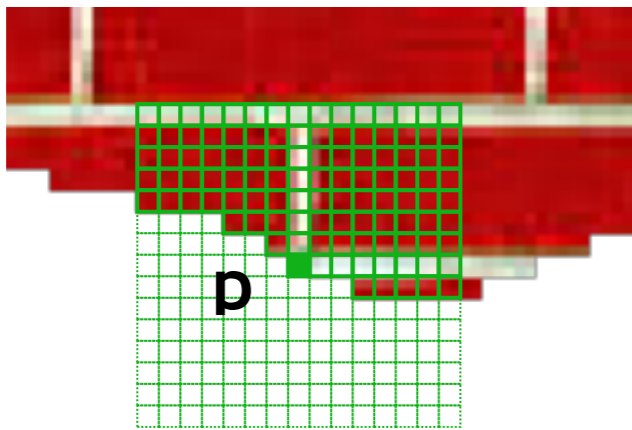
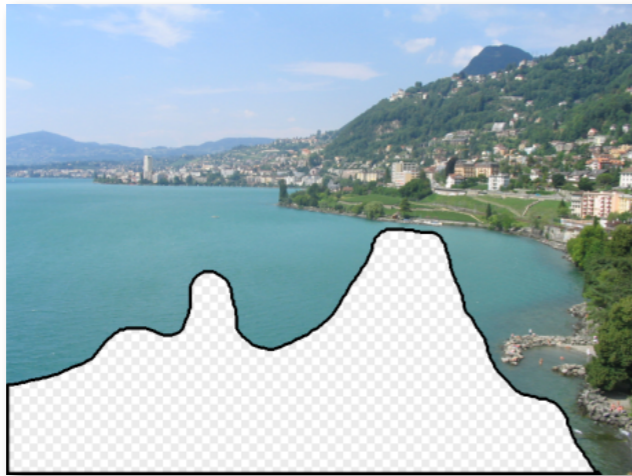
# Défi principal

- 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

# Rappel: 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»

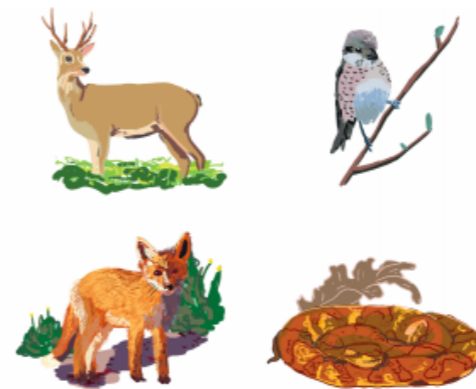


Assume que le monde  
possède une structure simple

Apprend une représentation  
qui capture cette structure



Classification units



PIT/AIT



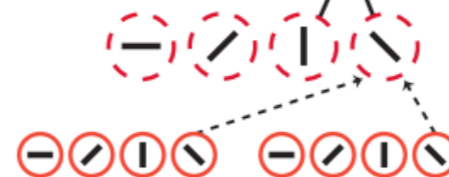
V4/PIT



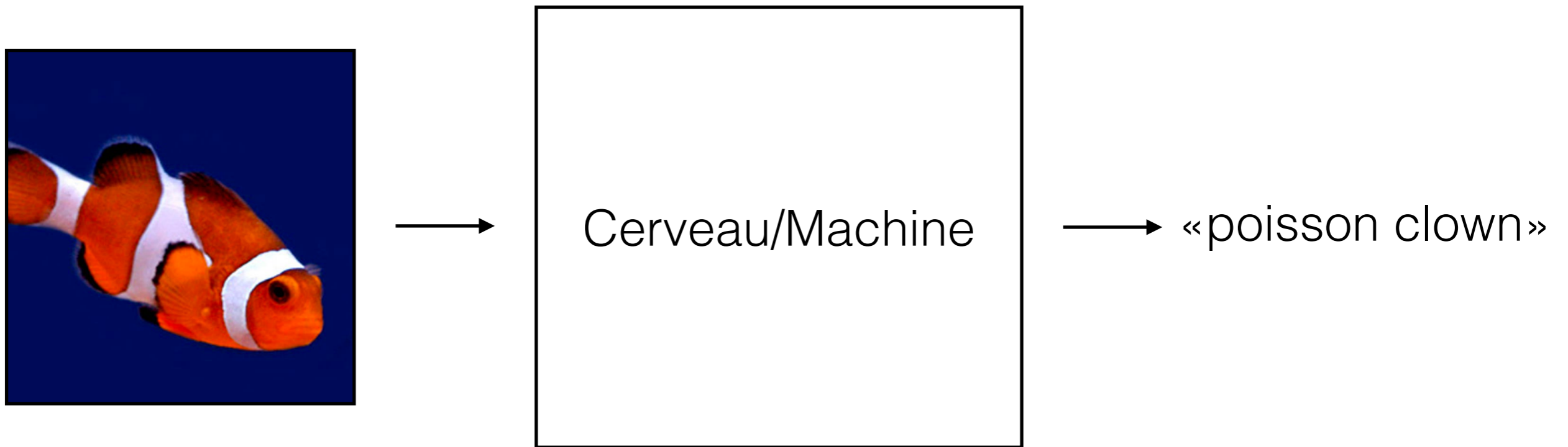
V2/V4



V1/V2



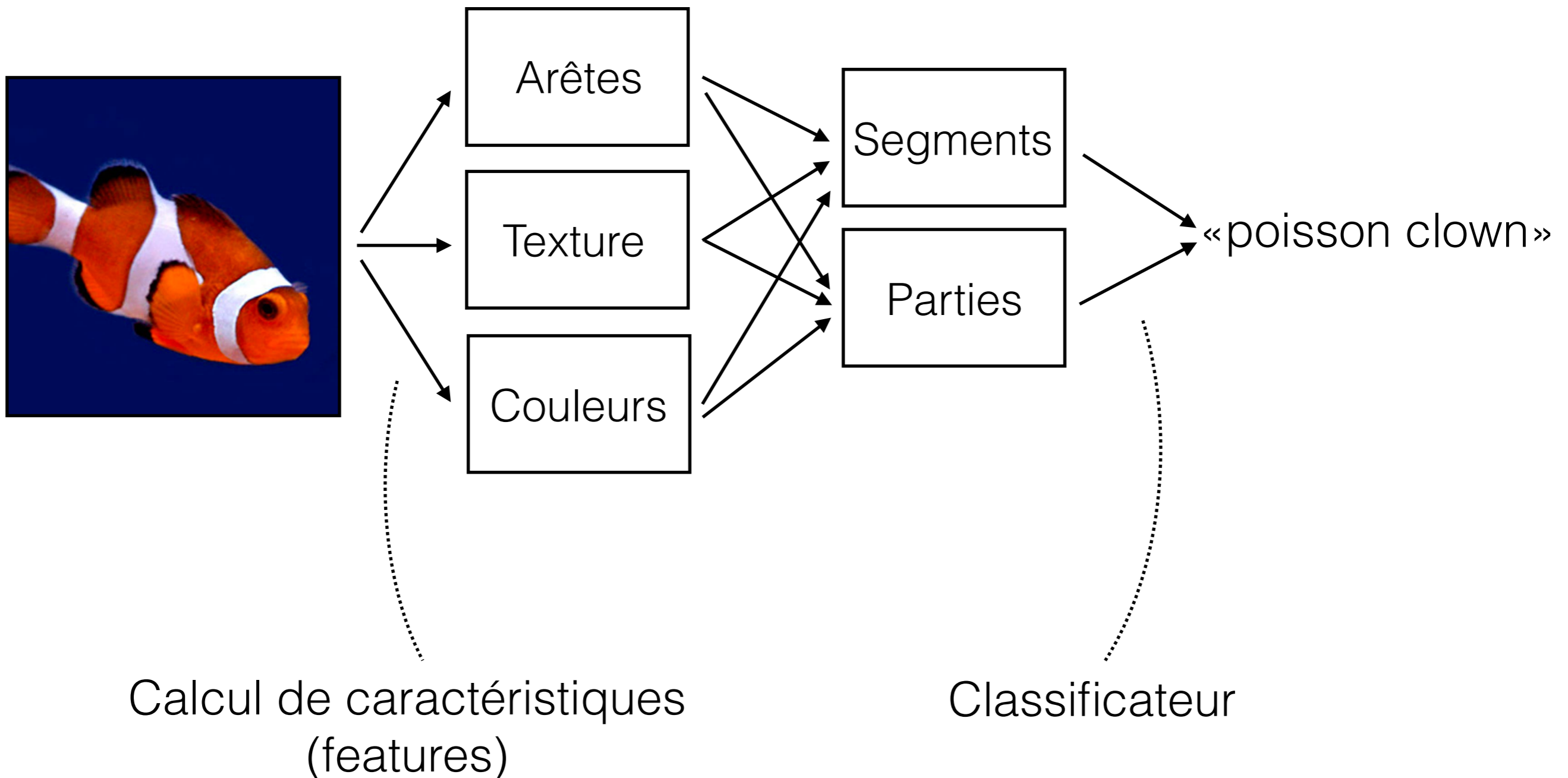
# Idée de base



Hiérarchie d'unités *simples*

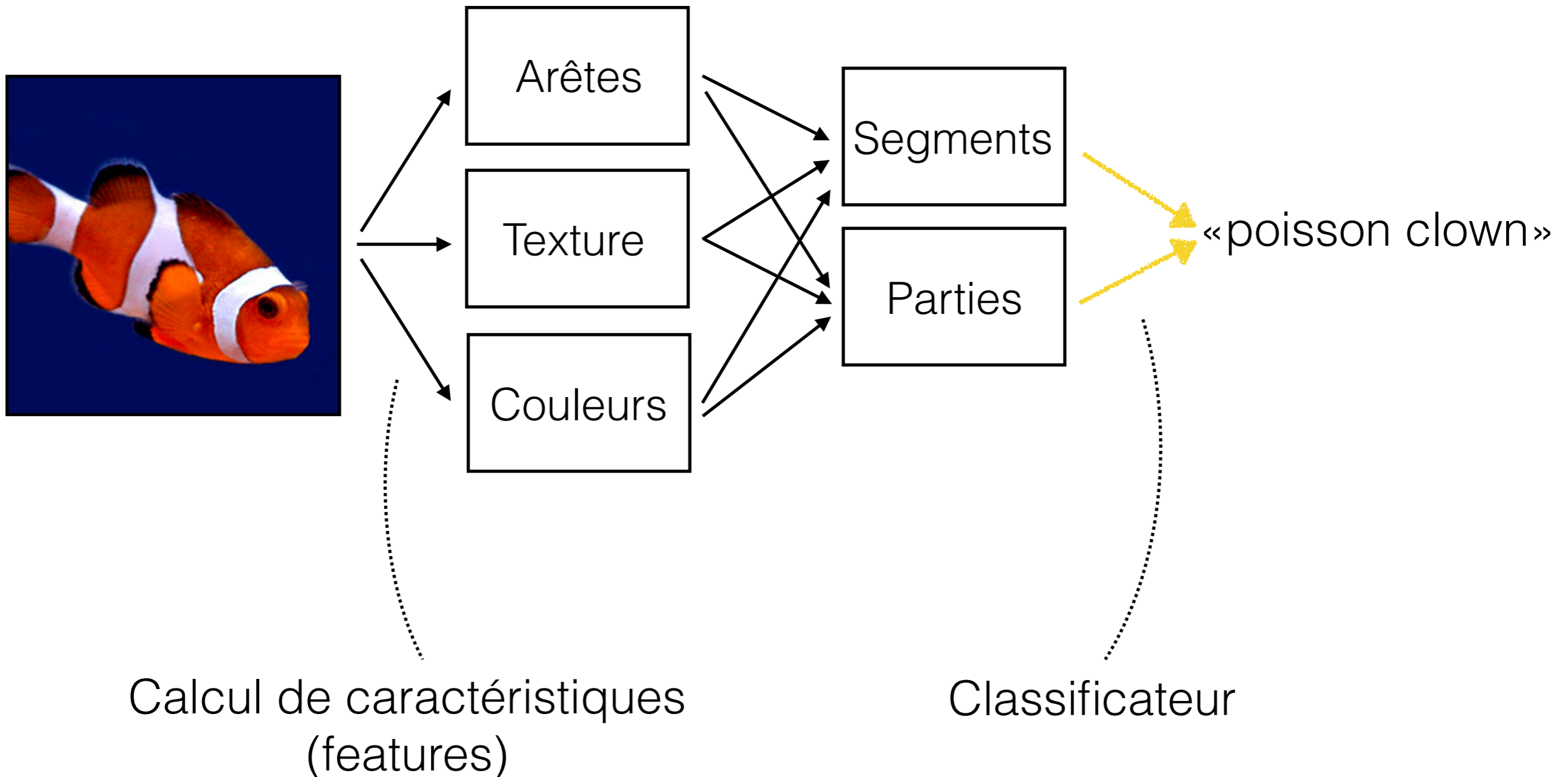


# Reconnaissance d'objets: approche «traditionnelle»



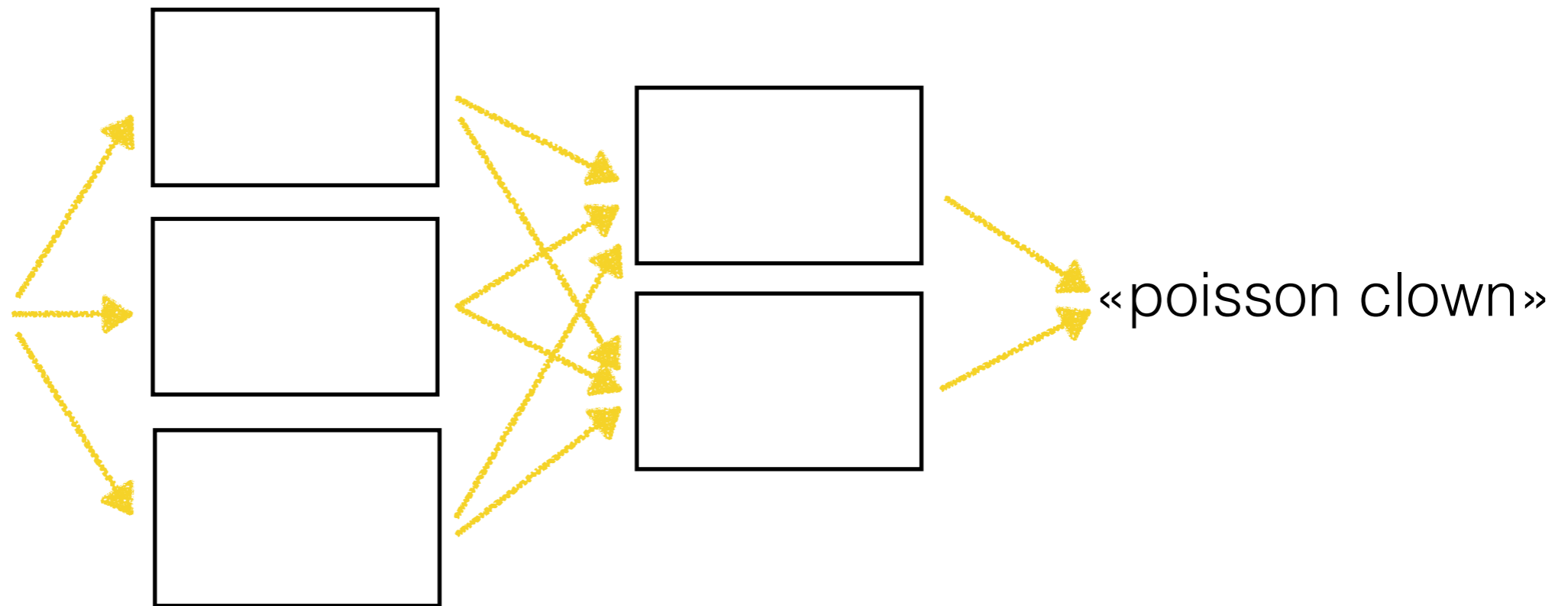
# Reconnaissance d'objets: approche «traditionnelle»

Appris automatiquement



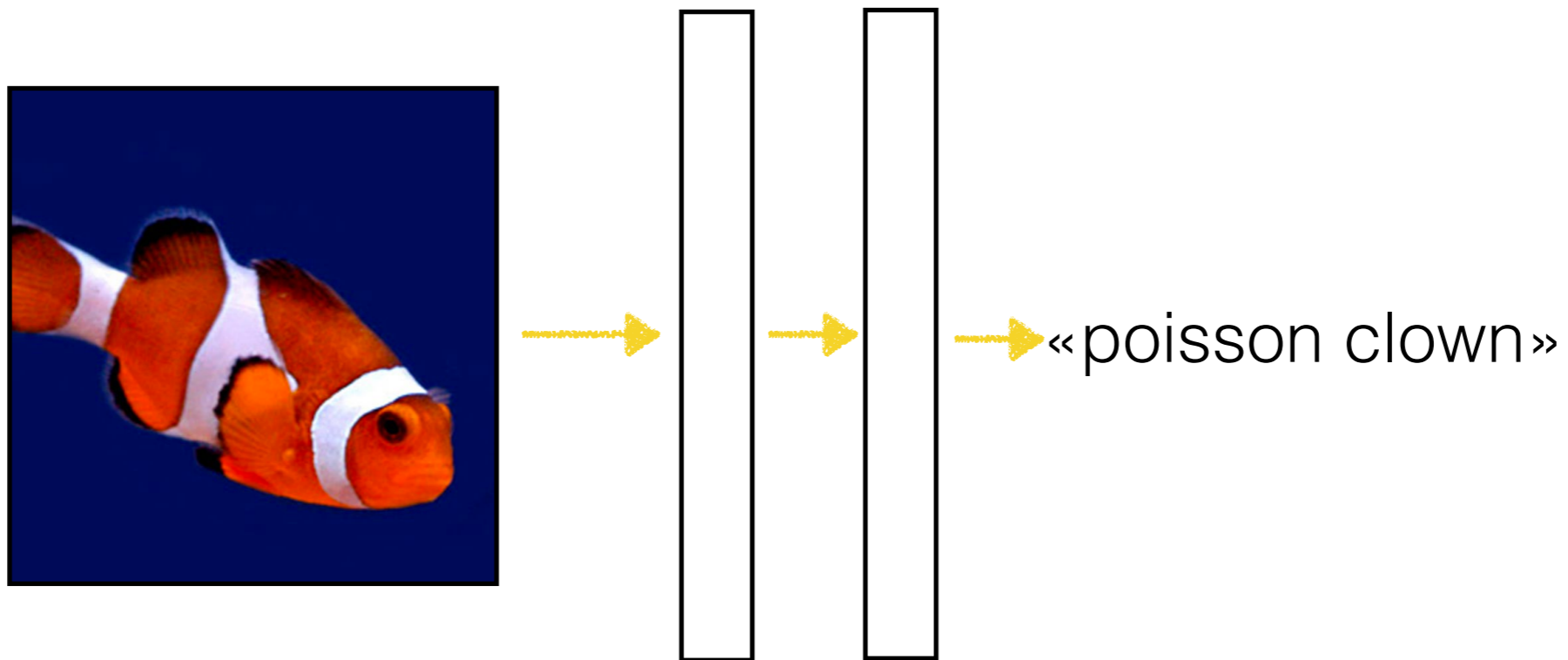
# Réseau de neurones

Appris automatiquement



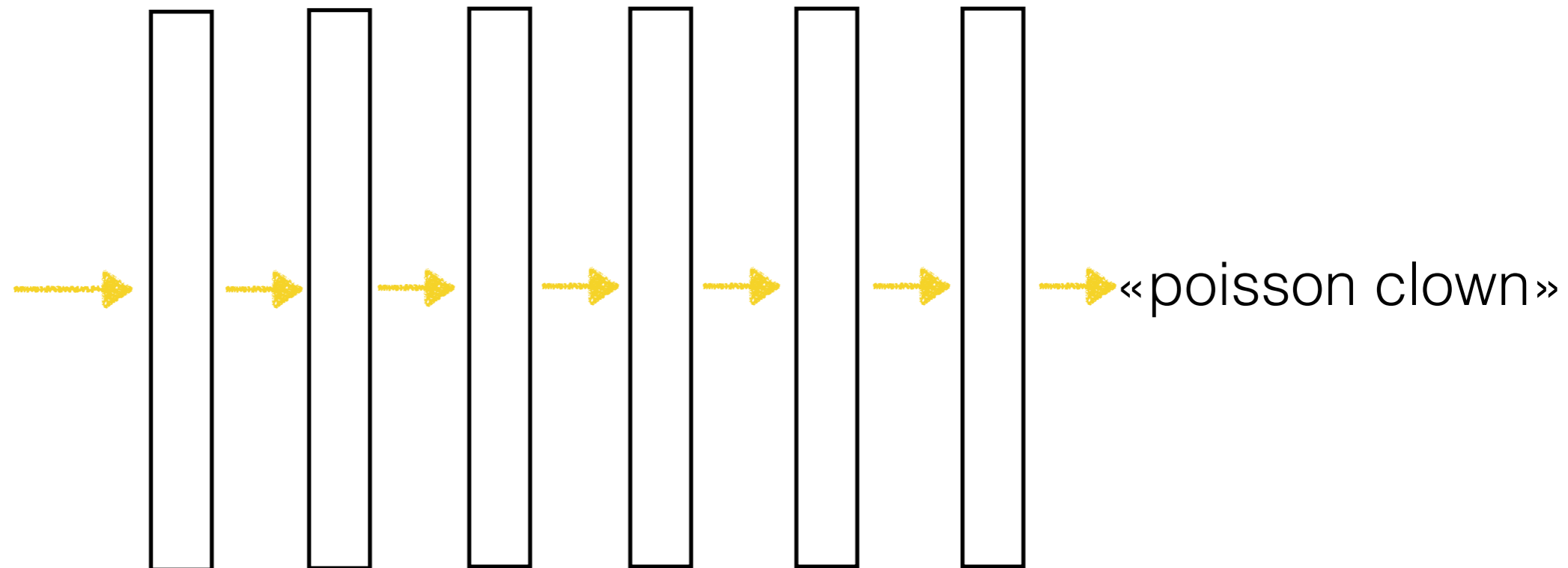
# Réseau de neurones

Appris automatiquement



# Réseau de neurones *profond*

Appris automatiquement

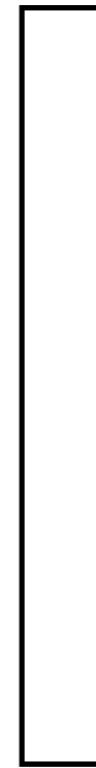


# Calculs dans un réseau de neurones

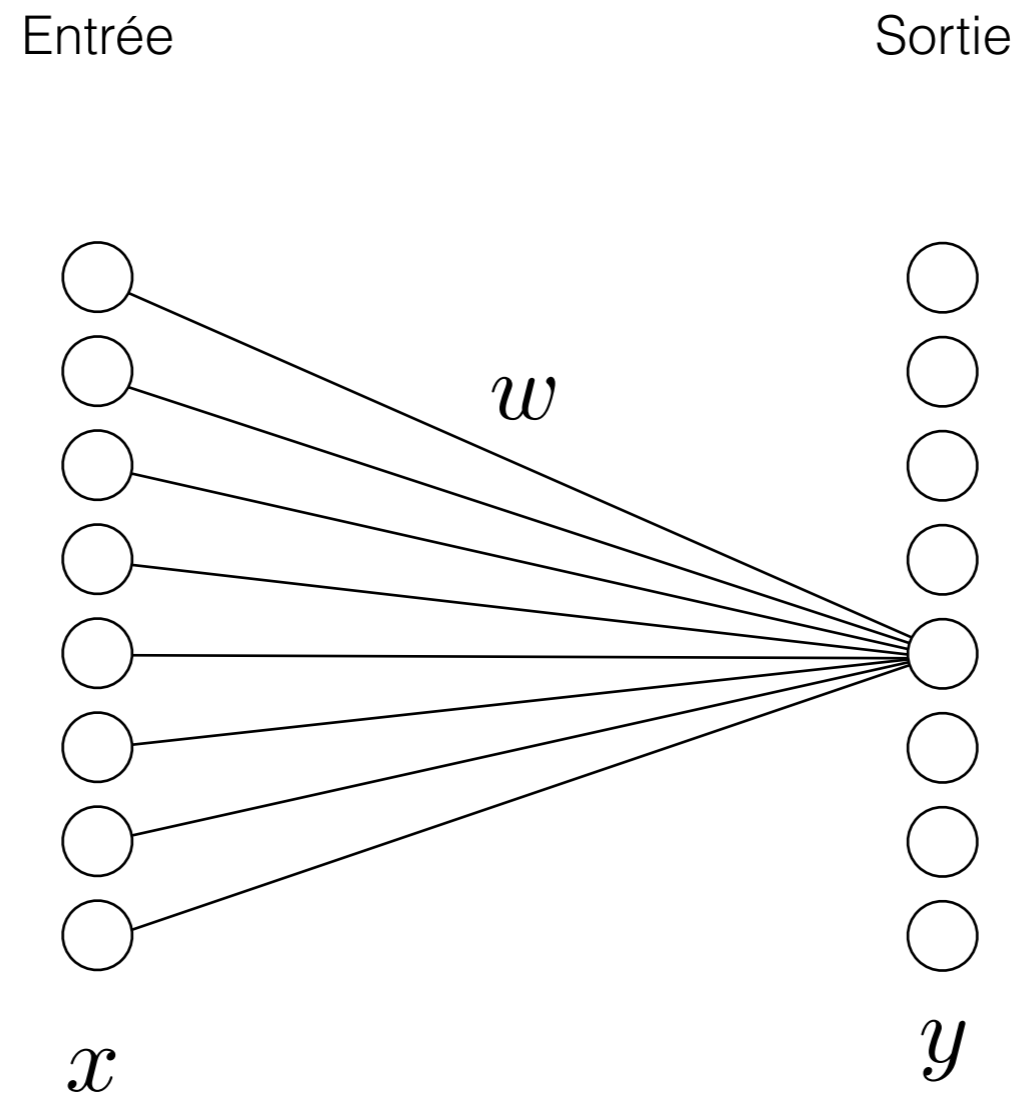
Que font ces flèches?  
Quels calculs sont effectués?

Entrée

Sortie

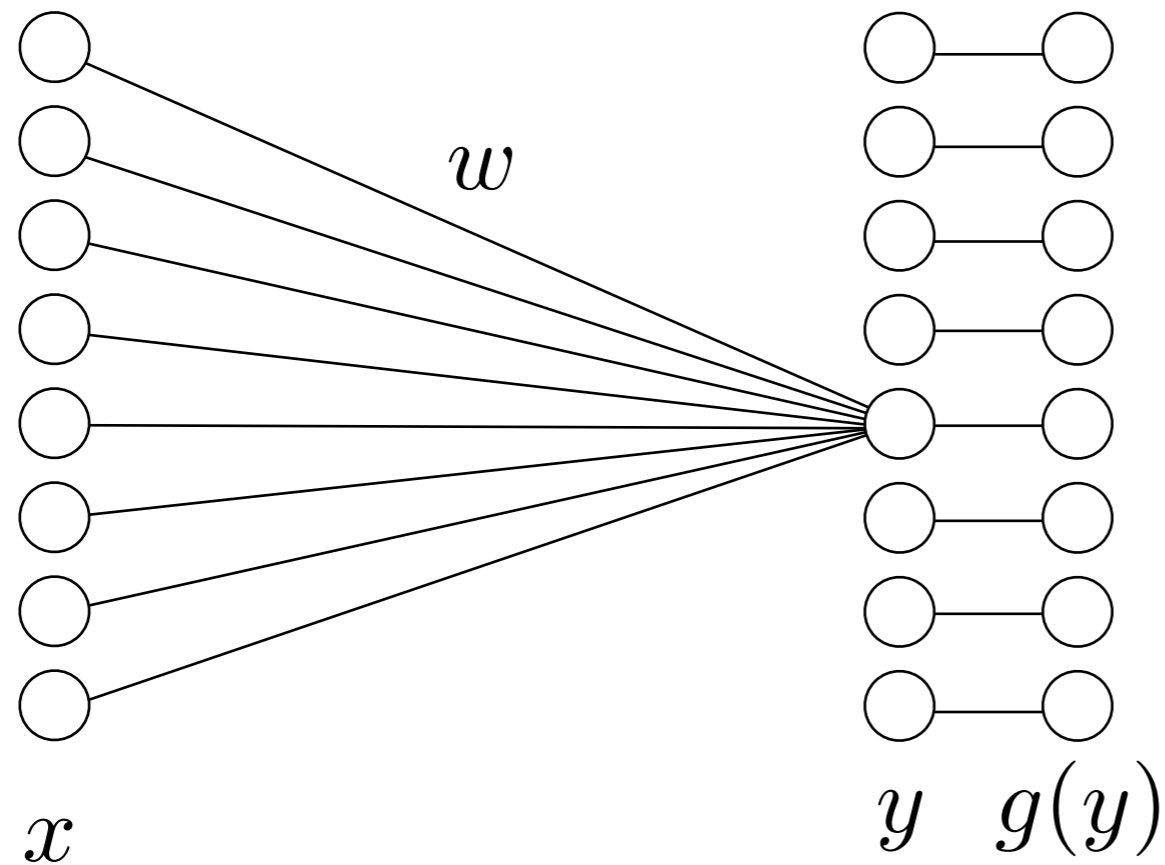


# Calculs dans un réseau de neurones

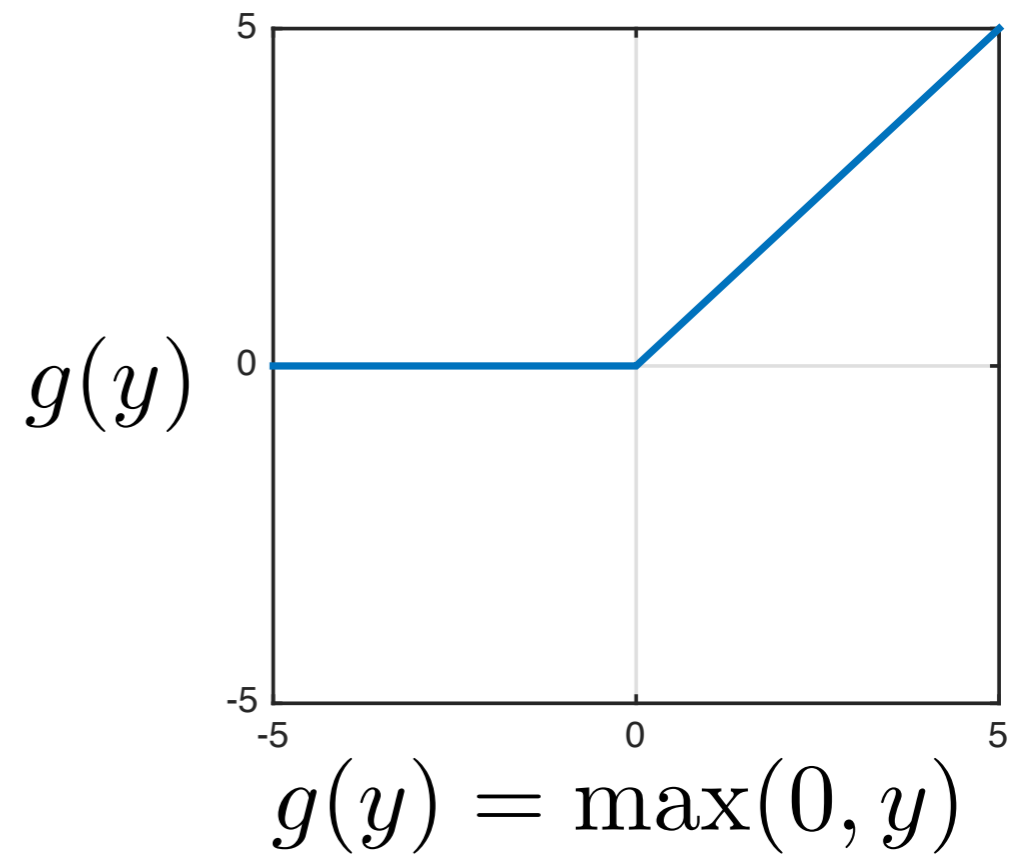


$$y_j = \sum_i w_{ij} x_i$$

# Calculs dans un réseau de neurones



«Rectified linear unit» (ReLU)

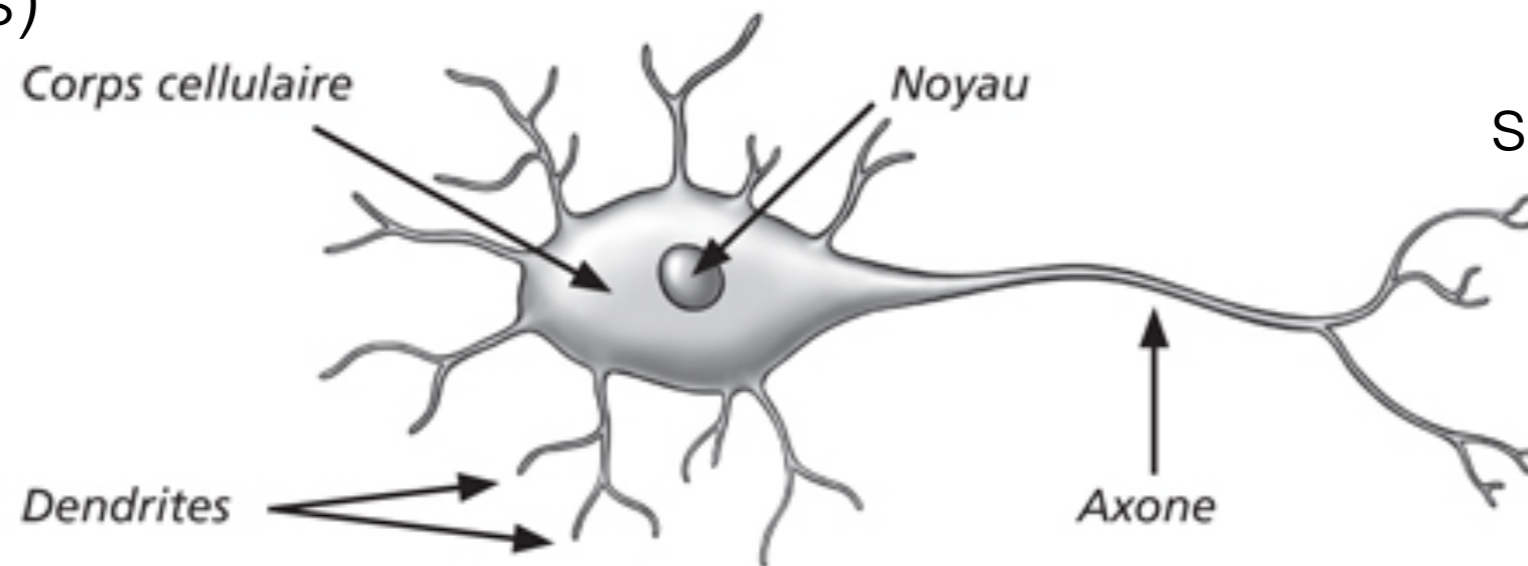




# Inspiration: neurone biologique

1. signaux provenant  
d'autres neurones  
(entrées)

$$g = f \left( \sum_i w_i x_i \right)$$



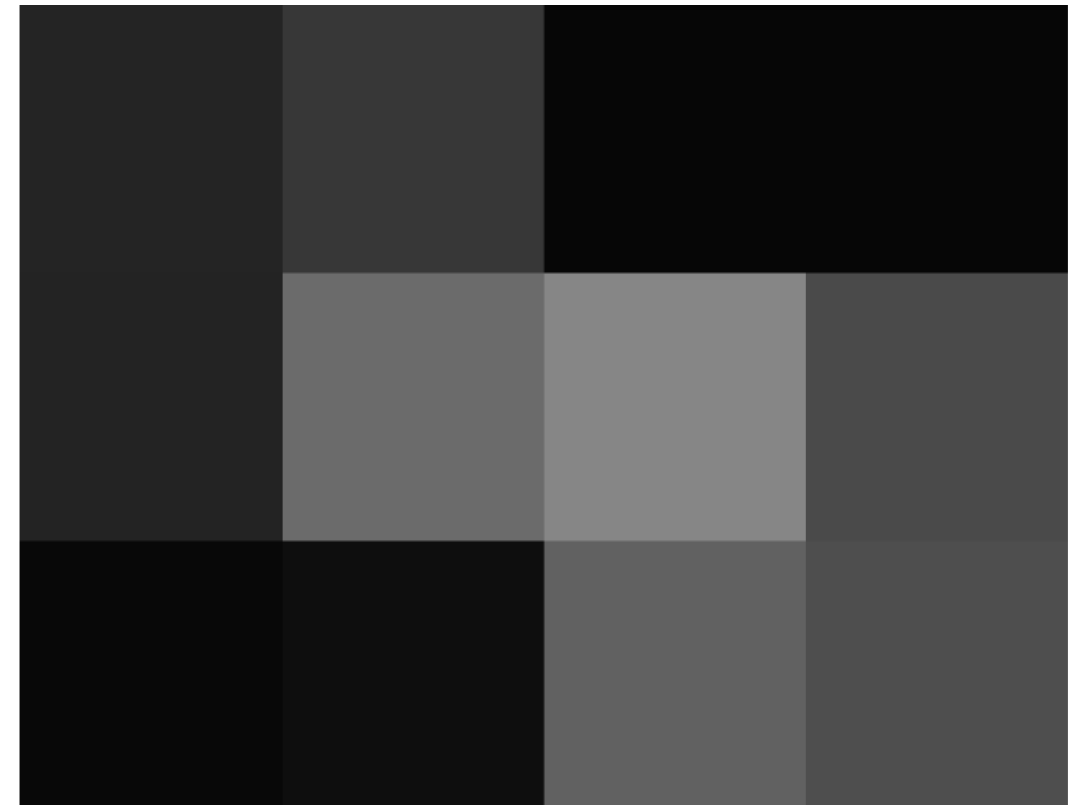
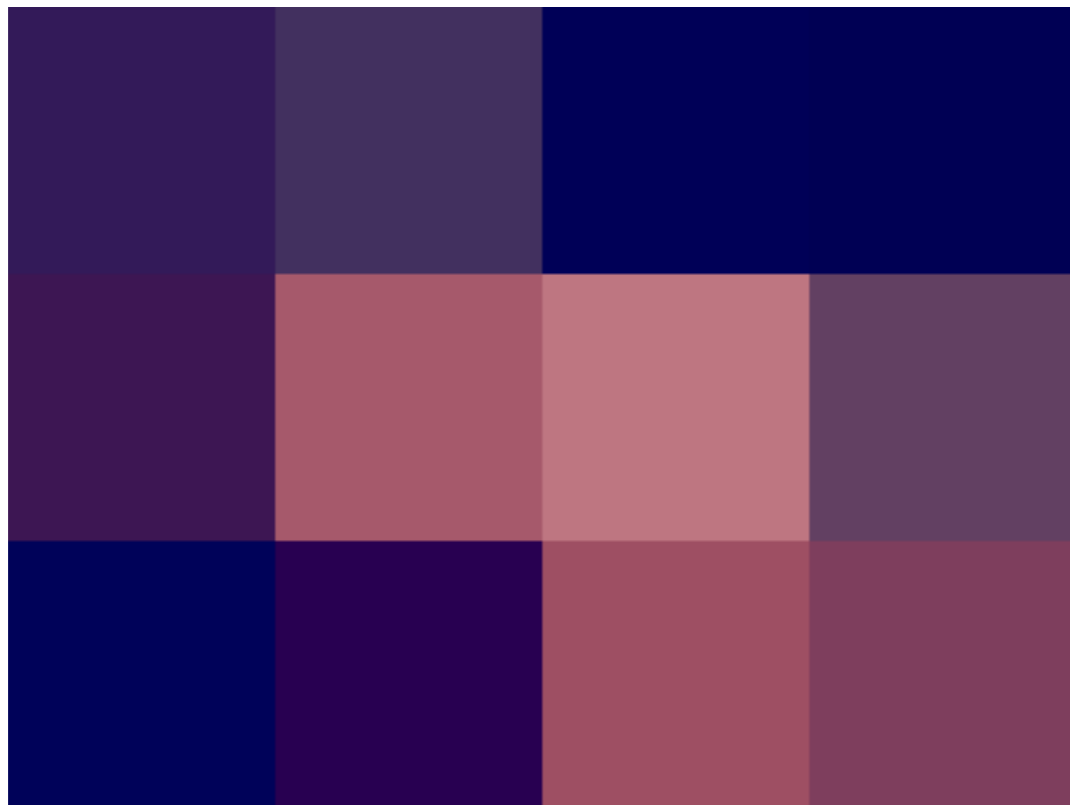
3. signal transmis  
si potentiel suffisant  
(activation)

2. accumulation (ou soustraction)  
de potentiel électrique  
(somme pondérée)

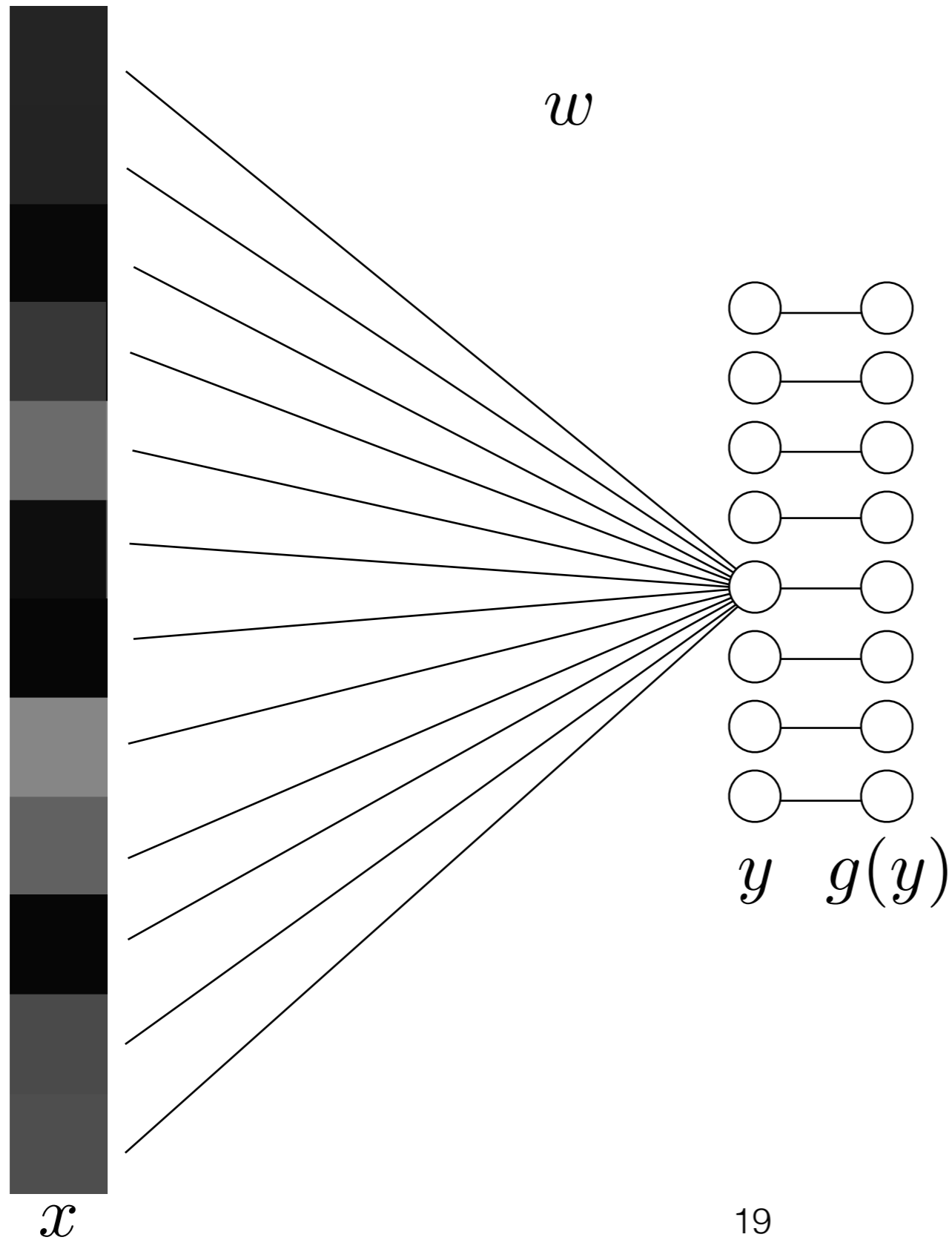
Image: Éditions Thierry Souccar

# Comment traiter une image?

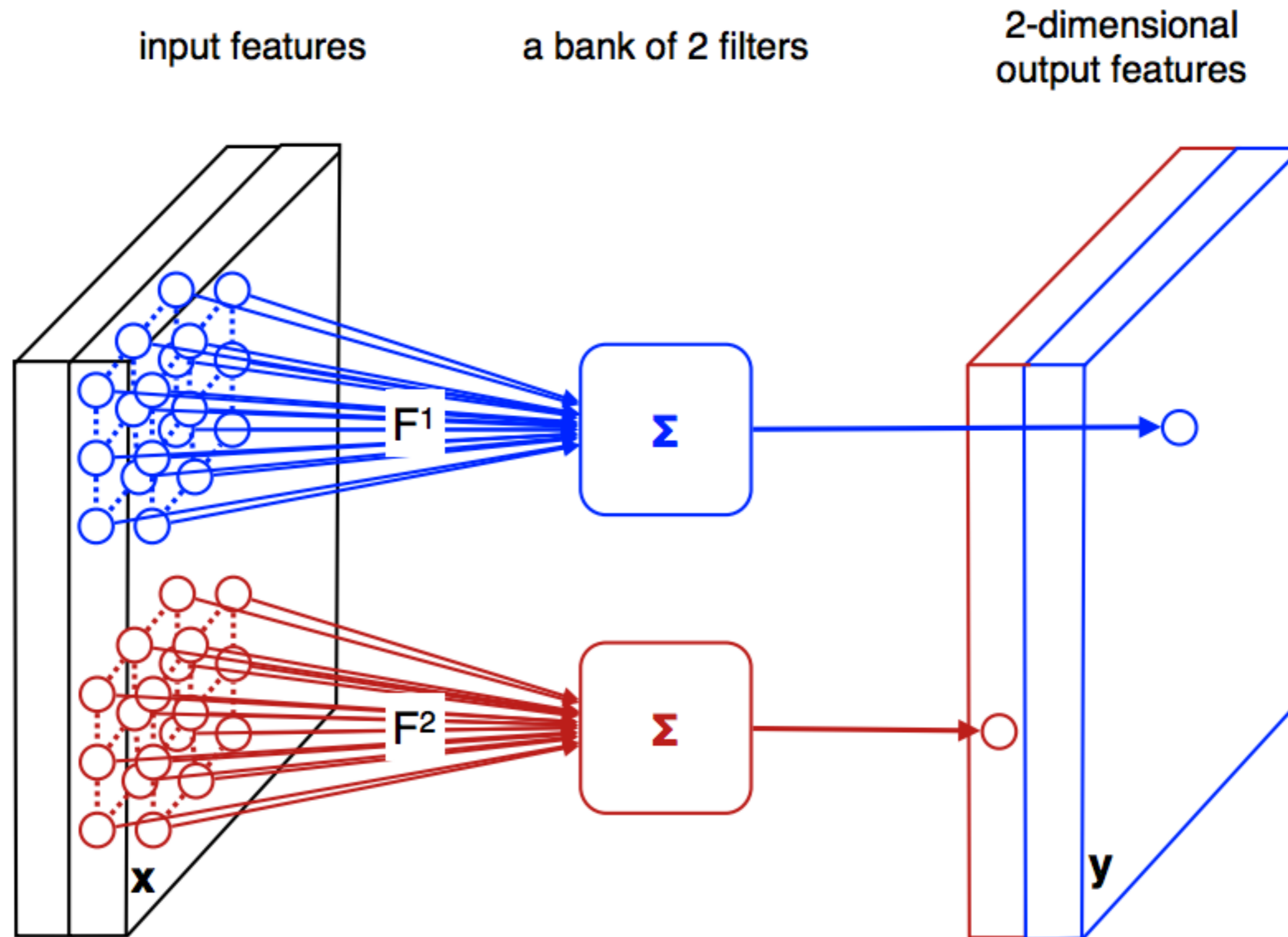
Ex: résolution 3x4



# Comment traiter une image?



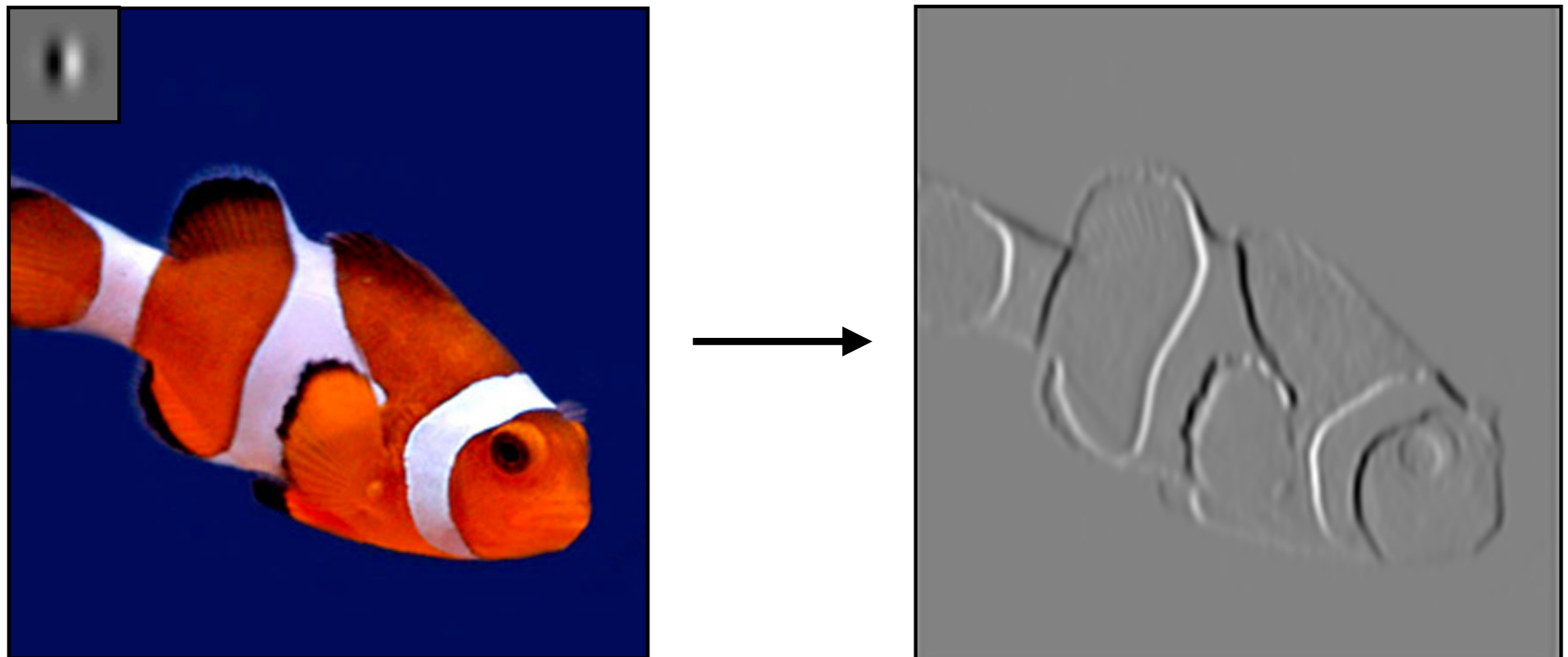
# Réseaux de neurones à *convolution* (CNN)



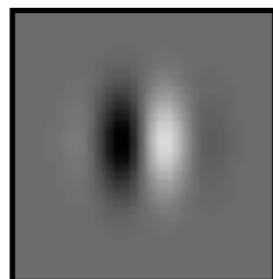
# Réseaux de neurones à convolution (CNN)

## Convolution

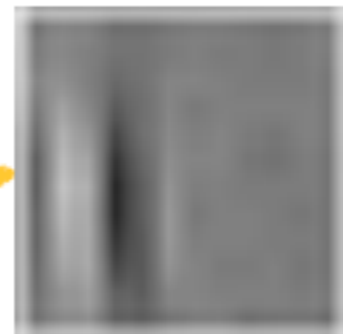
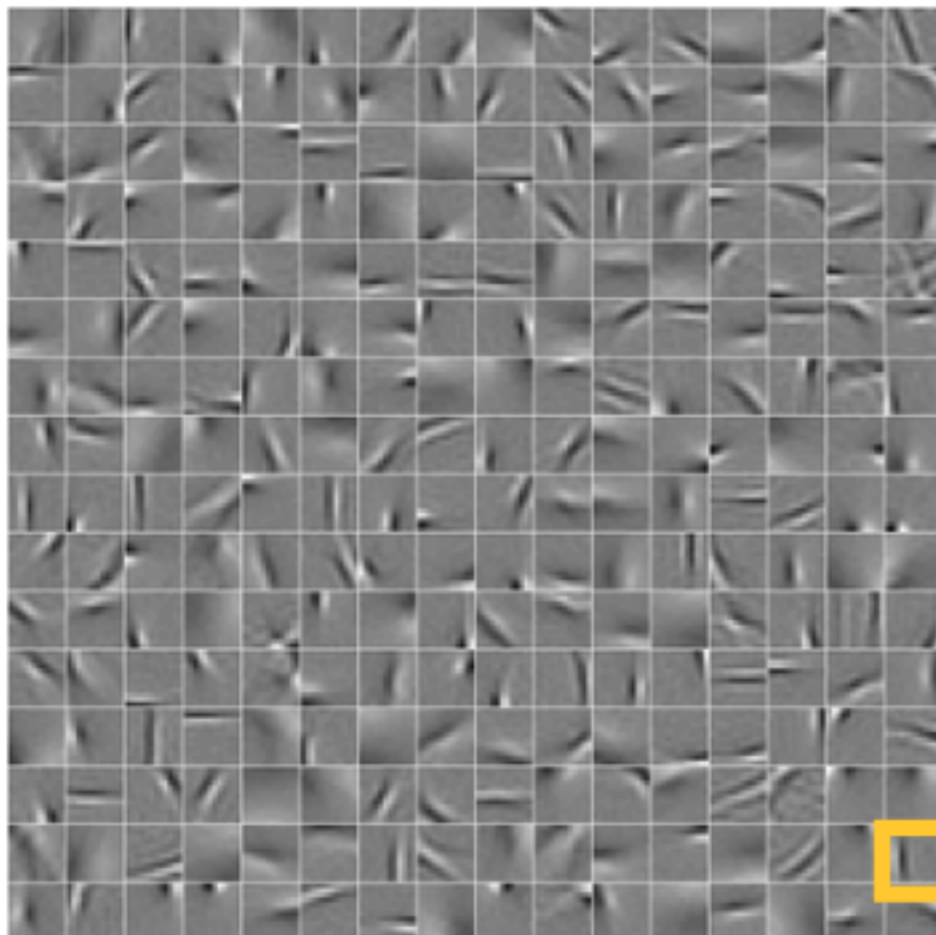
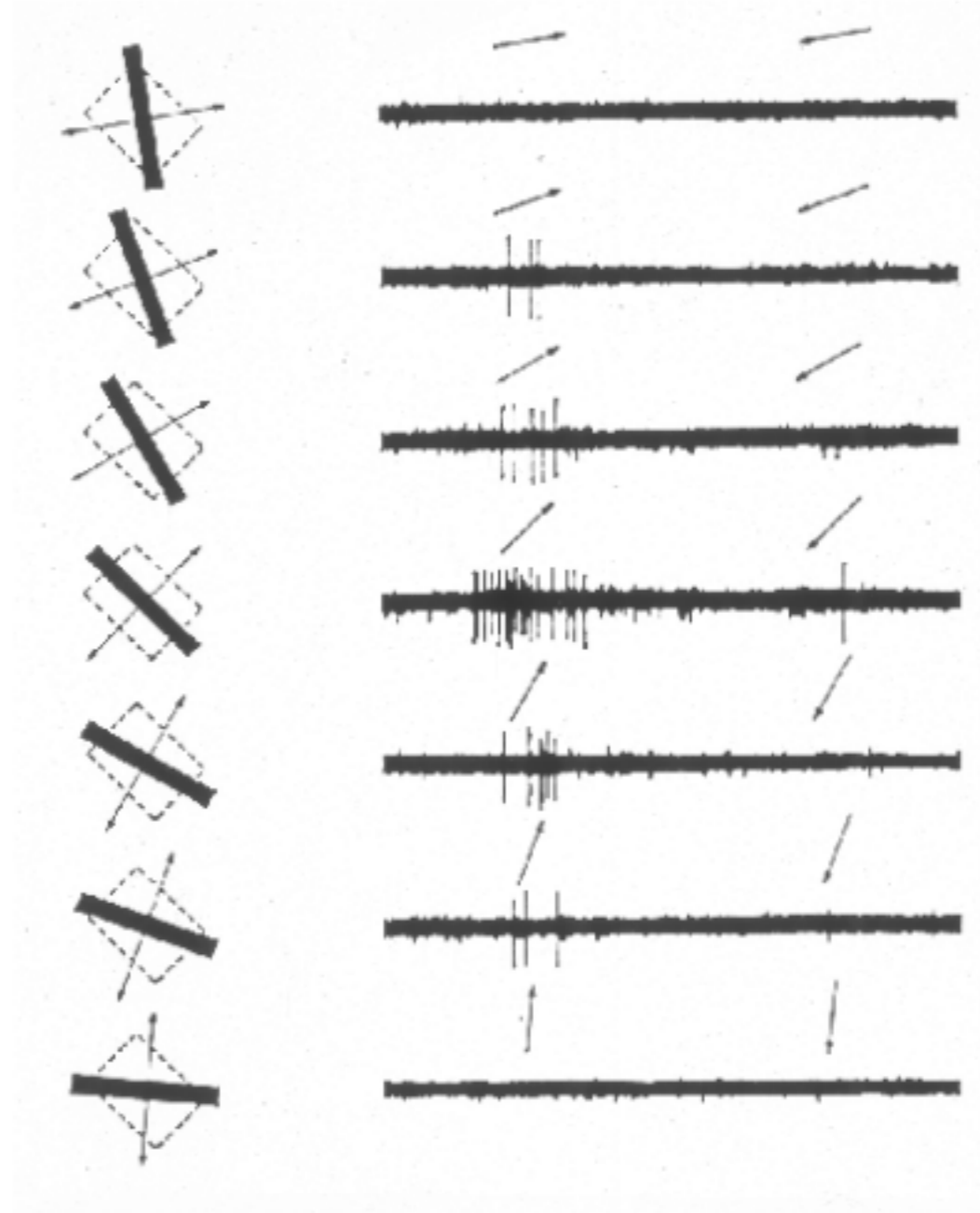
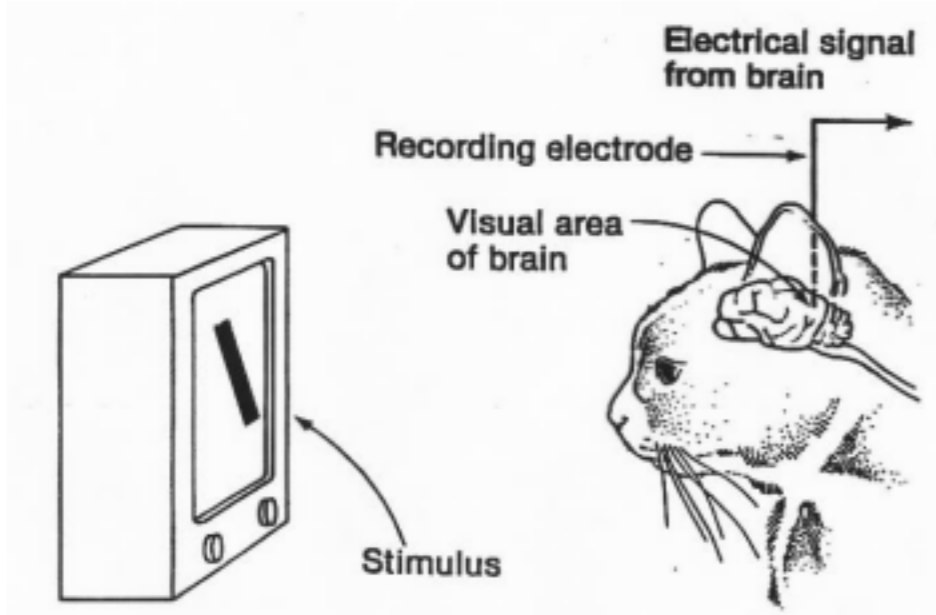
Avantage: les paramètres sont *partagés*  
(toutes les fenêtres de l'image sont traitées de la même façon)



filtre

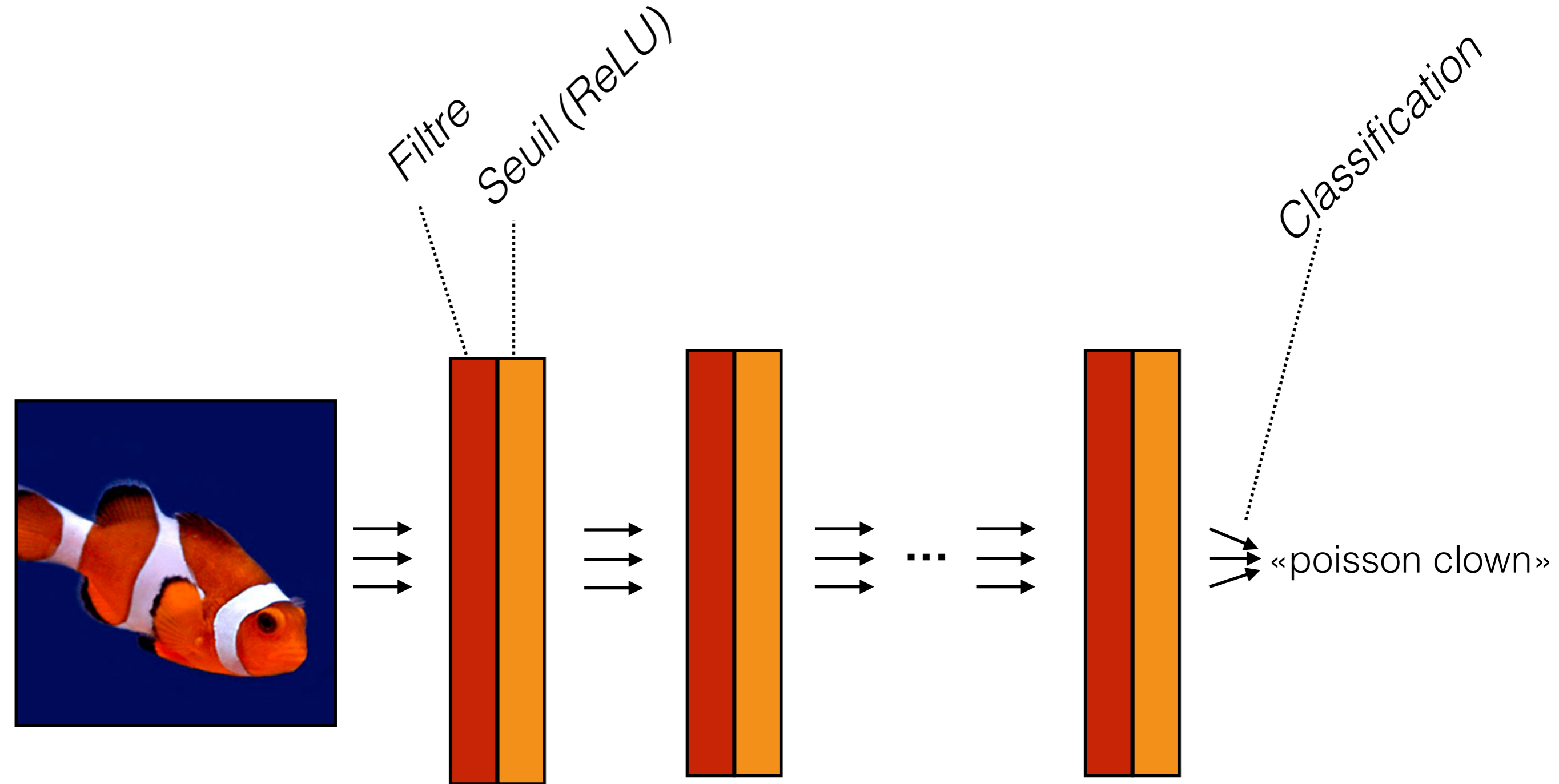


# [Hubel and Wiesel 59]



oriented filter

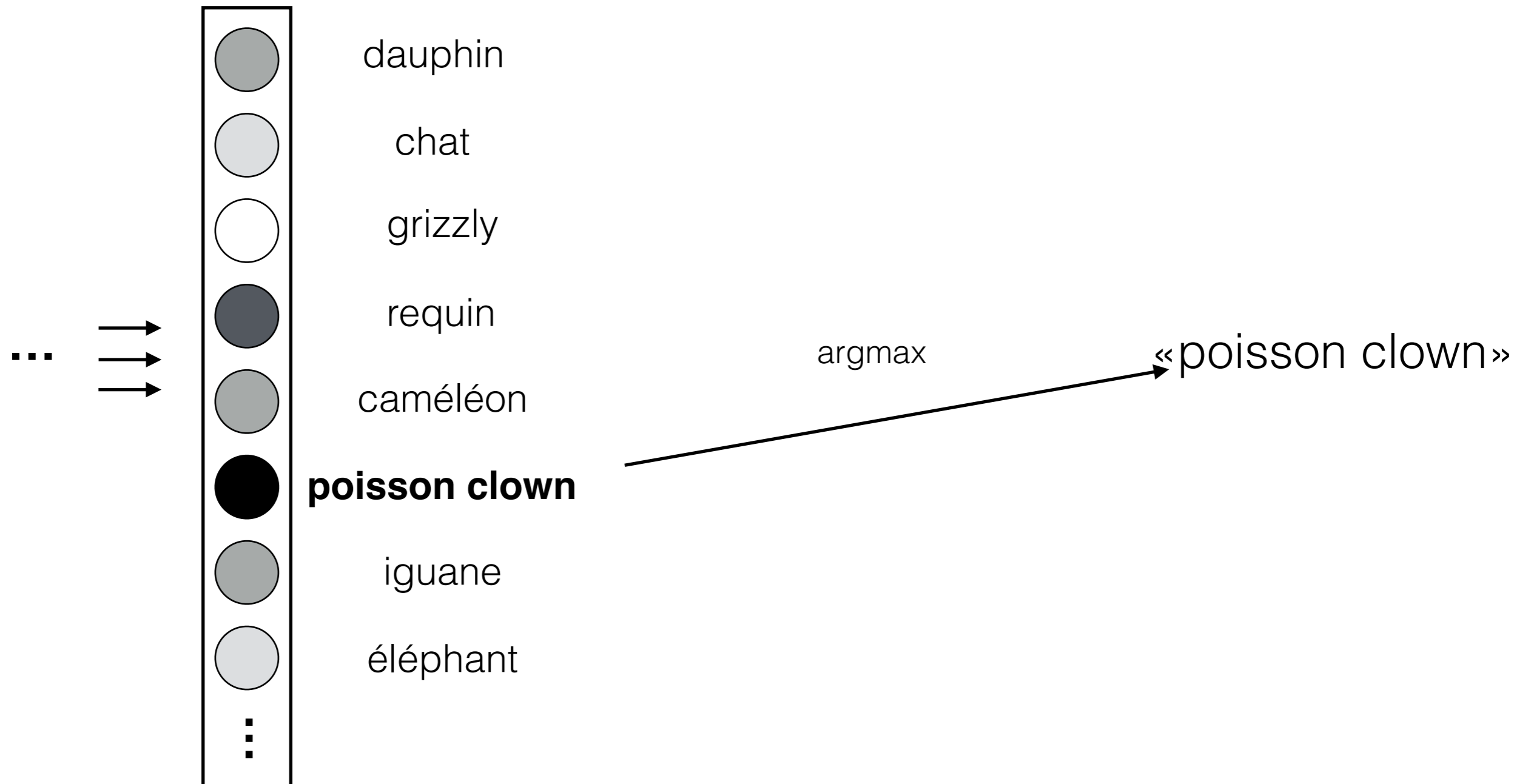
# Calculs dans un réseau de neurones



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

# Calculs dans un réseau de neurones

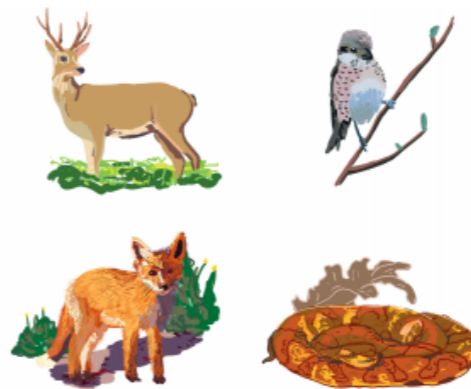
Dernière couche  
(classification)







Classification  
units



PIT/AIT



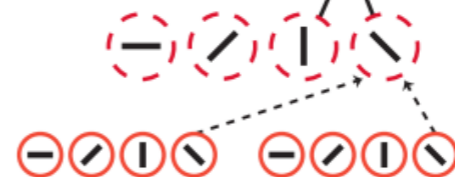
V4/PIT



V2/V4

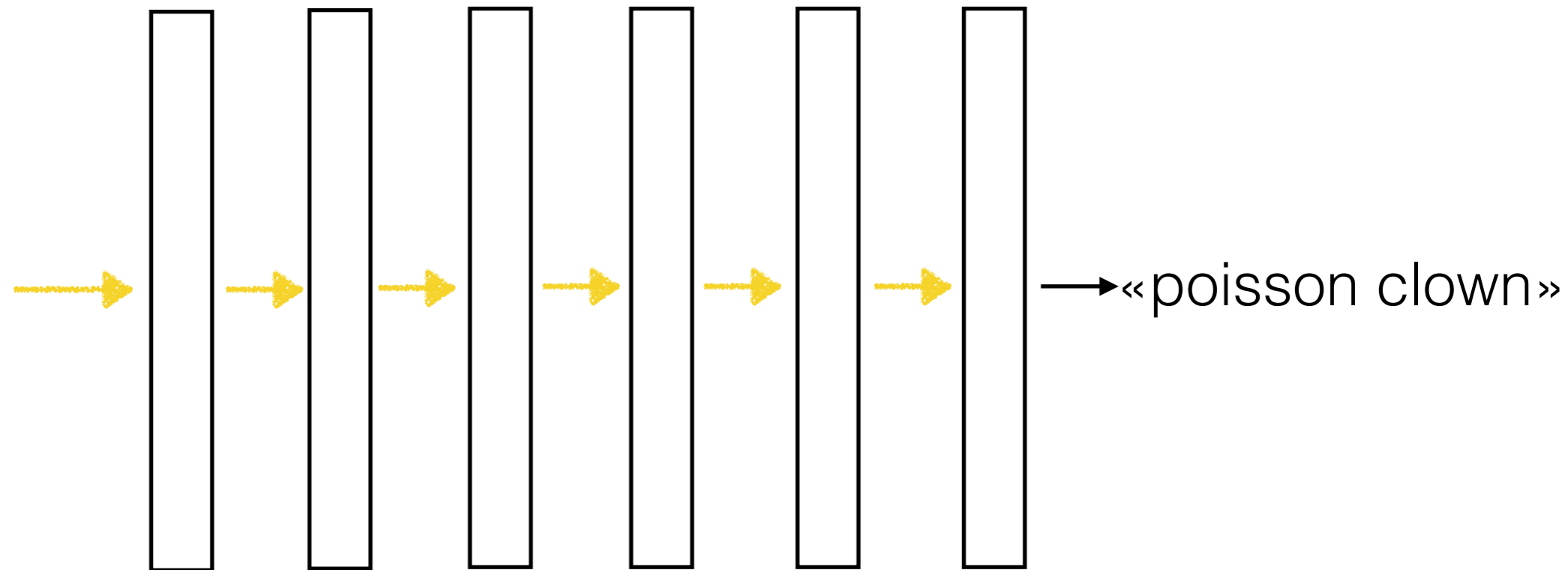


V1/V2



# 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»



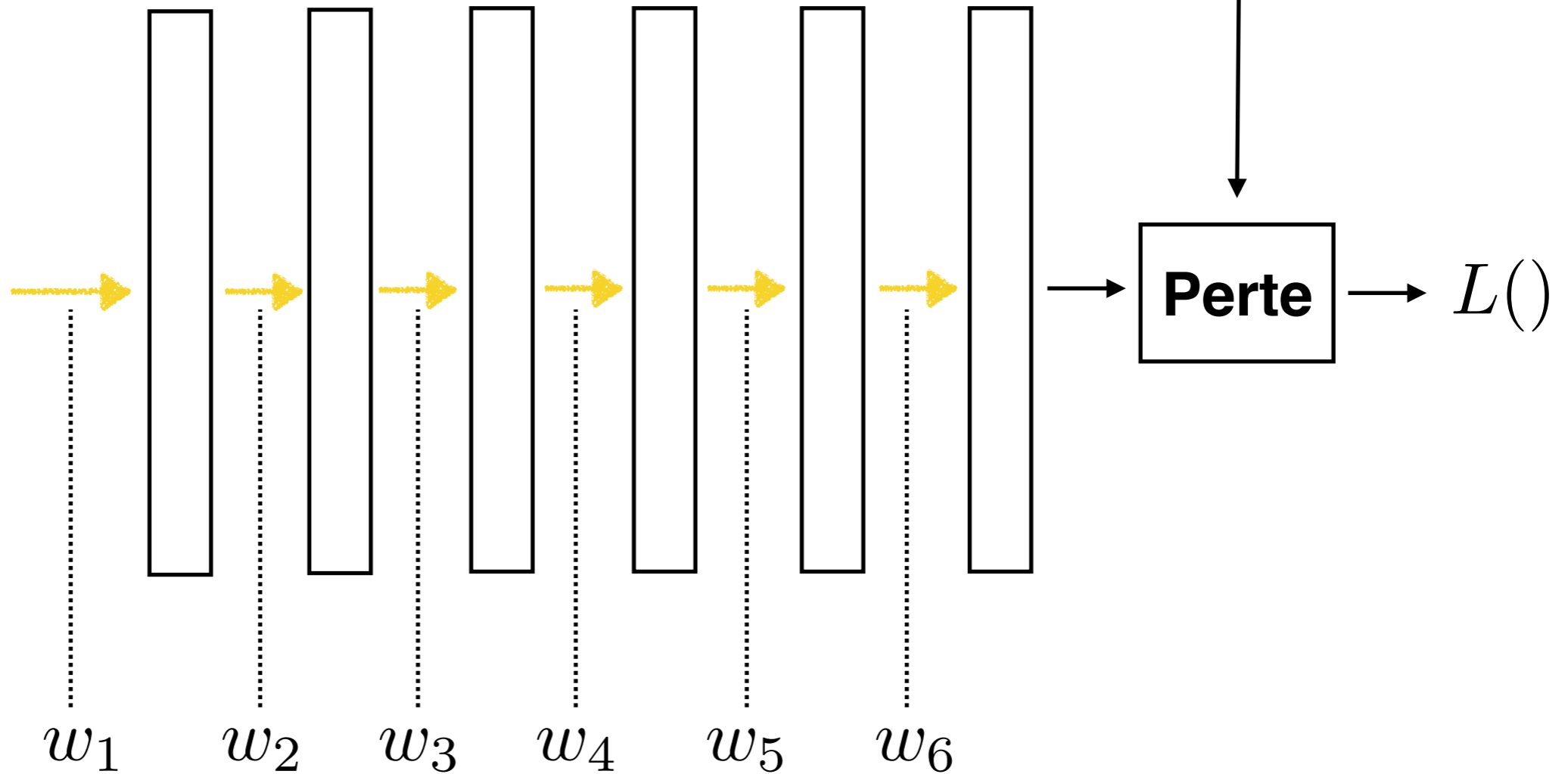
→ «caméléon»

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

# Apprentissage par réseaux profonds

Appris automatiquement

«poisson clown»

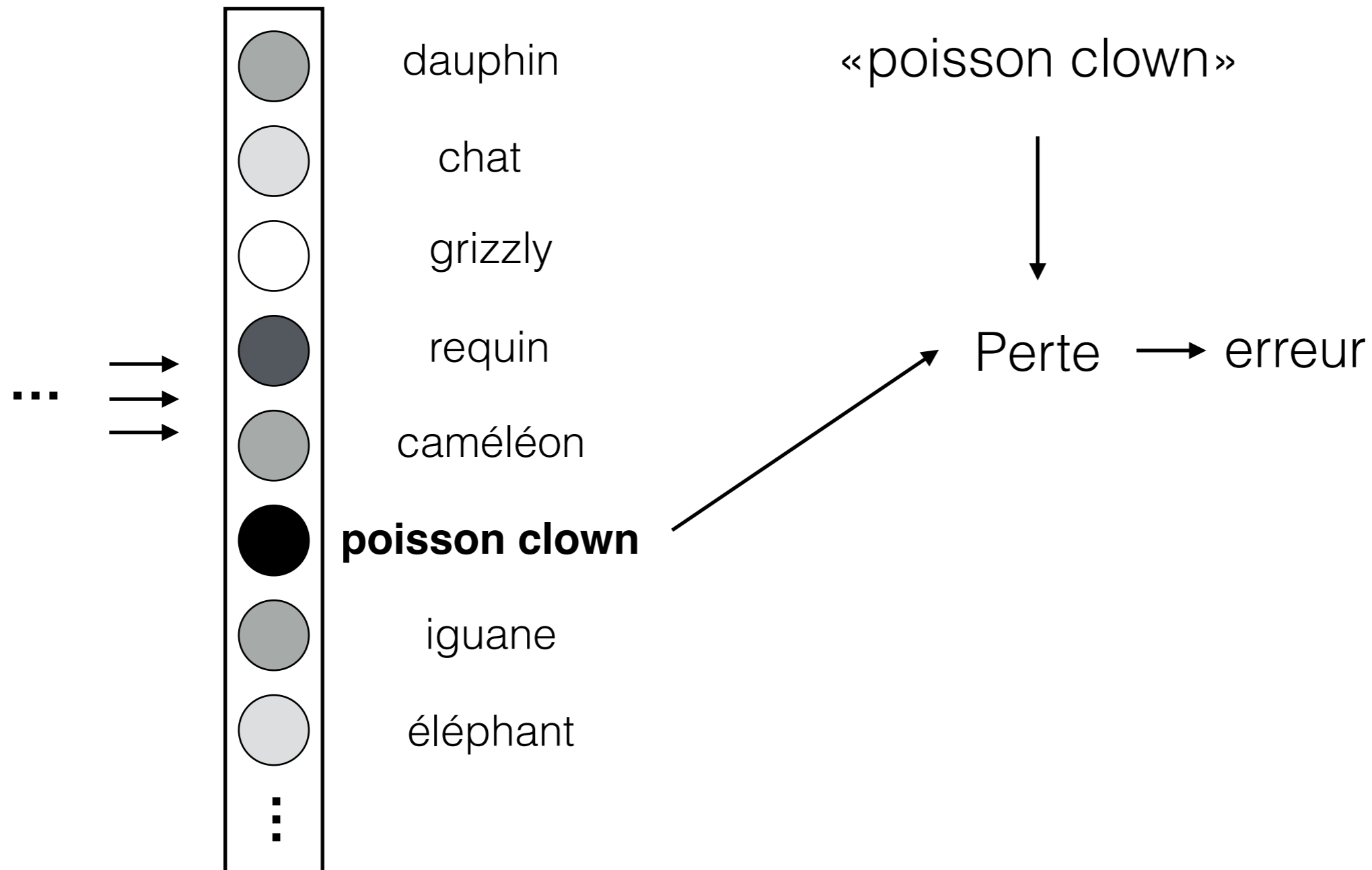


$$\underset{\mathbf{w}}{\operatorname{argmin}} L(w_1, \dots, w_6)$$

# Fonction de perte (*loss function*)

Sortie du réseau

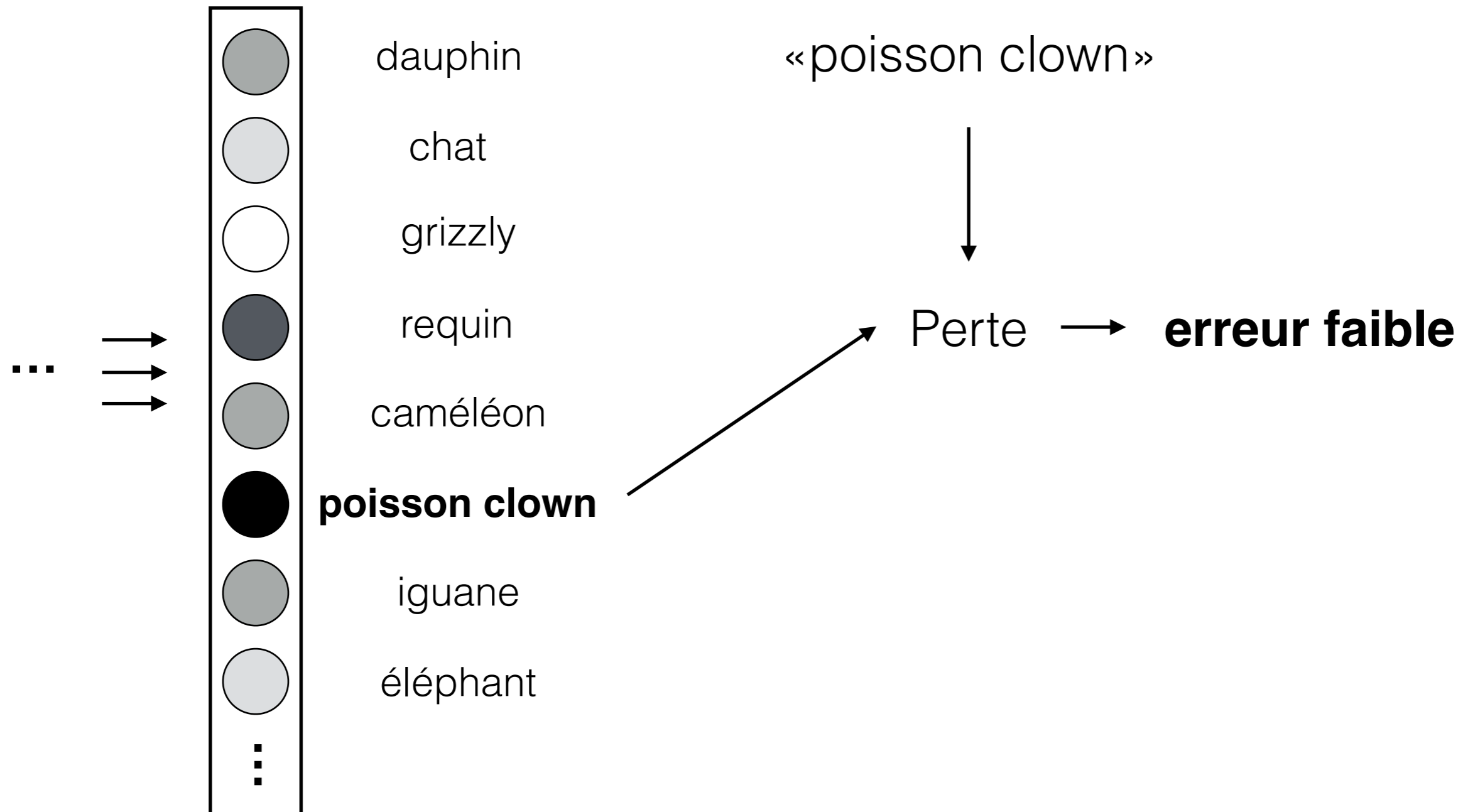
«Vraie» étiquette



# Fonction de perte (*loss function*)

Sortie du réseau

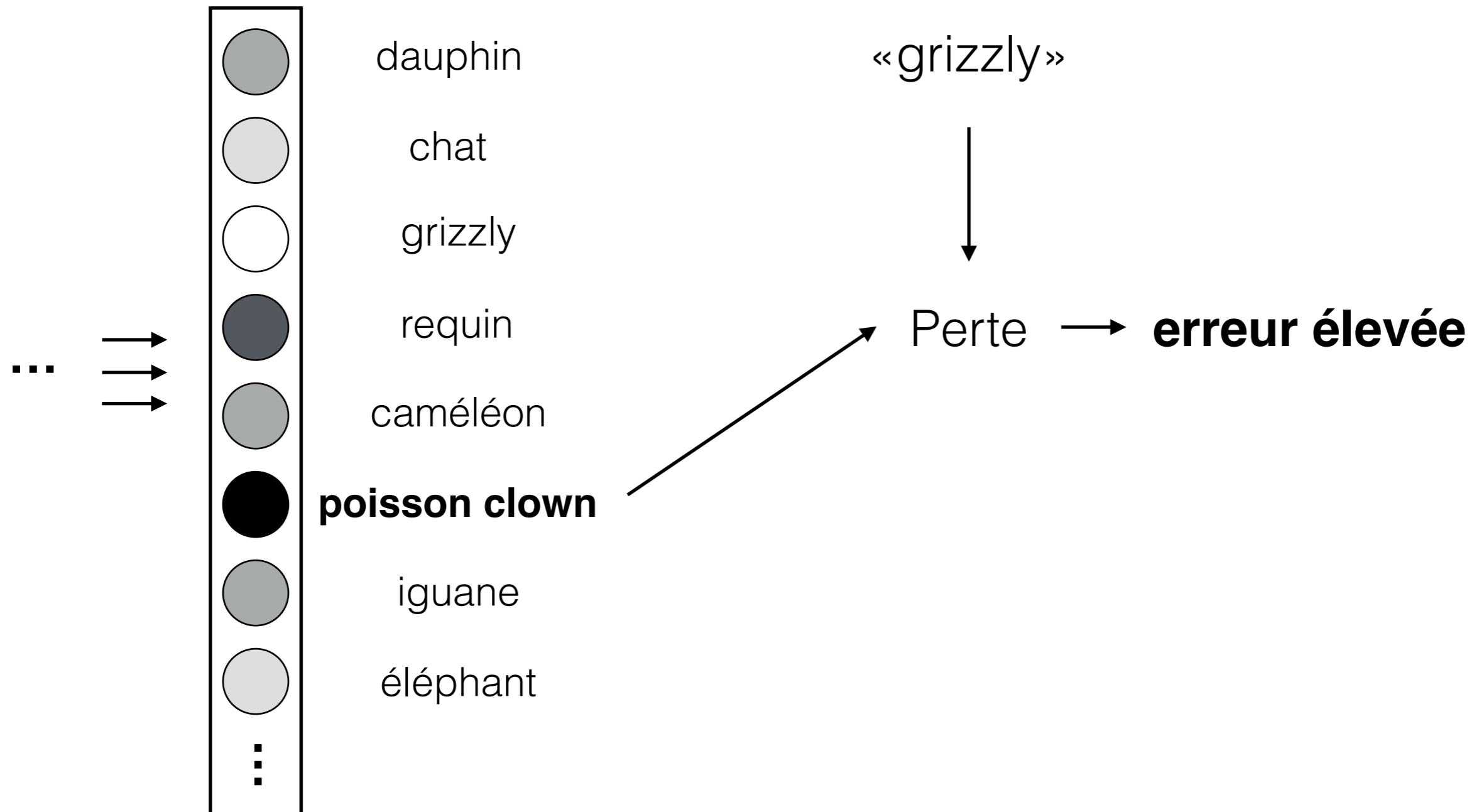
«Vraie» étiquette



# Fonction de perte (*loss function*)

Sortie du réseau

«Vraie» étiquette



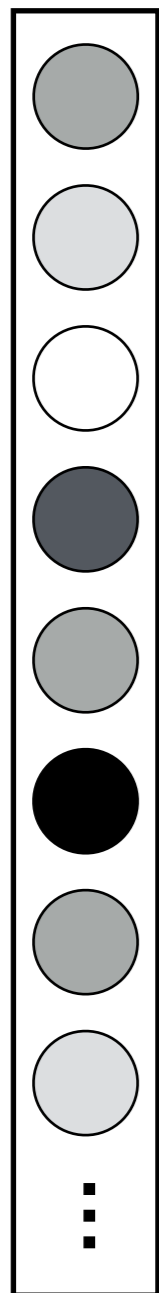
# Fonction de perte pour classification

Sortie du réseau

«Vraie» étiquette

$\hat{\mathbf{z}}$

$\mathbf{z}$



dauphin

chat

**grizzly**

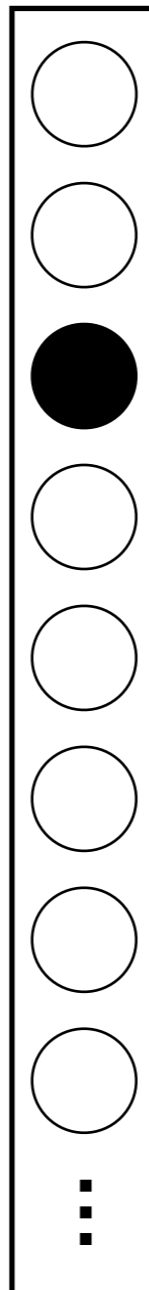
requin

caméléon

**poisson clown**

iguane

éléphant



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



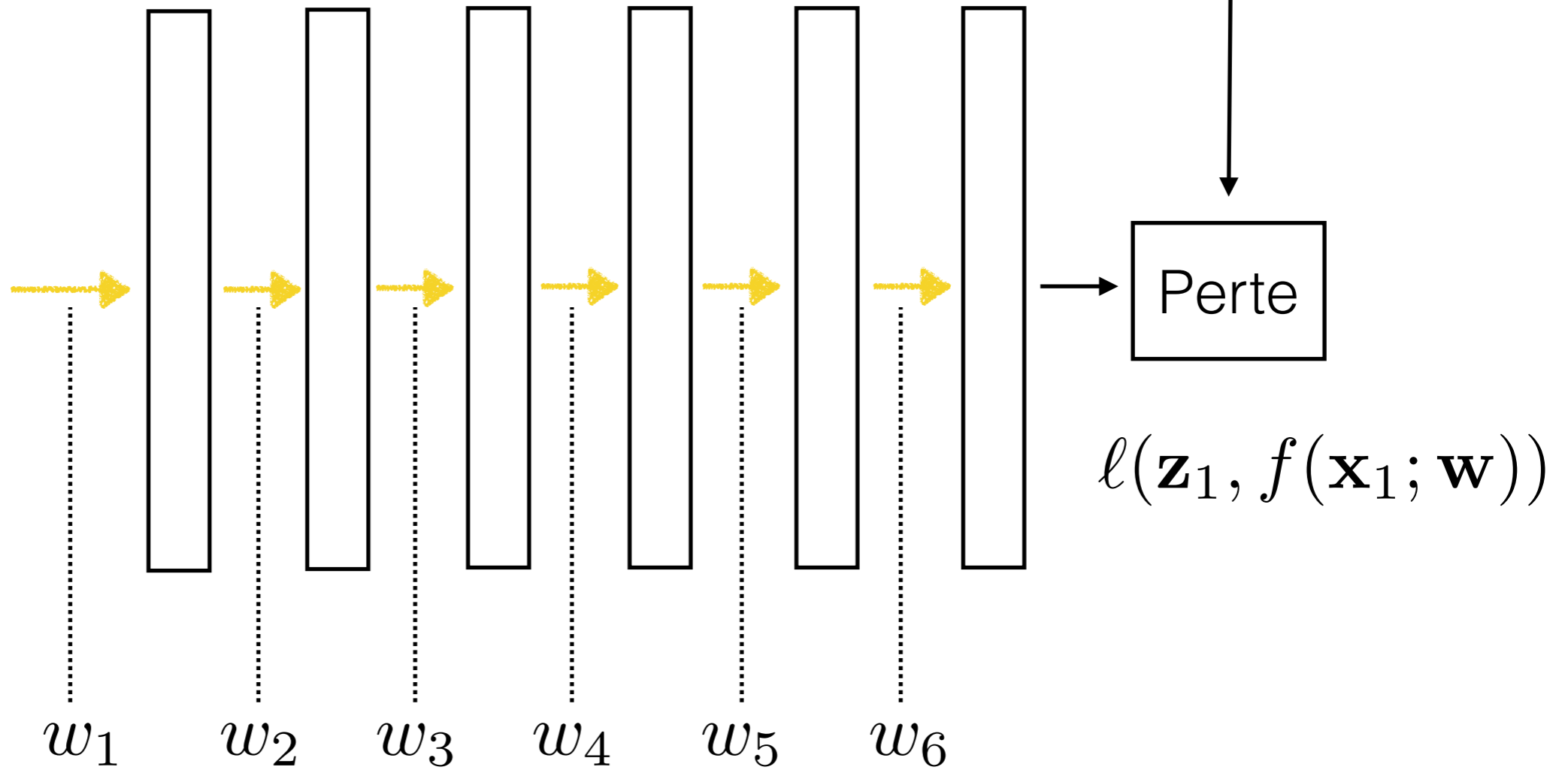
# Apprentissage par réseaux profonds

Appris automatiquement

$\mathbf{z}_1$

«poisson clown»

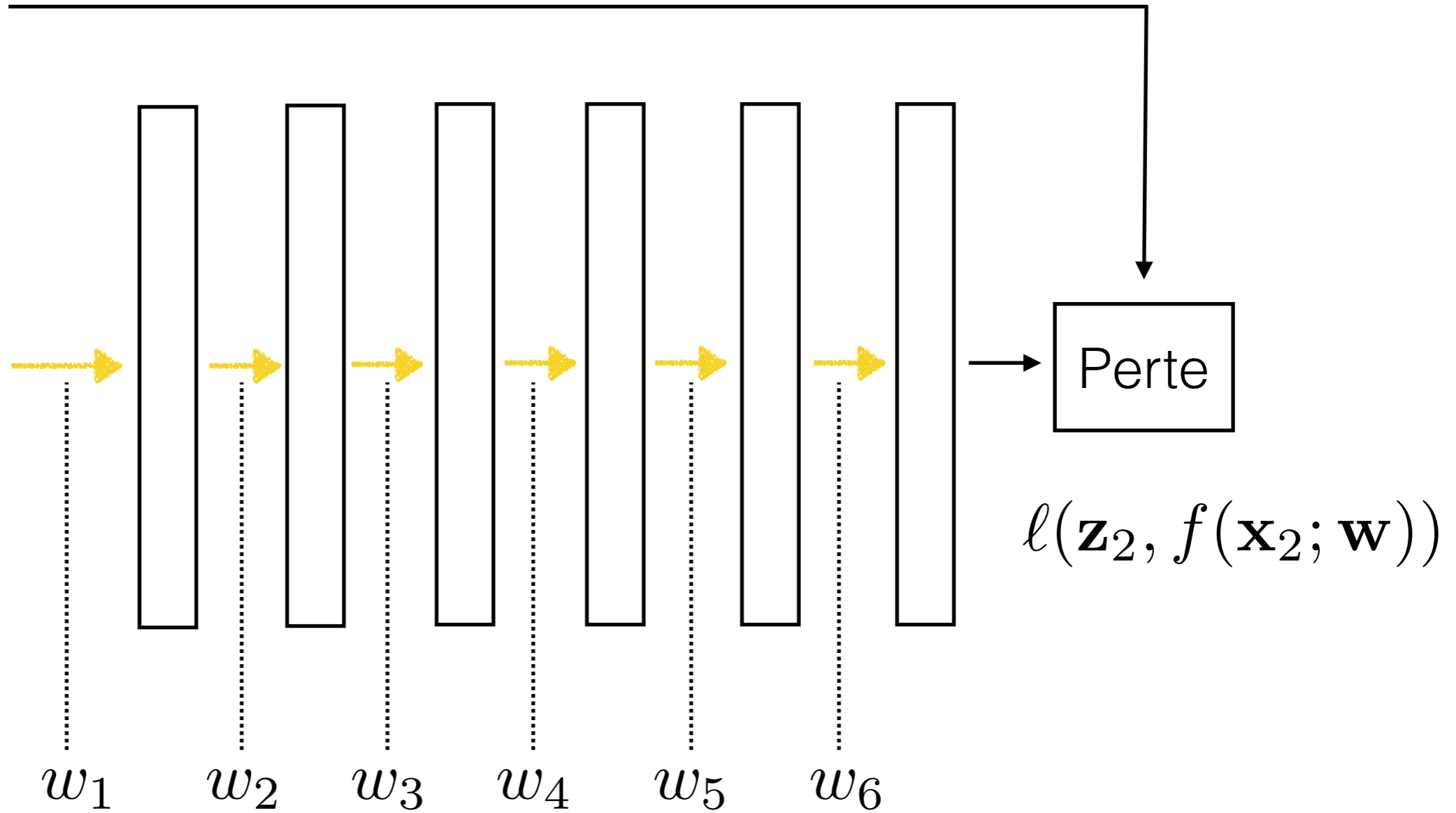
$\mathbf{x}_1$



# Apprentissage par réseaux profonds

Appris automatiquement

$\mathbf{z}_2$   
«grizzly»



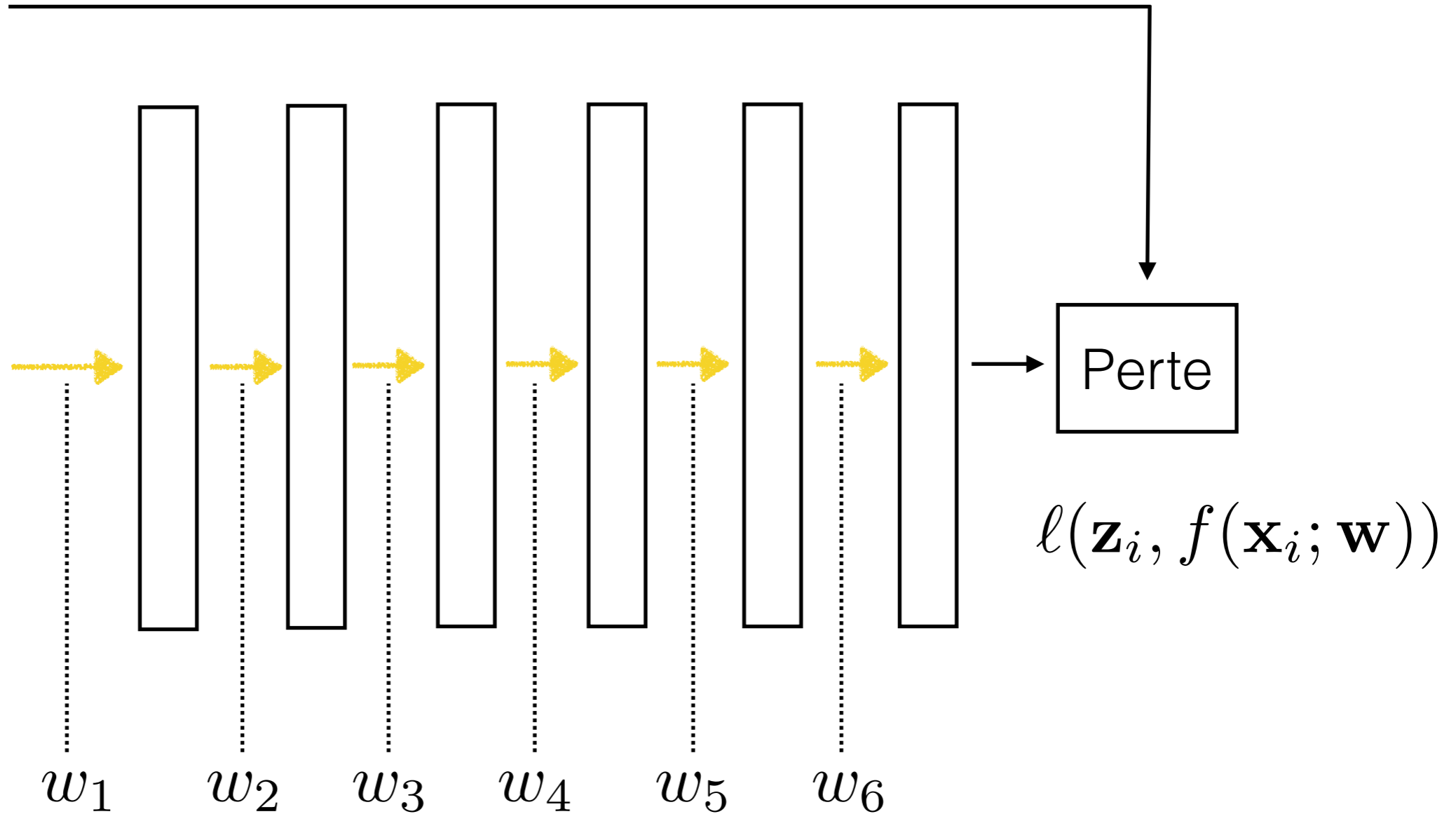
# Apprentissage par réseaux profonds

Appris automatiquement

$\mathbf{z}_i$

«caméléon»

$\mathbf{x}_i$



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

# Descente du gradient

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

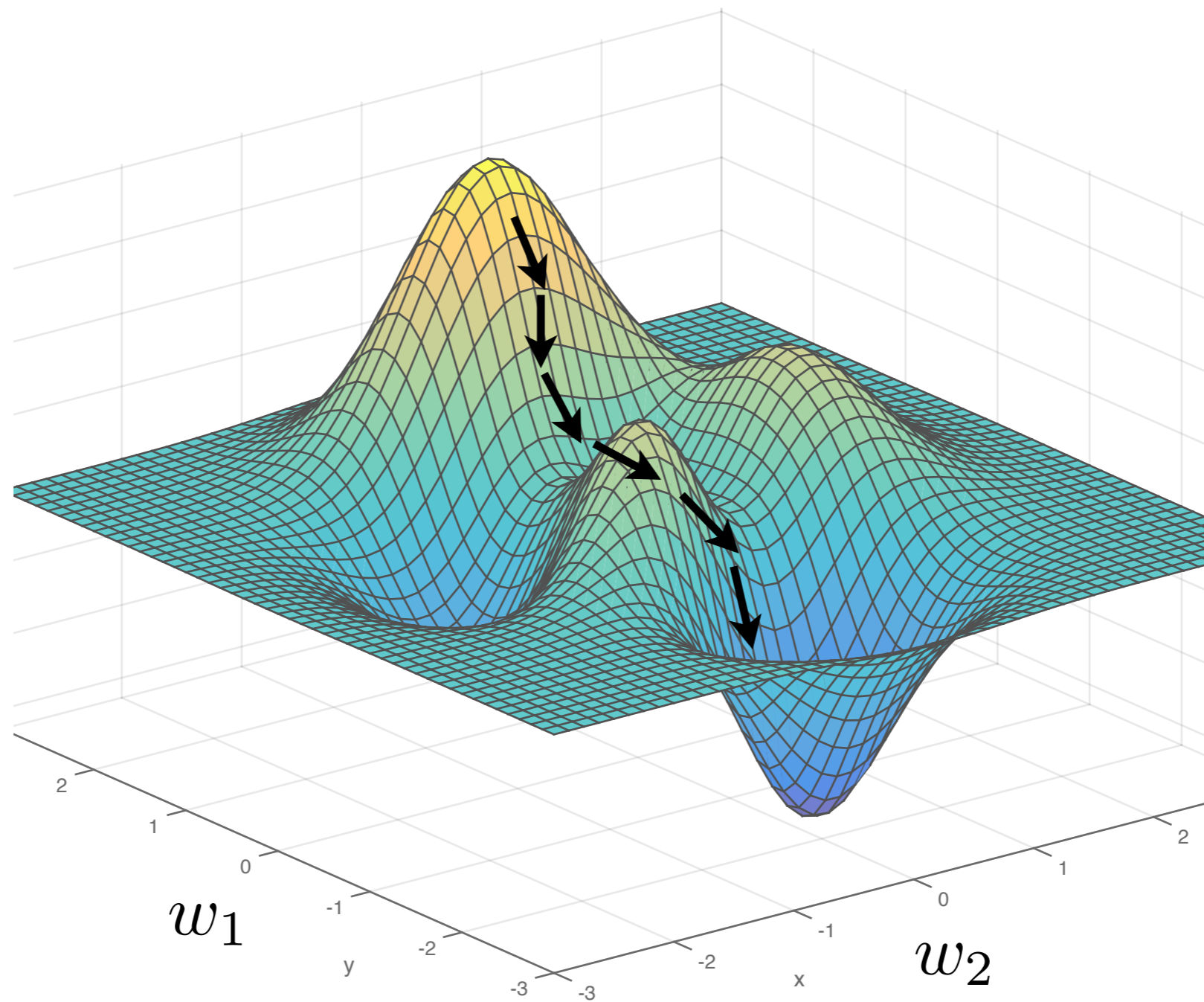
Une itération de descente du gradient

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

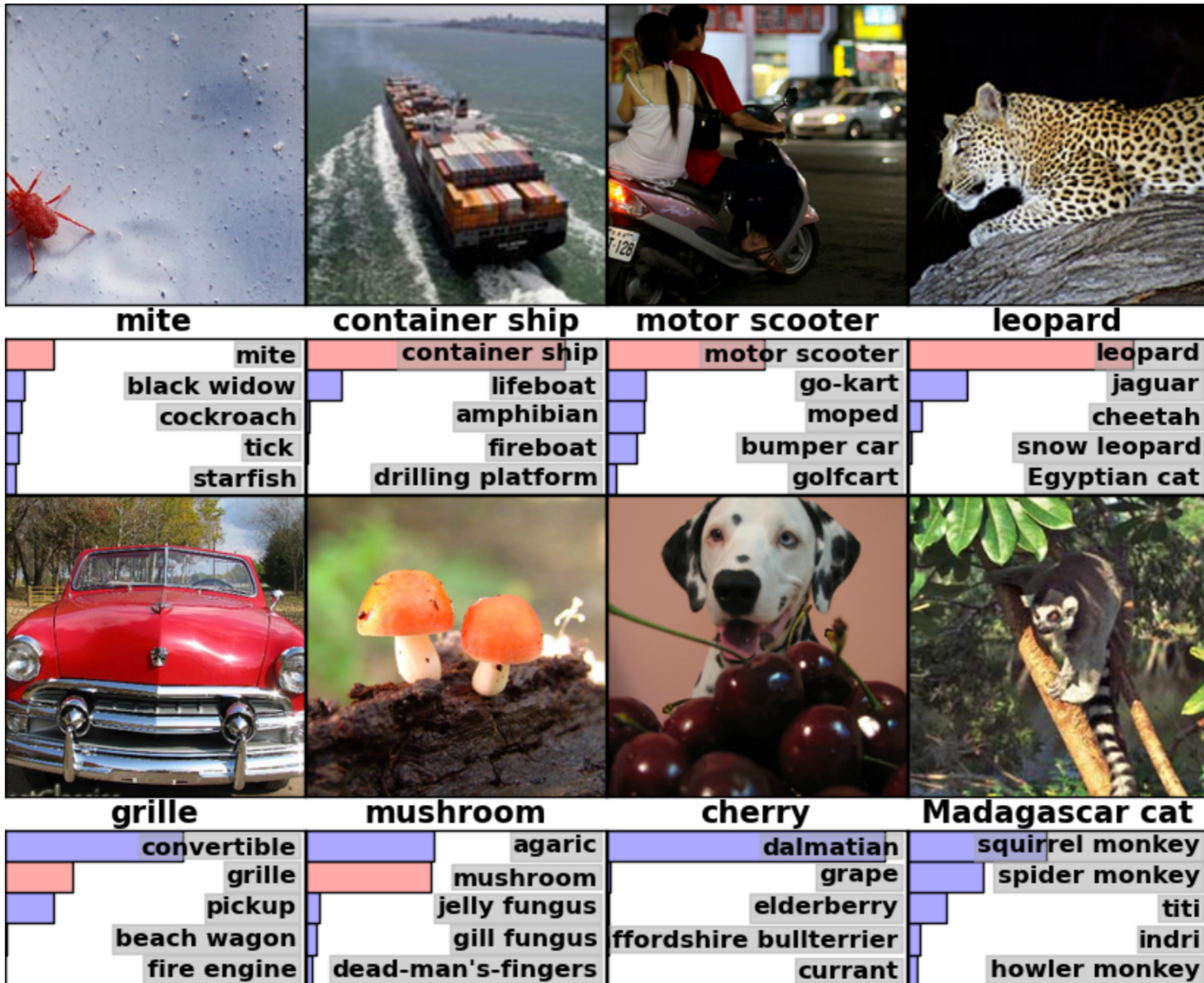
taux d'apprentissage

# Descente du gradient

$L(\mathbf{w})$



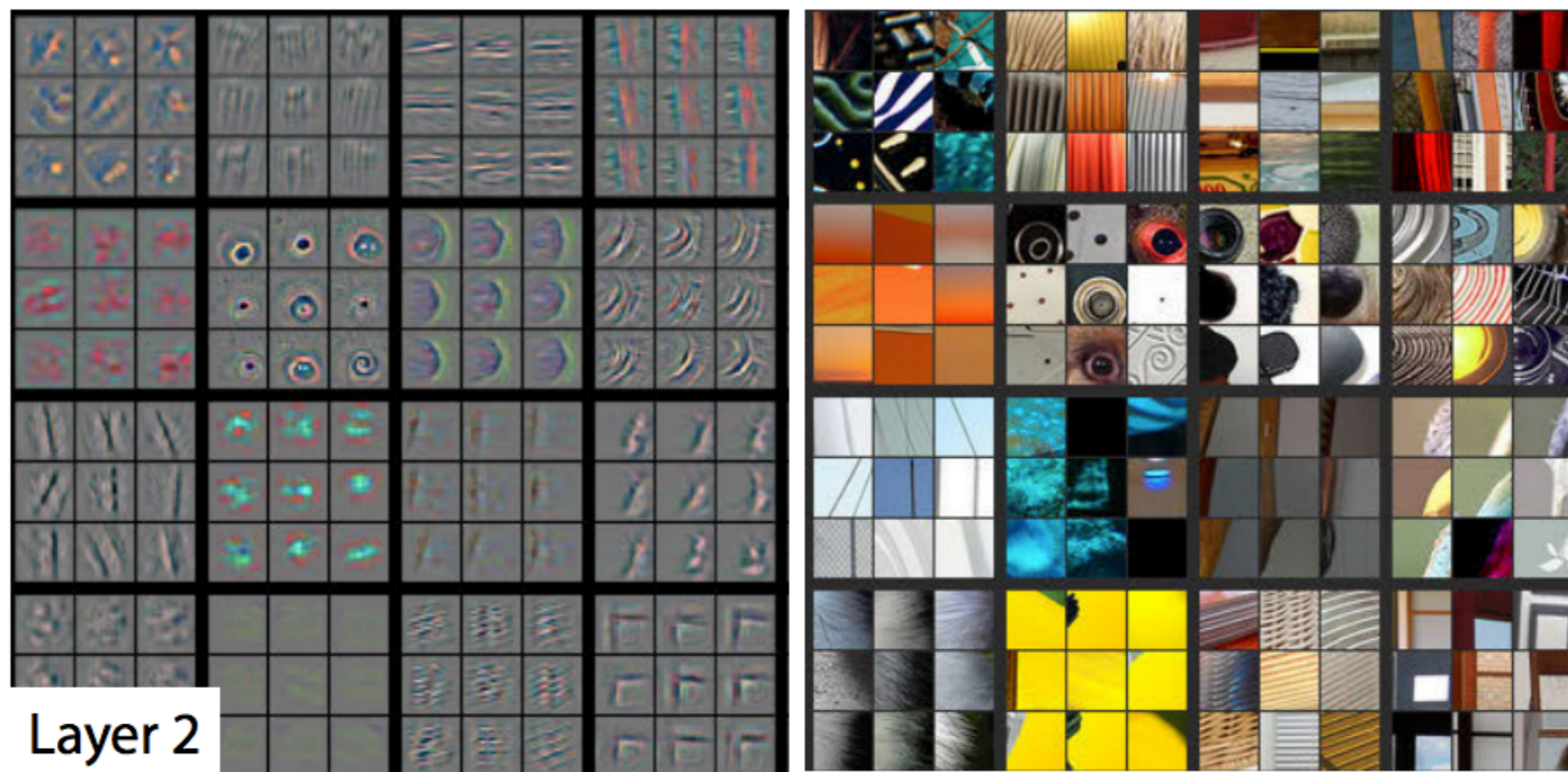
$$p(c|\mathbf{x})$$



# Qu'est-ce que le réseau apprend?

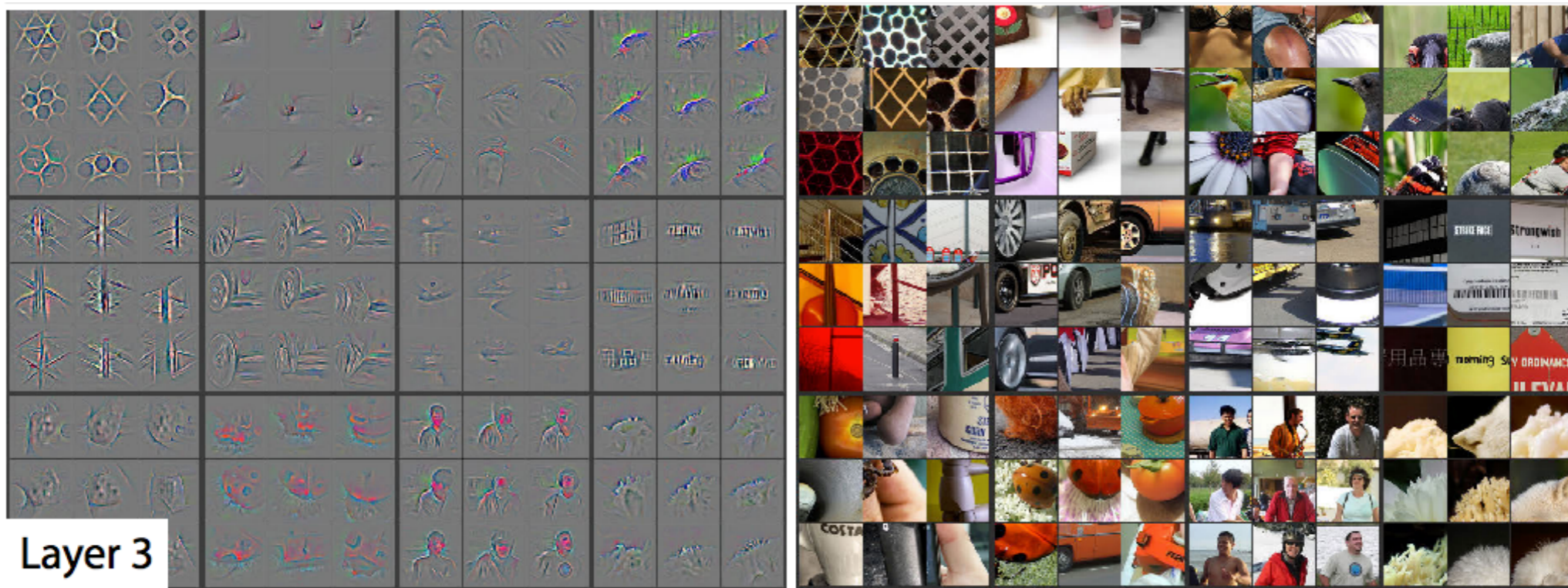


# Qu'est-ce que le réseau apprend?

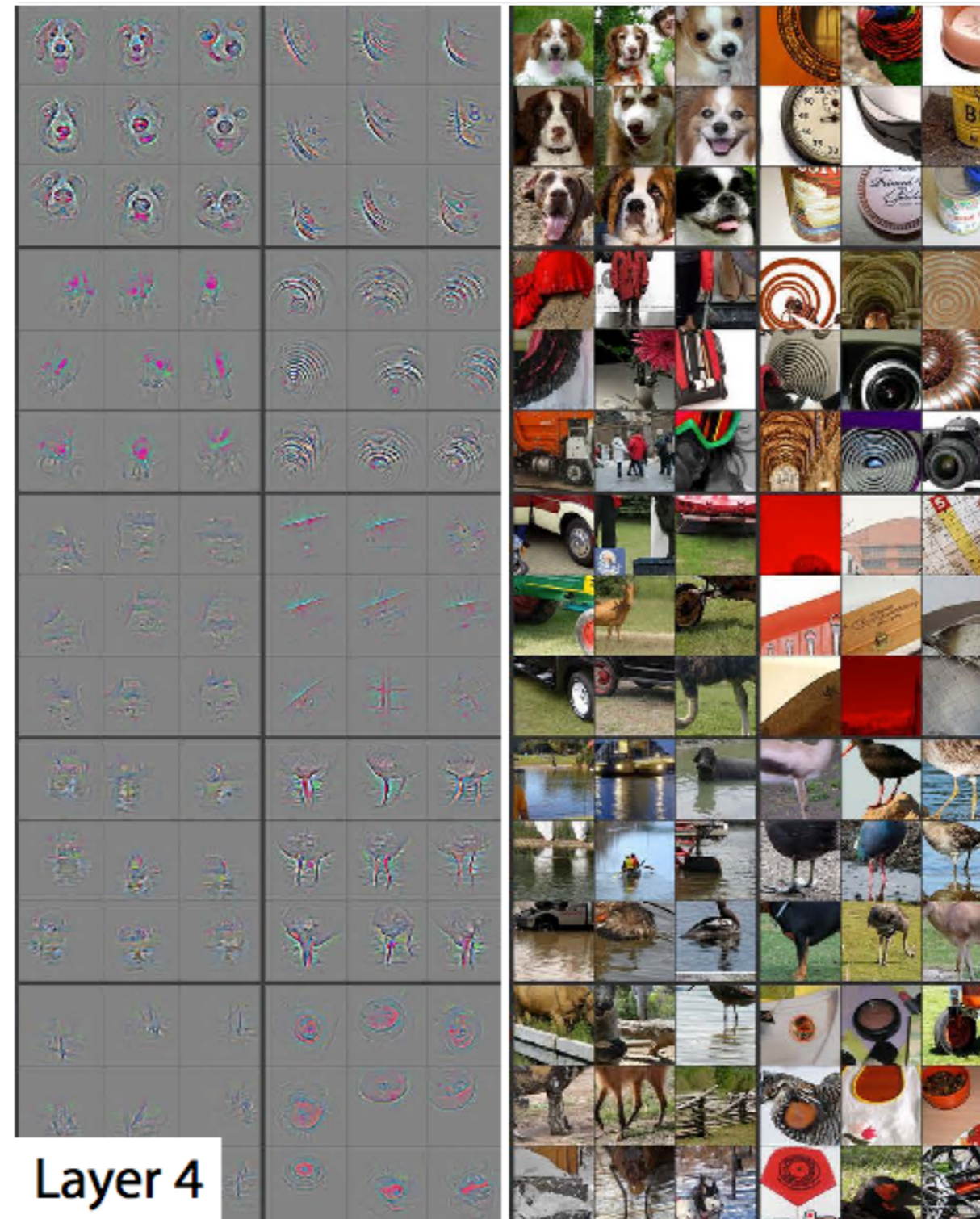




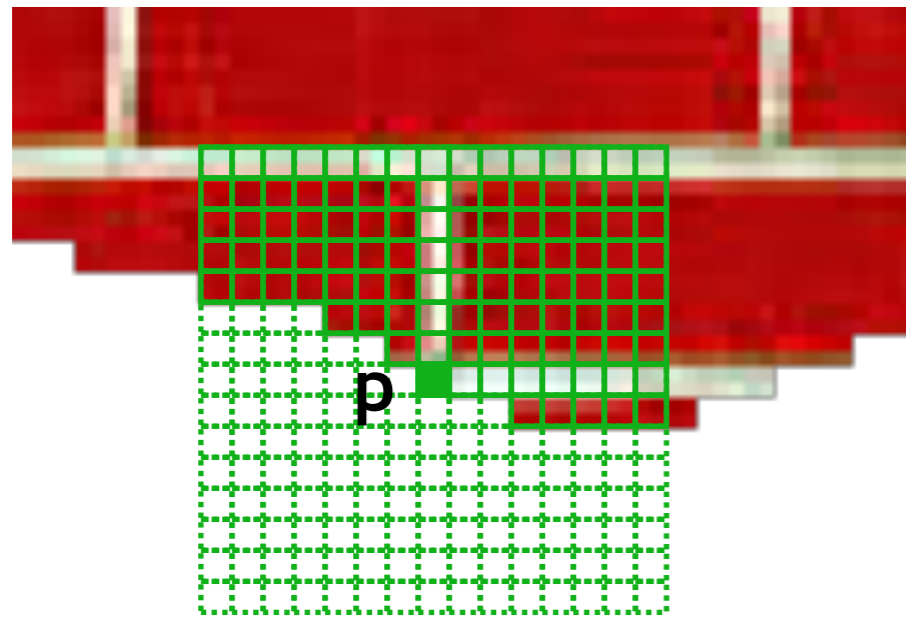
# Qu'est-ce que le réseau apprend?



# Qu'est-ce que le réseau apprend?



# Synthèse de texture: rappel



Synthétise un pixel

échantillonnage  
non-paramétrique

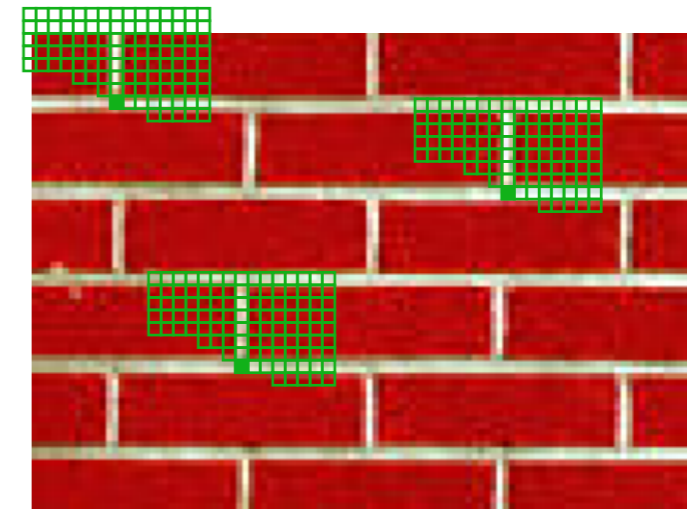
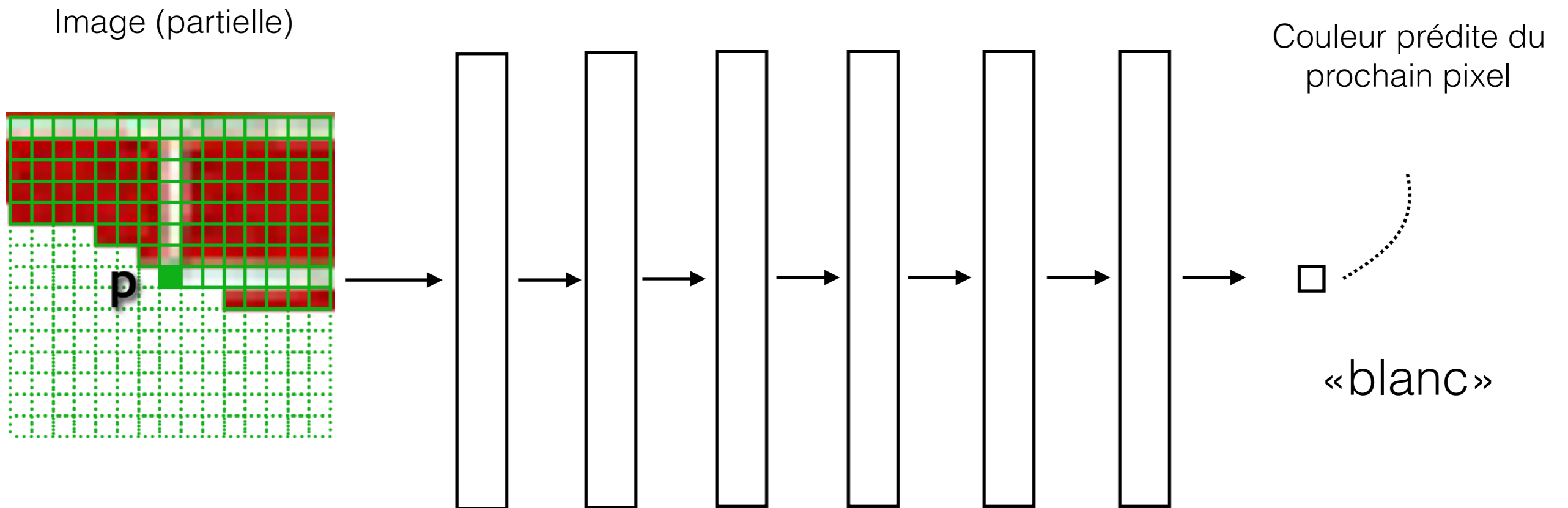


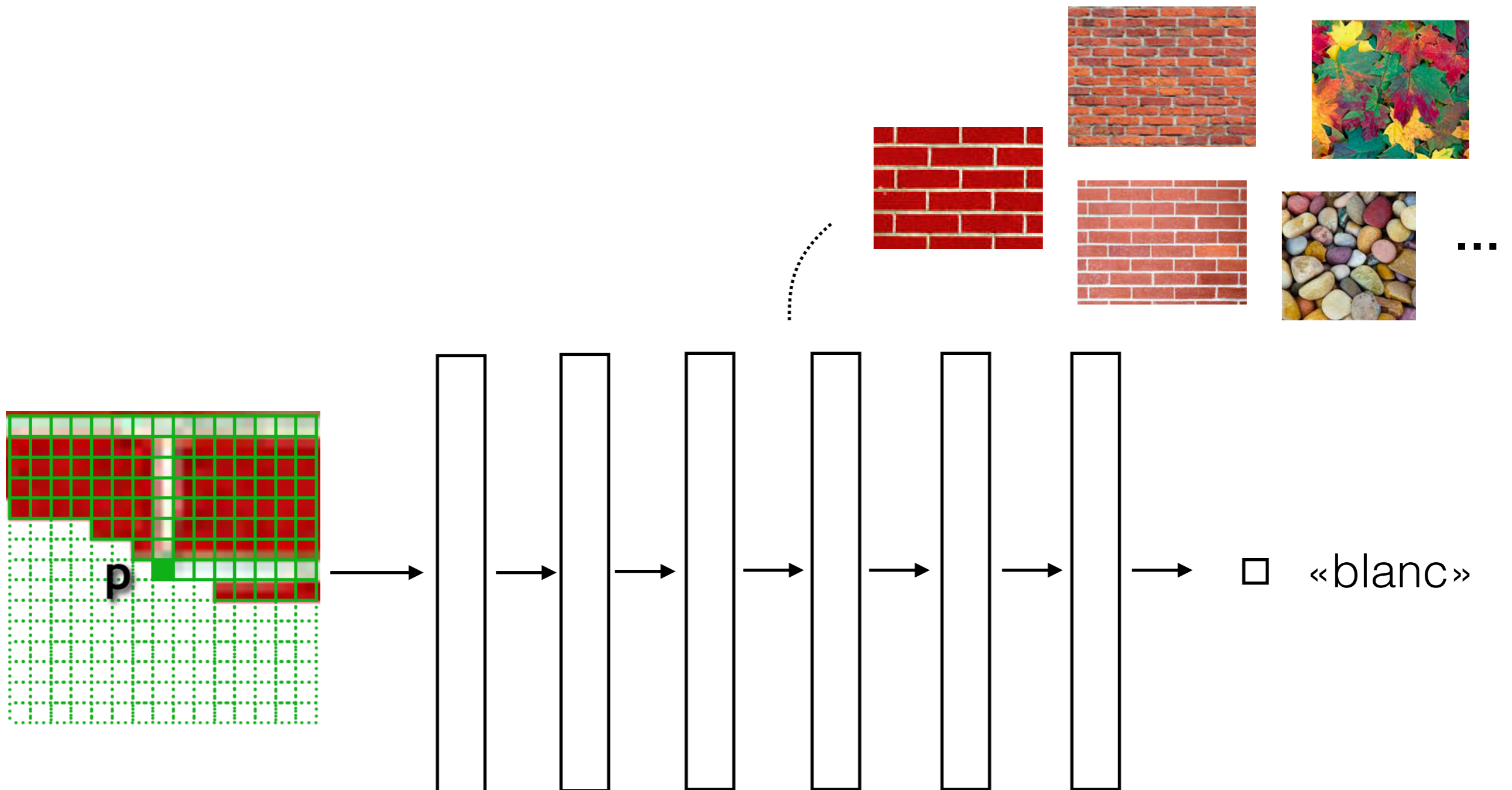
Image d'entrée

Modélise  $P(p|N(p))$

# Synthèse de texture par réseau profond

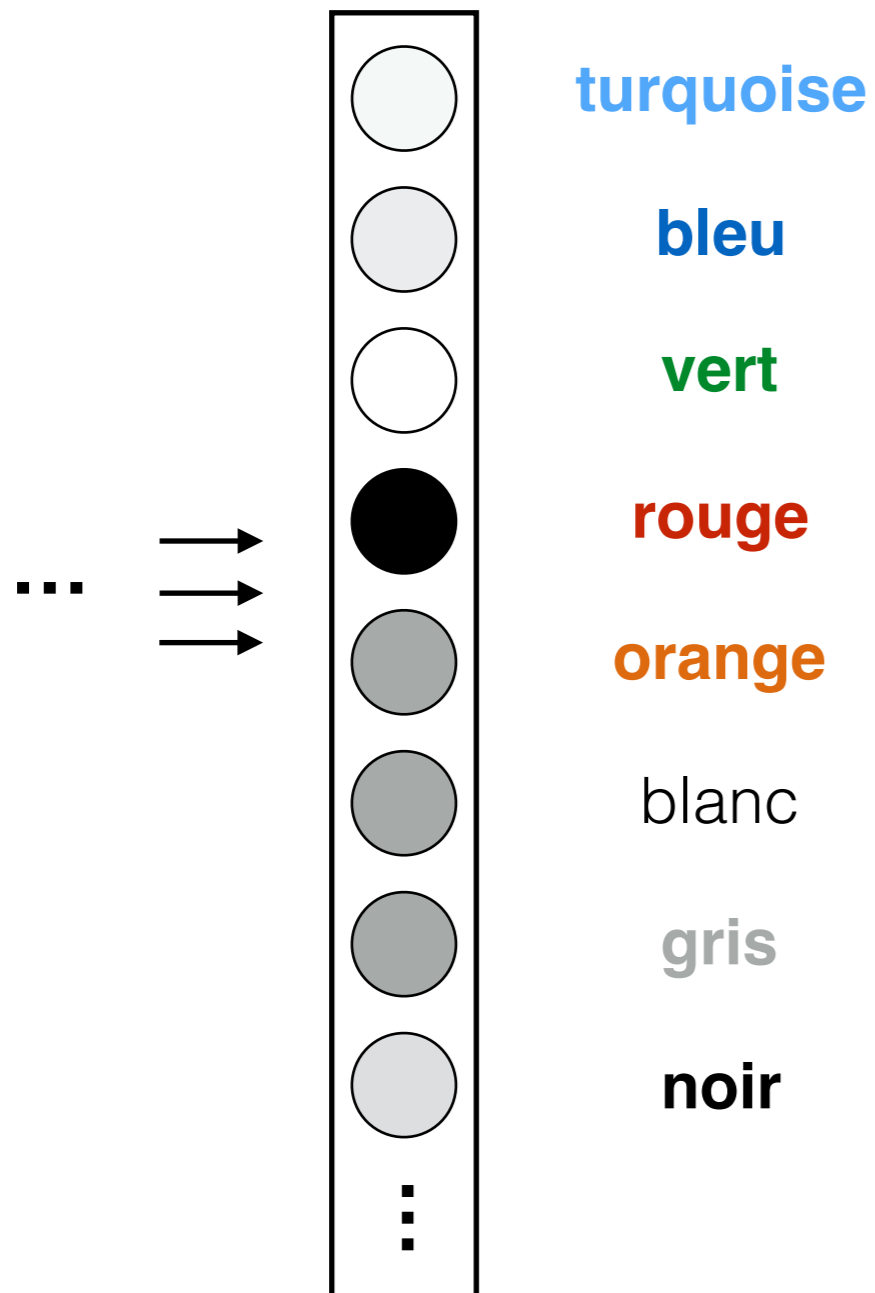


# Synthèse de texture par réseau profond



# Échantillonnage

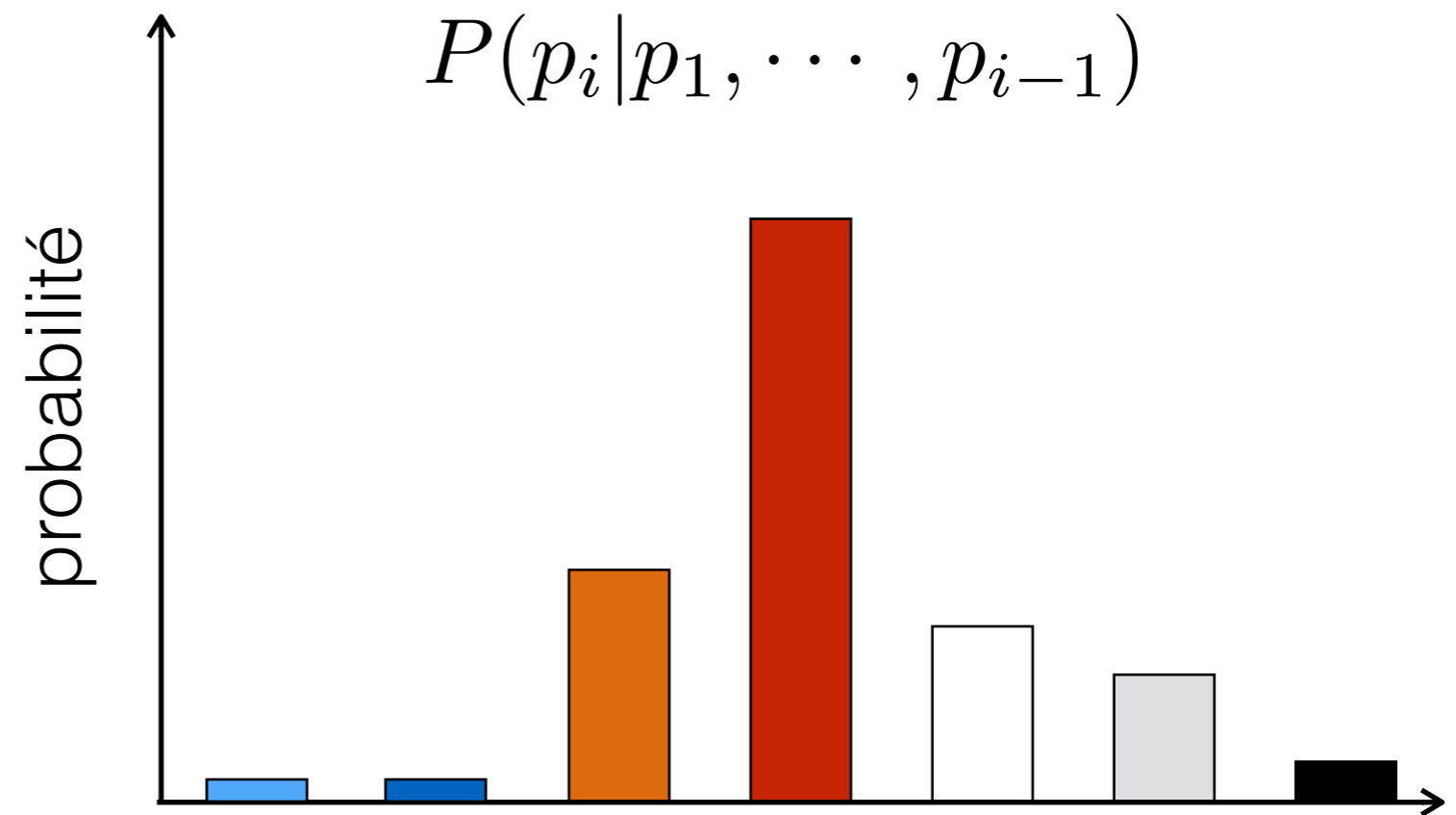
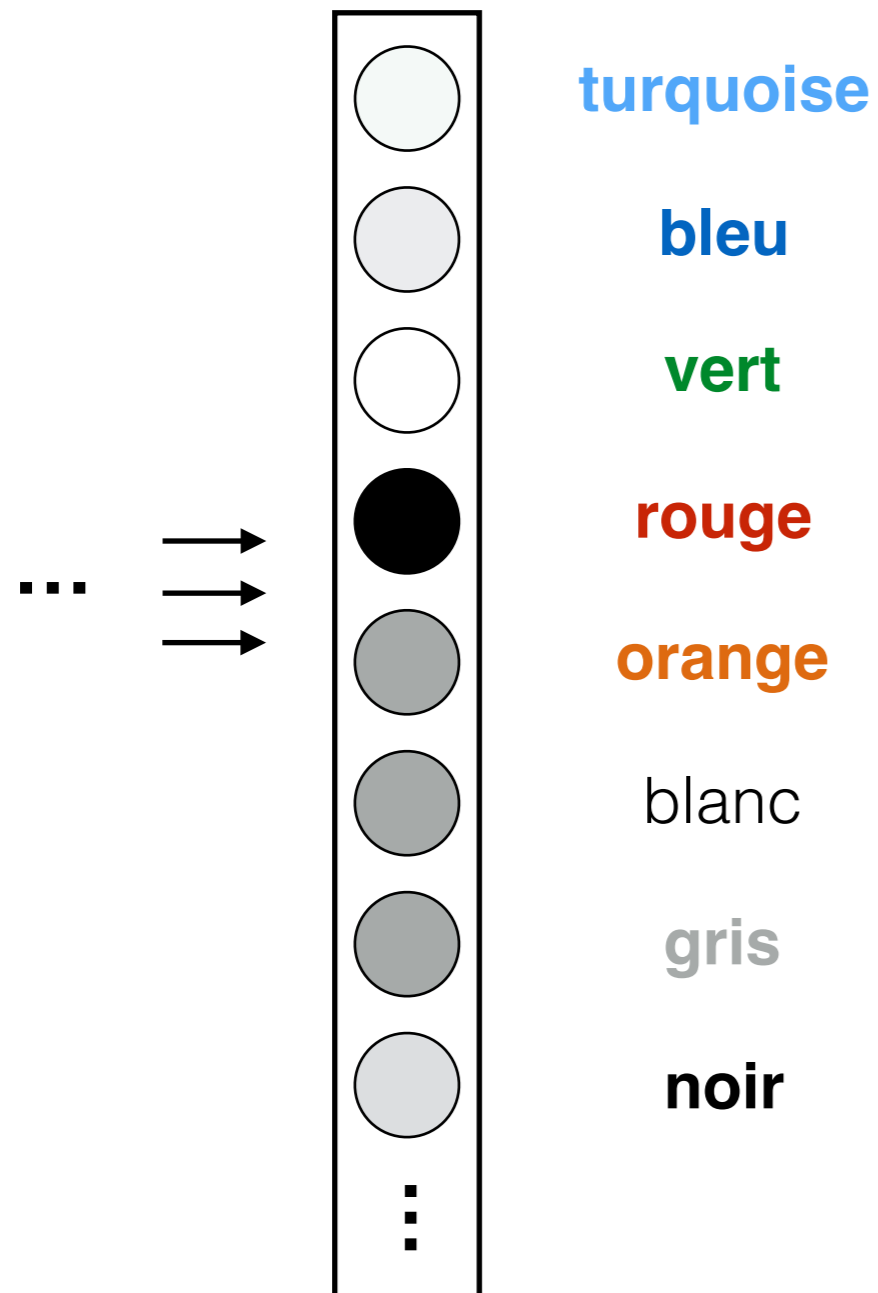
Sortie du réseau



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

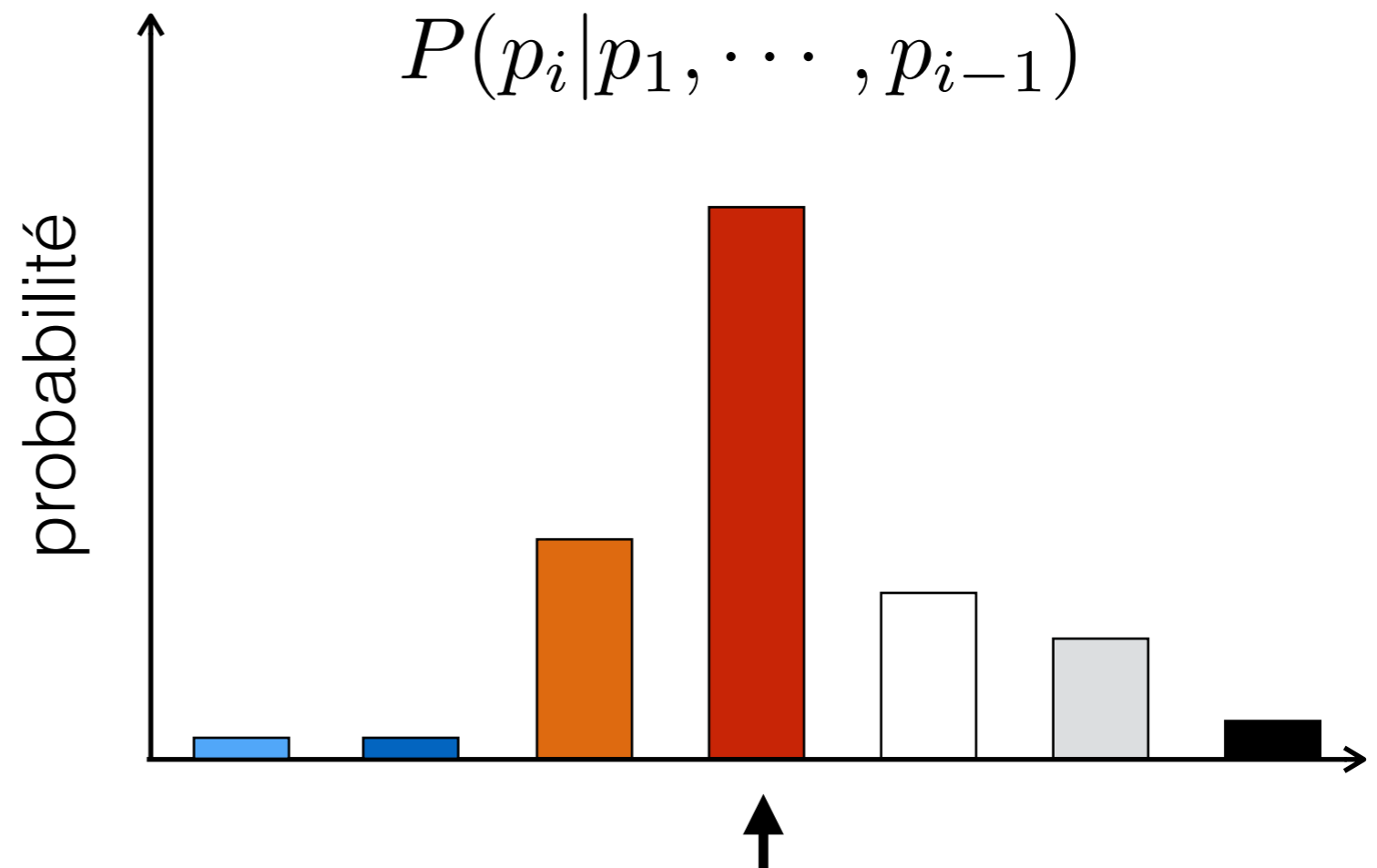
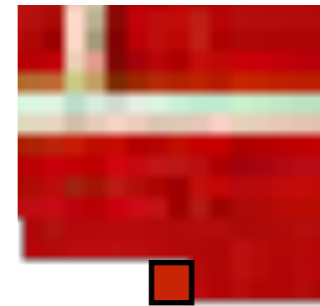
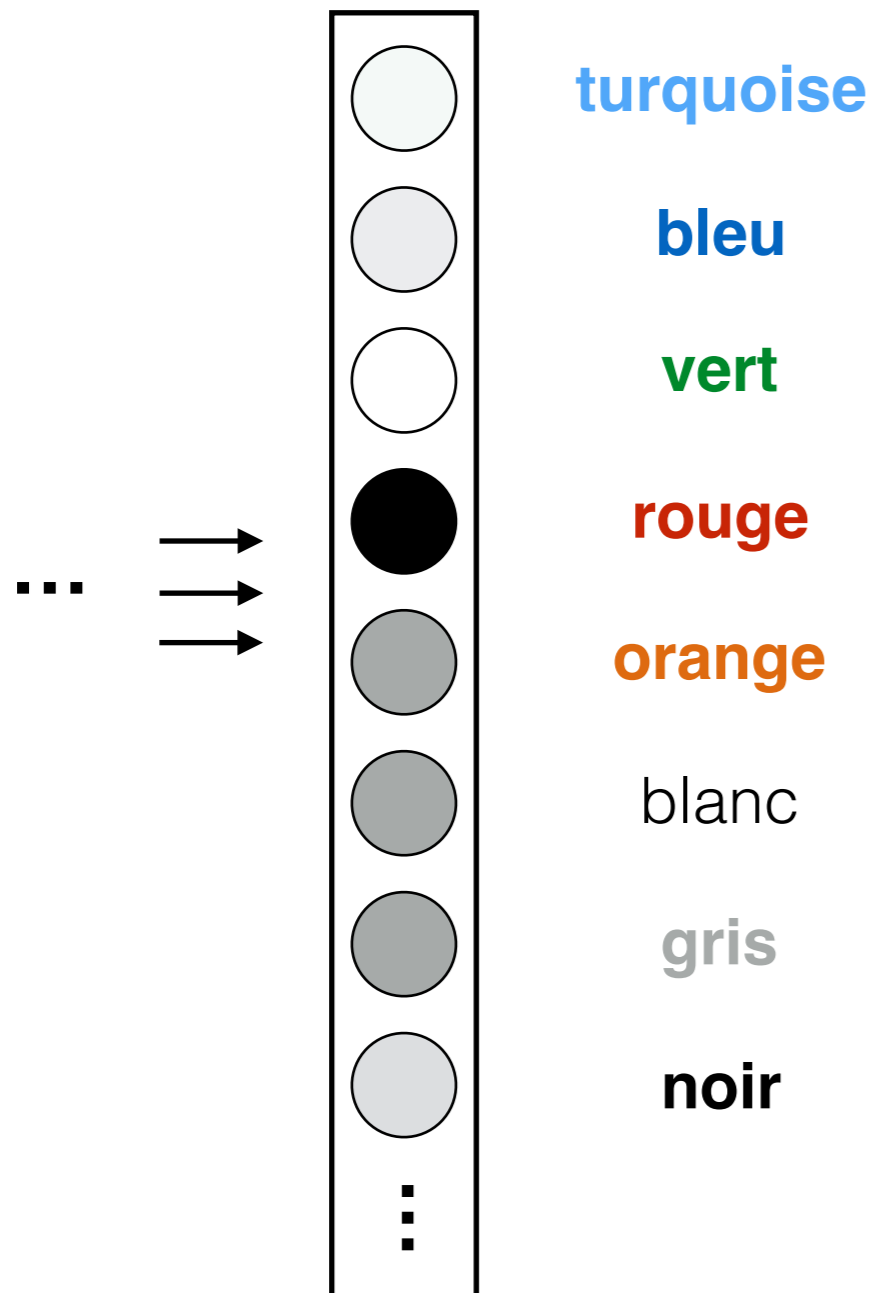
# Échantillonnage

Sortie du réseau



# Échantillonnage

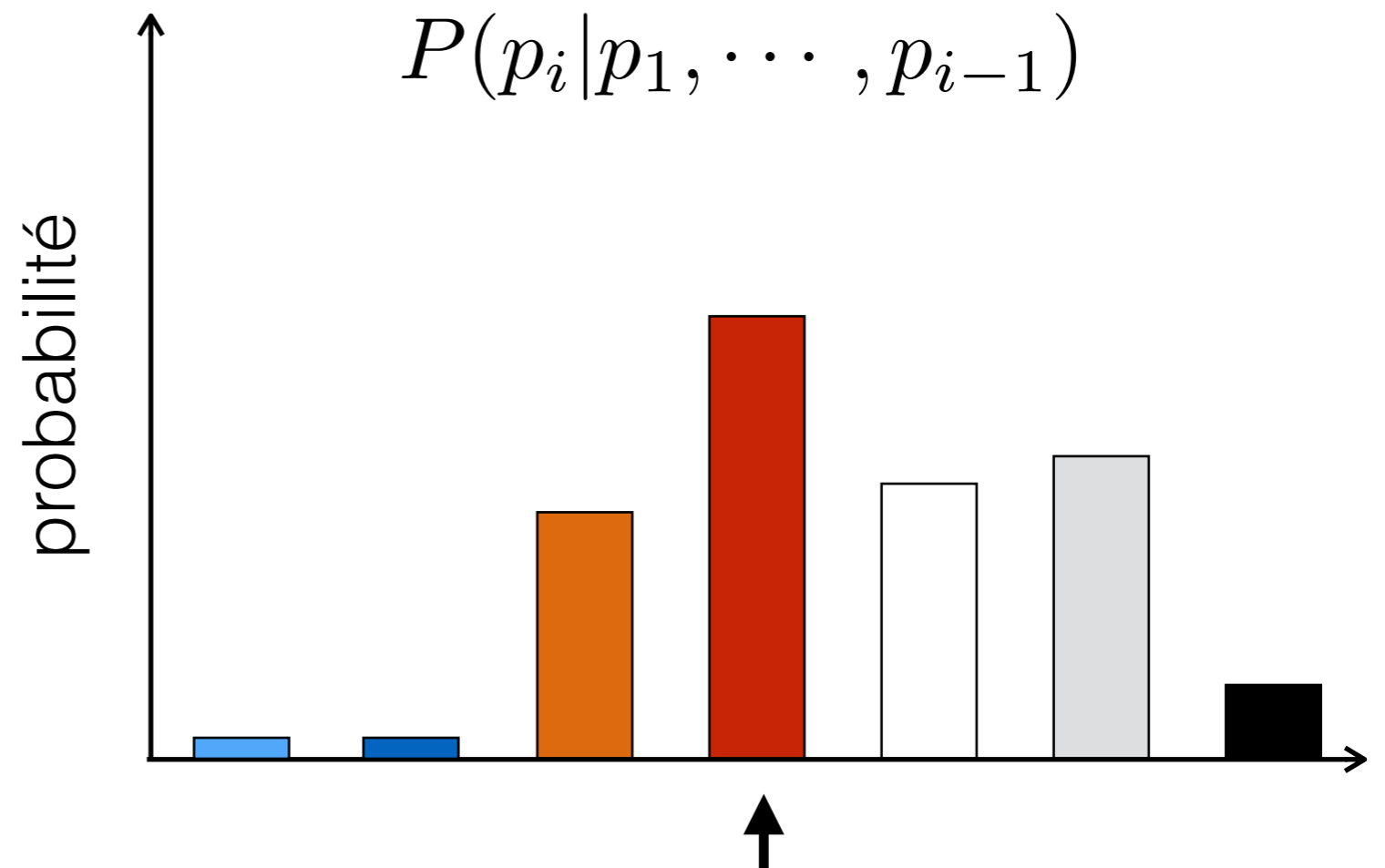
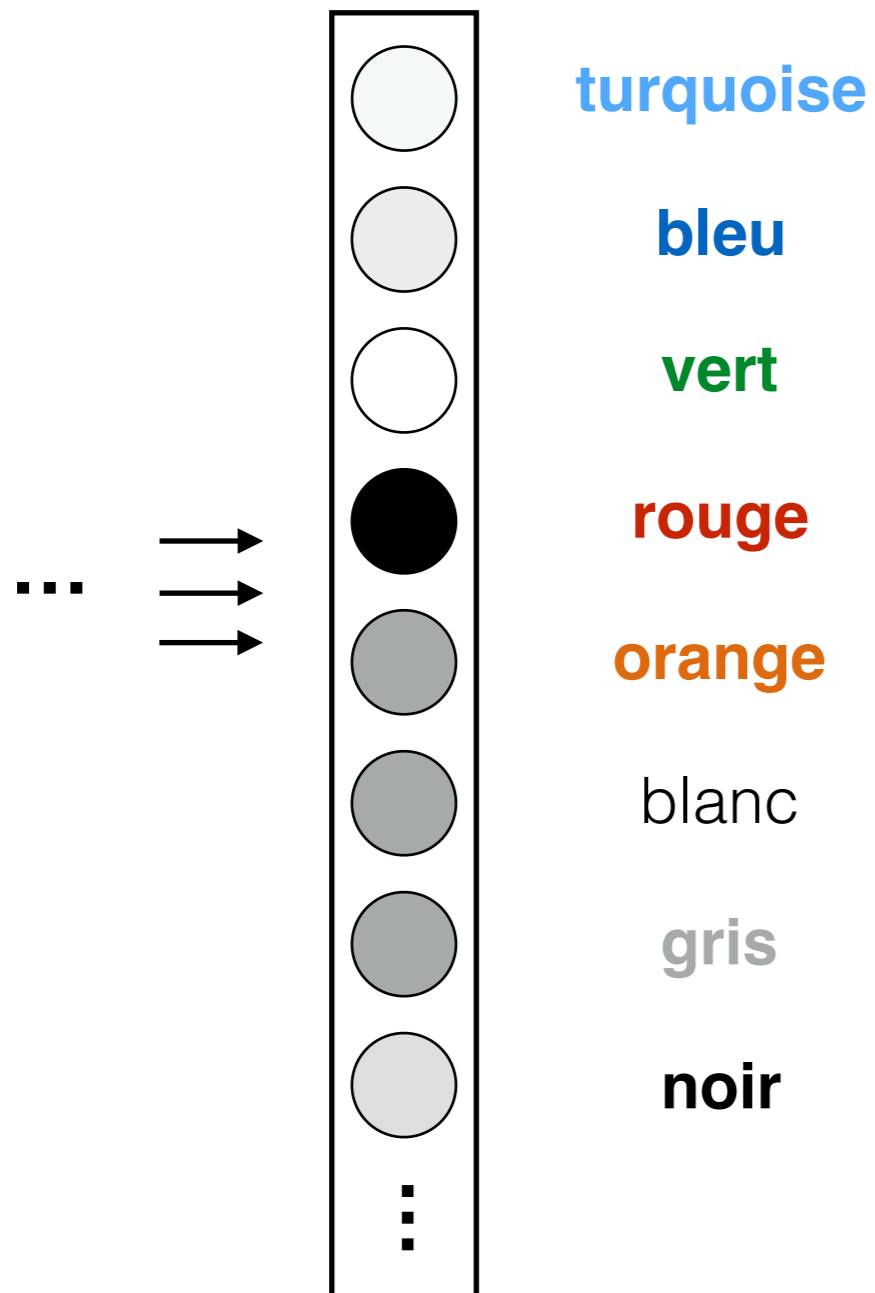
Sortie du réseau





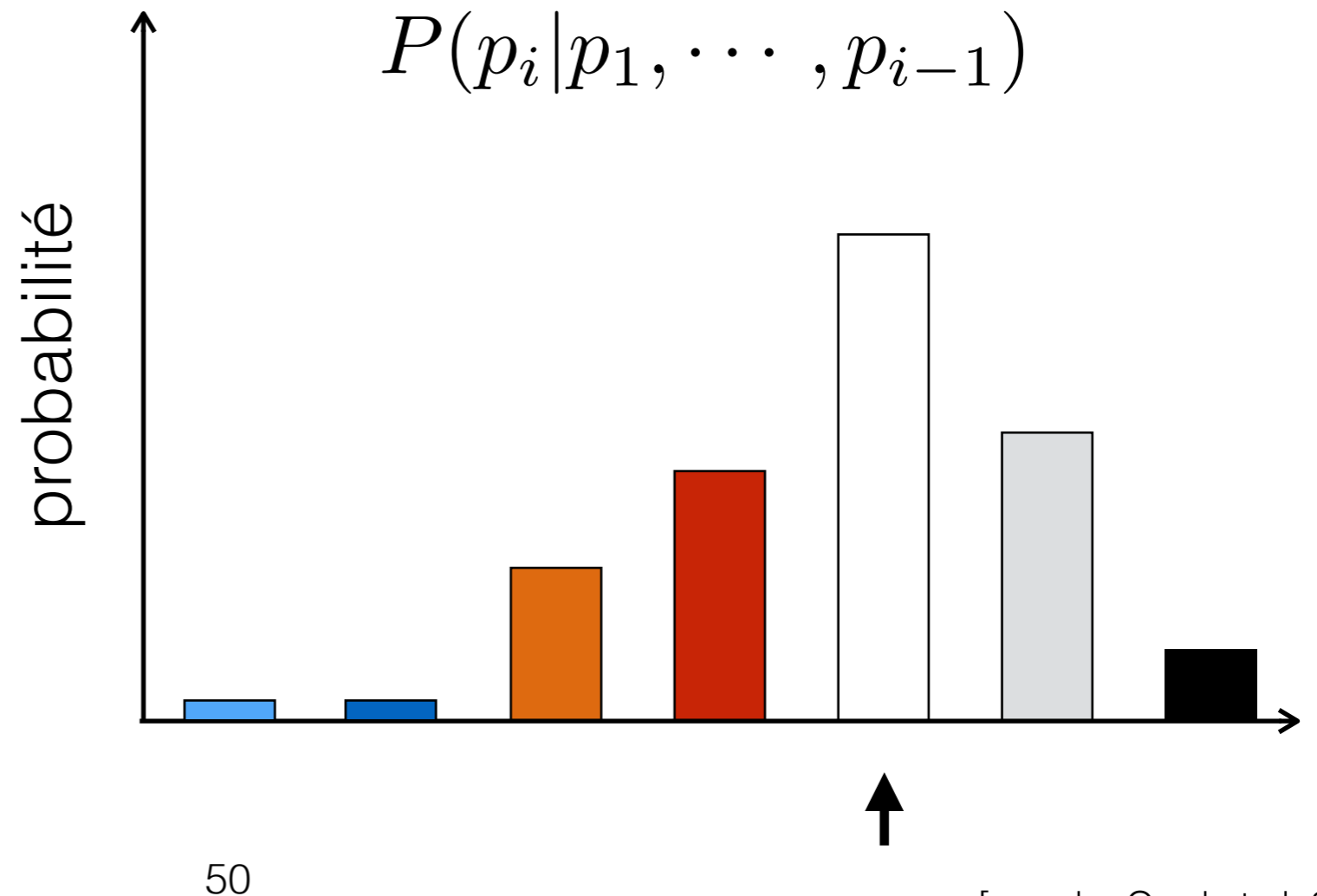
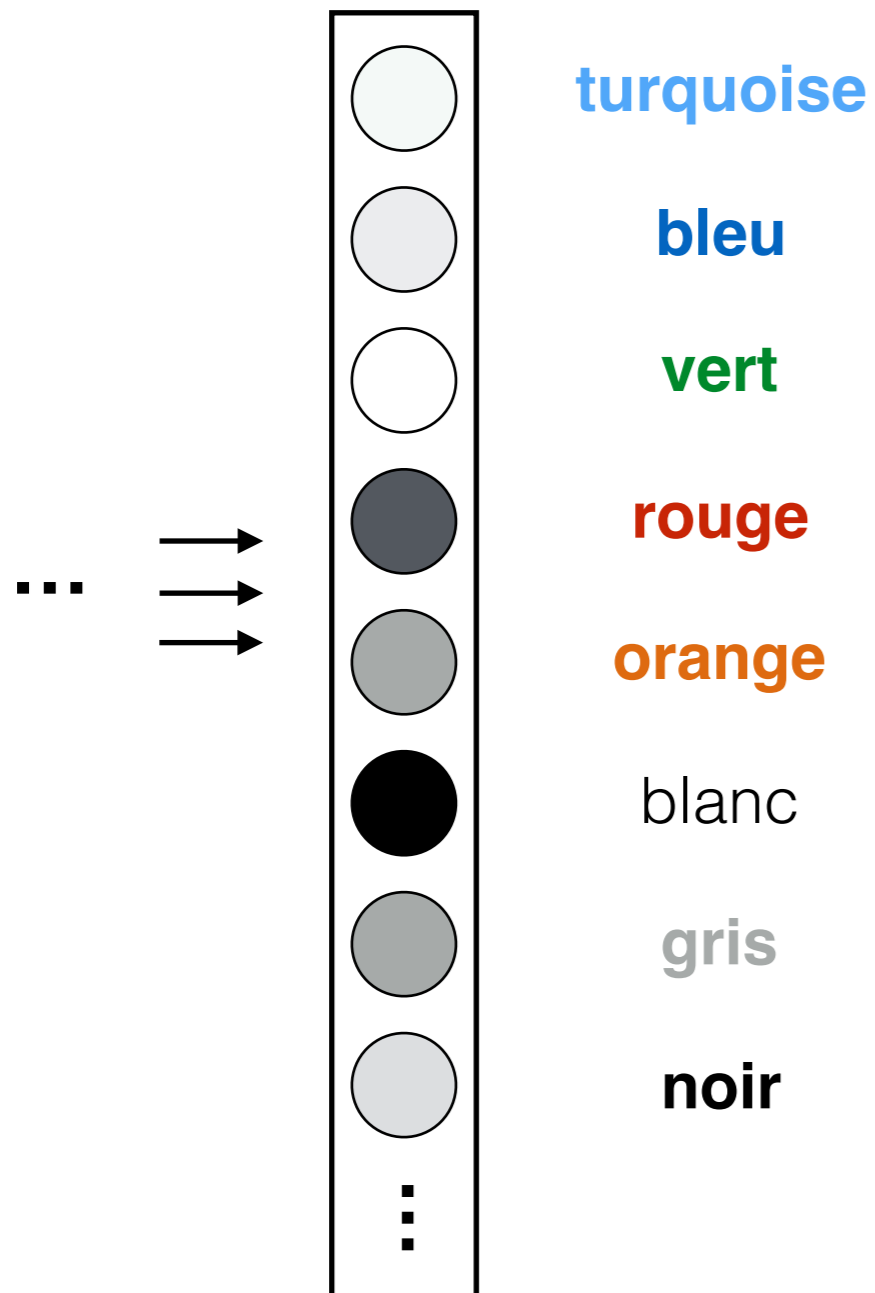
# Échantillonnage

Sortie du réseau



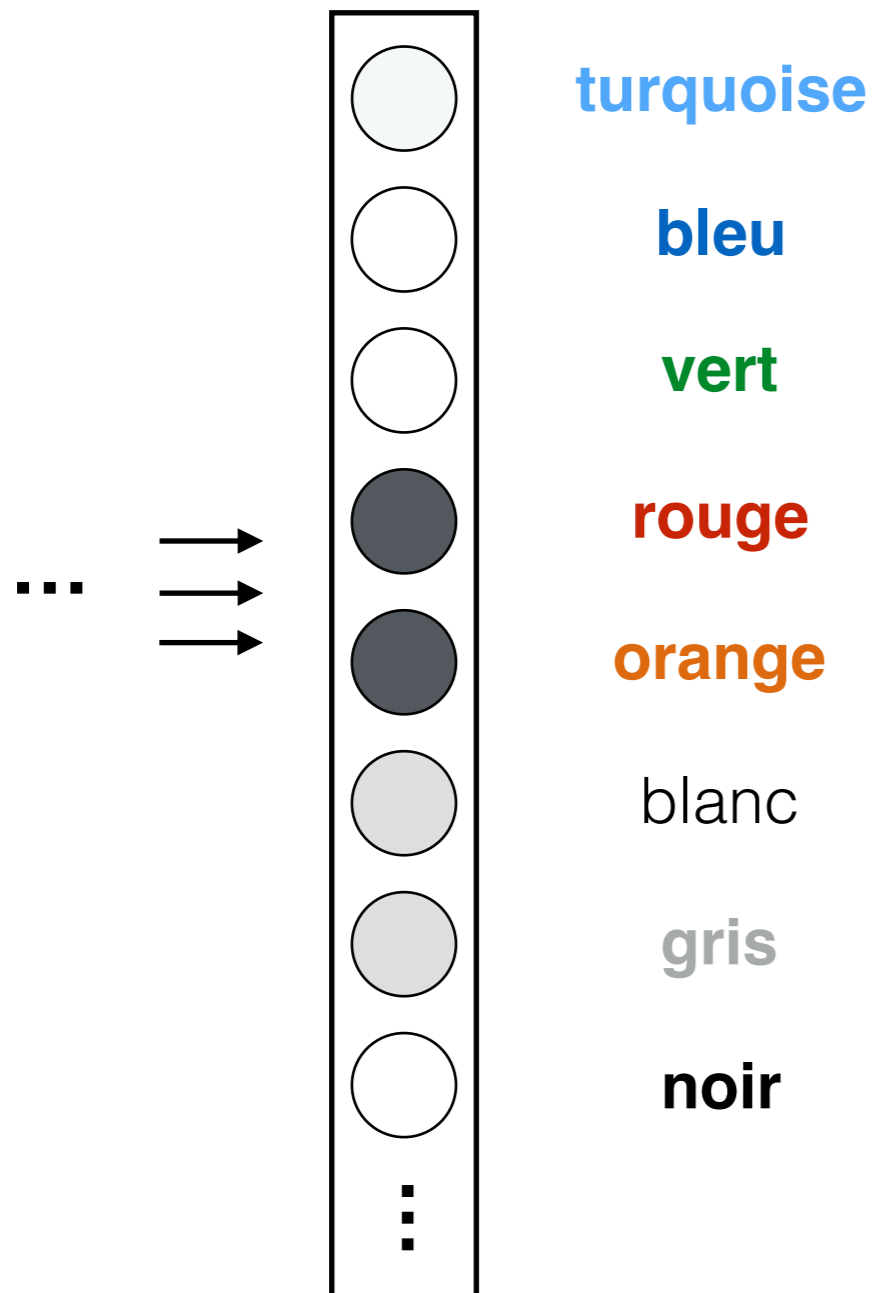
# Échantillonnage

Sortie du réseau

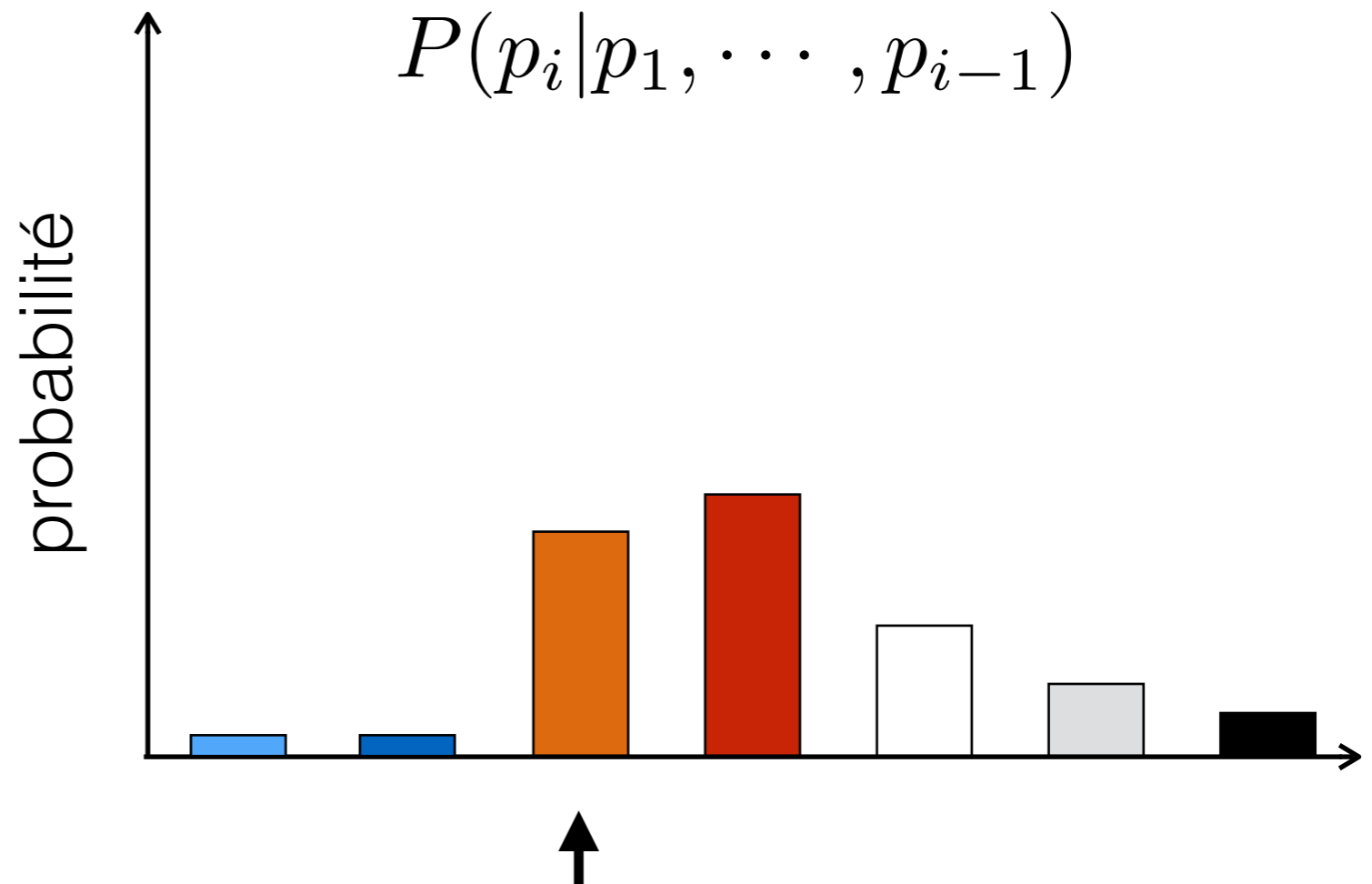


# Échantillonnage

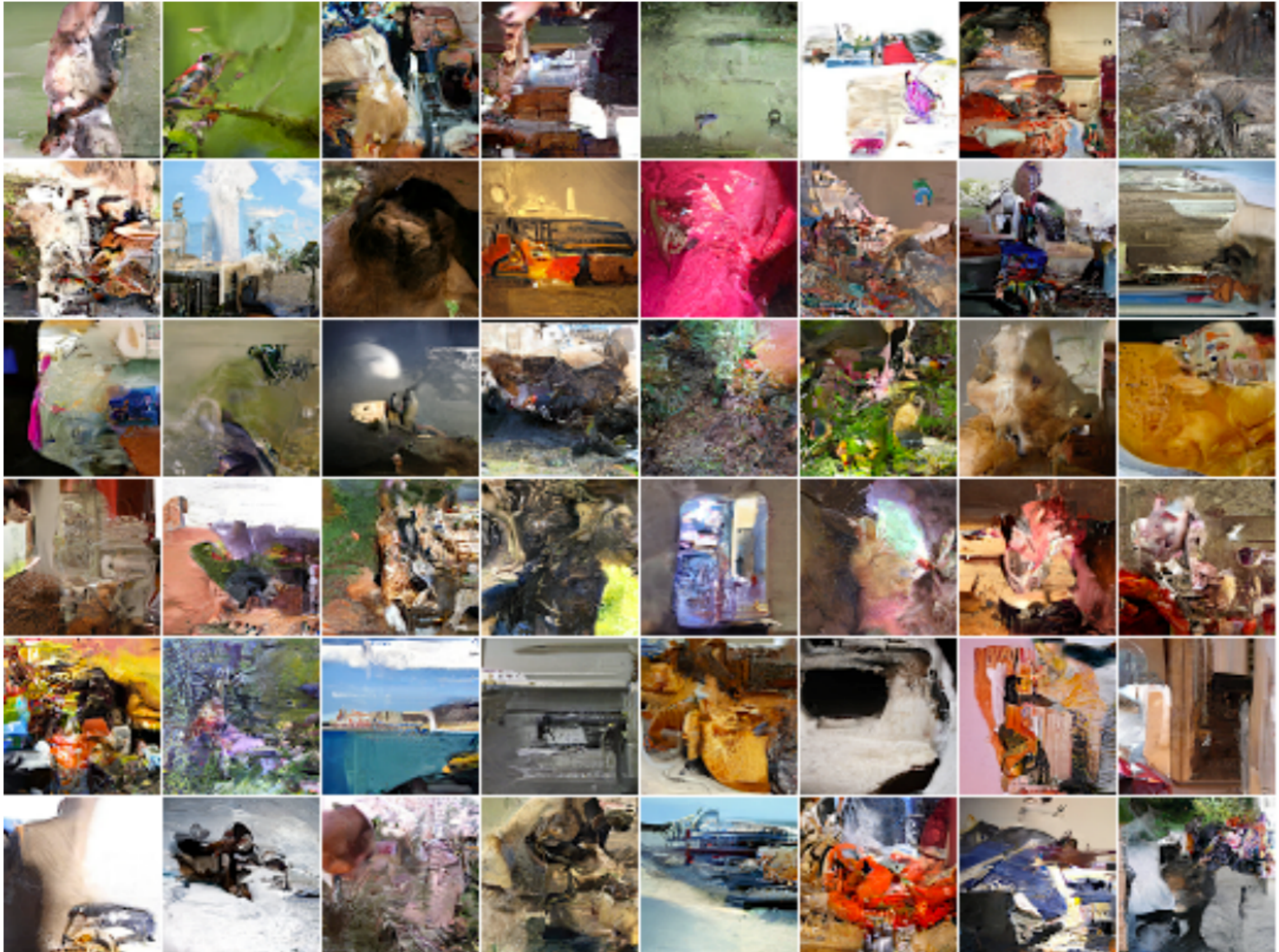
Sortie du réseau



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



# Générer des images entières



# Compléter une image

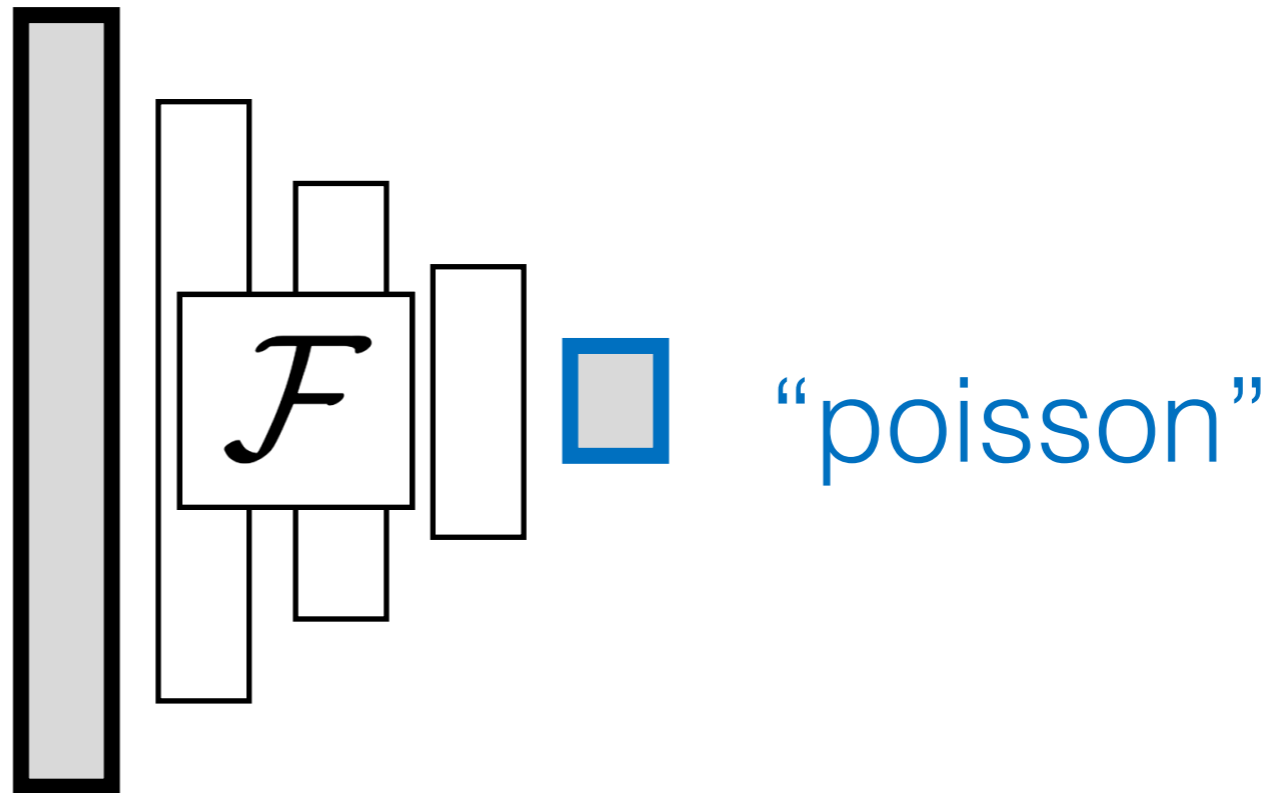
occlusion

résultats

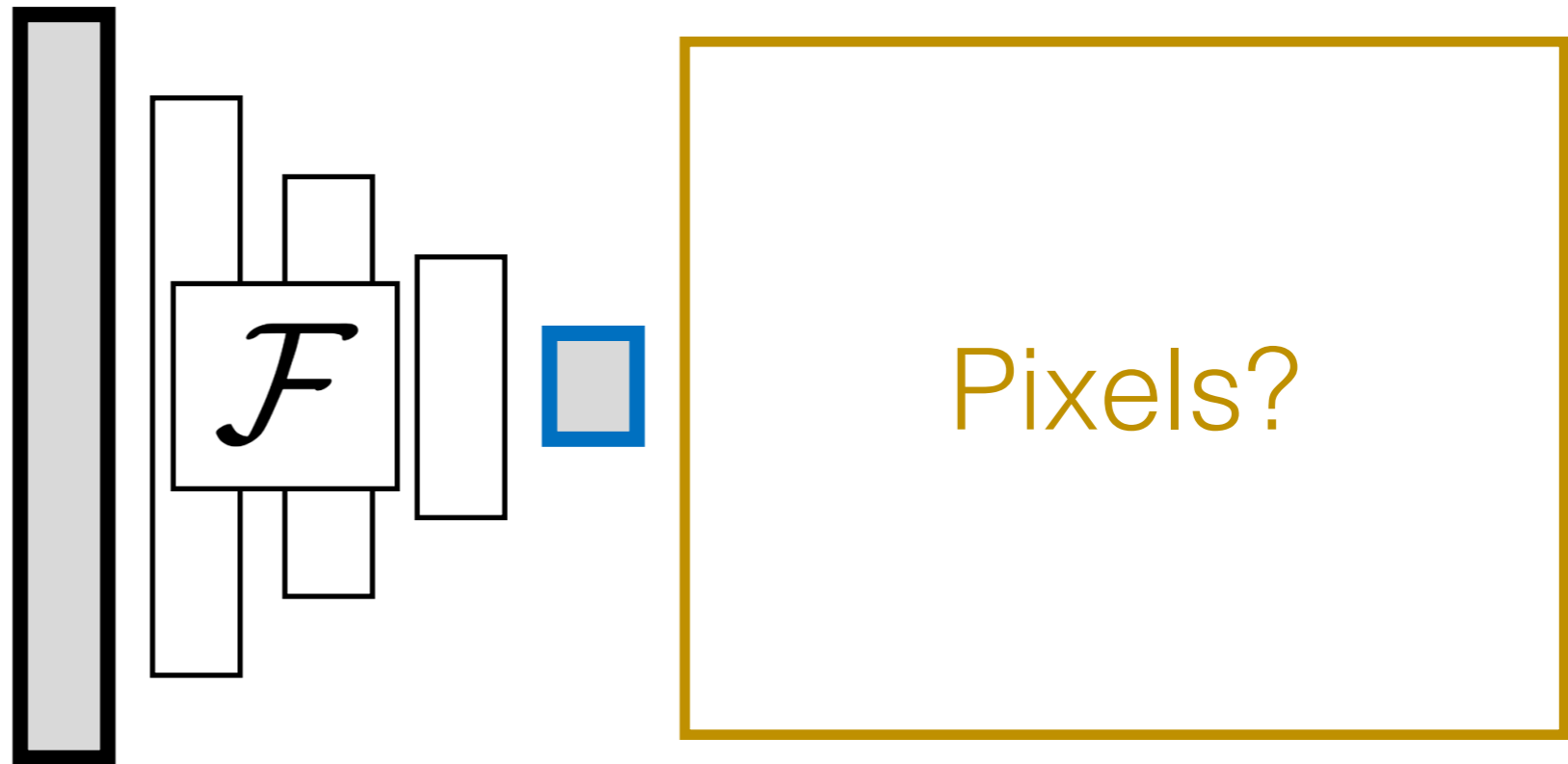
image  
originale



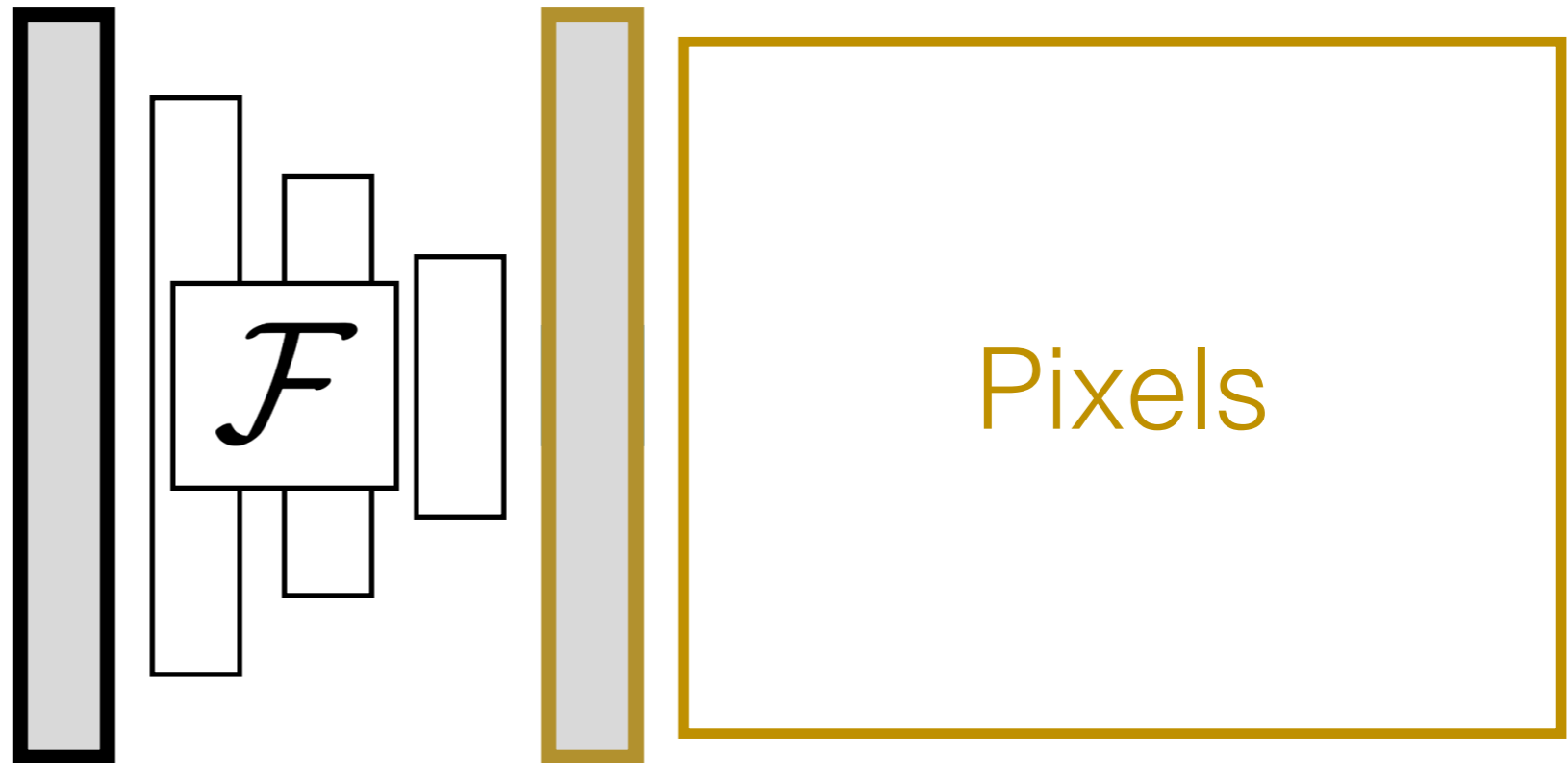
# Réseaux profonds discriminatifs



# Réseaux profonds discriminatifs



# Réseaux profonds *génératifs*







Ansel Adams, Yosemite Valley Bridge



[Zhang et al. 2016]

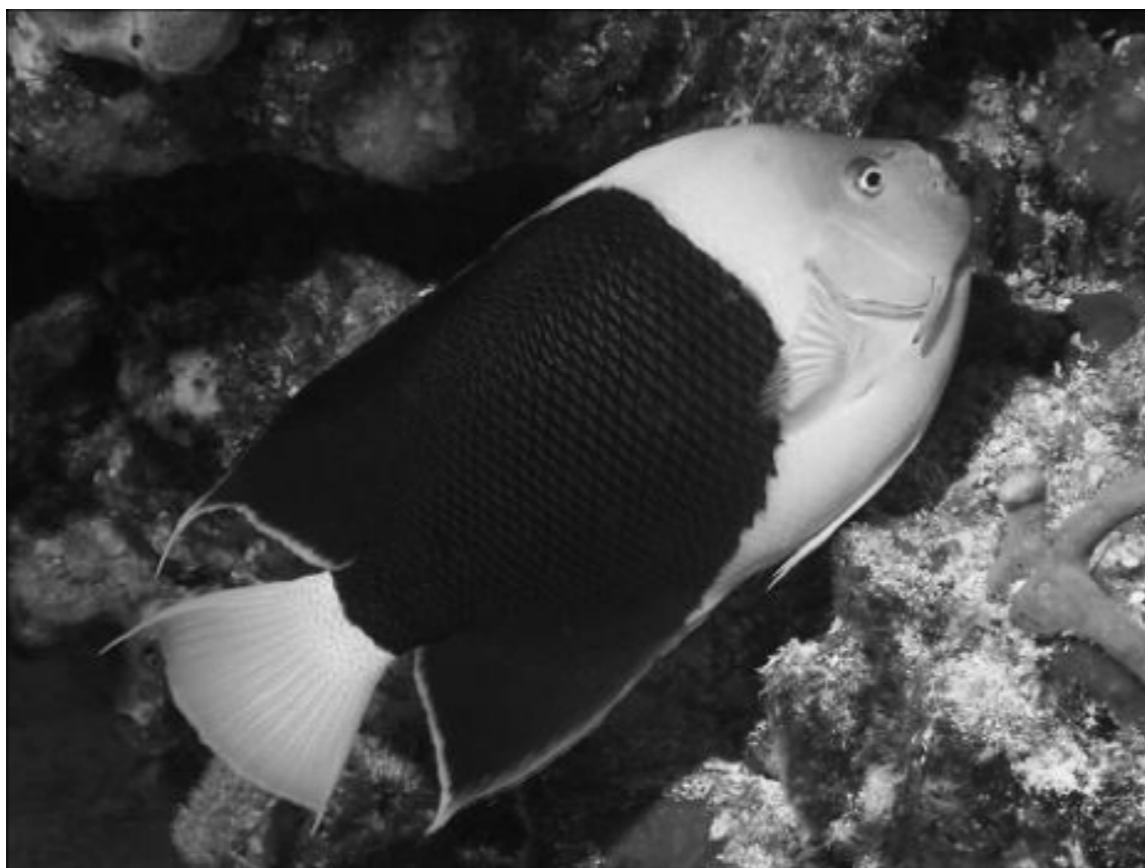
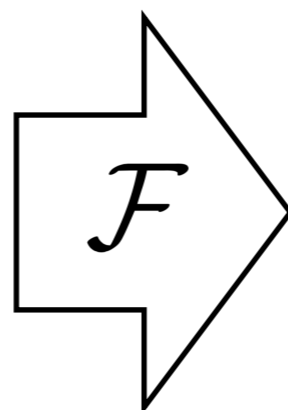


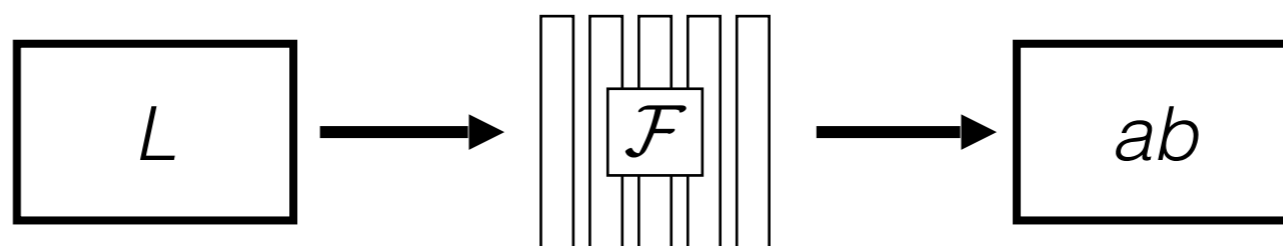
Image d'entrée: canal L seulement

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



Information en couleurs: canaux ab (de Lab)

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



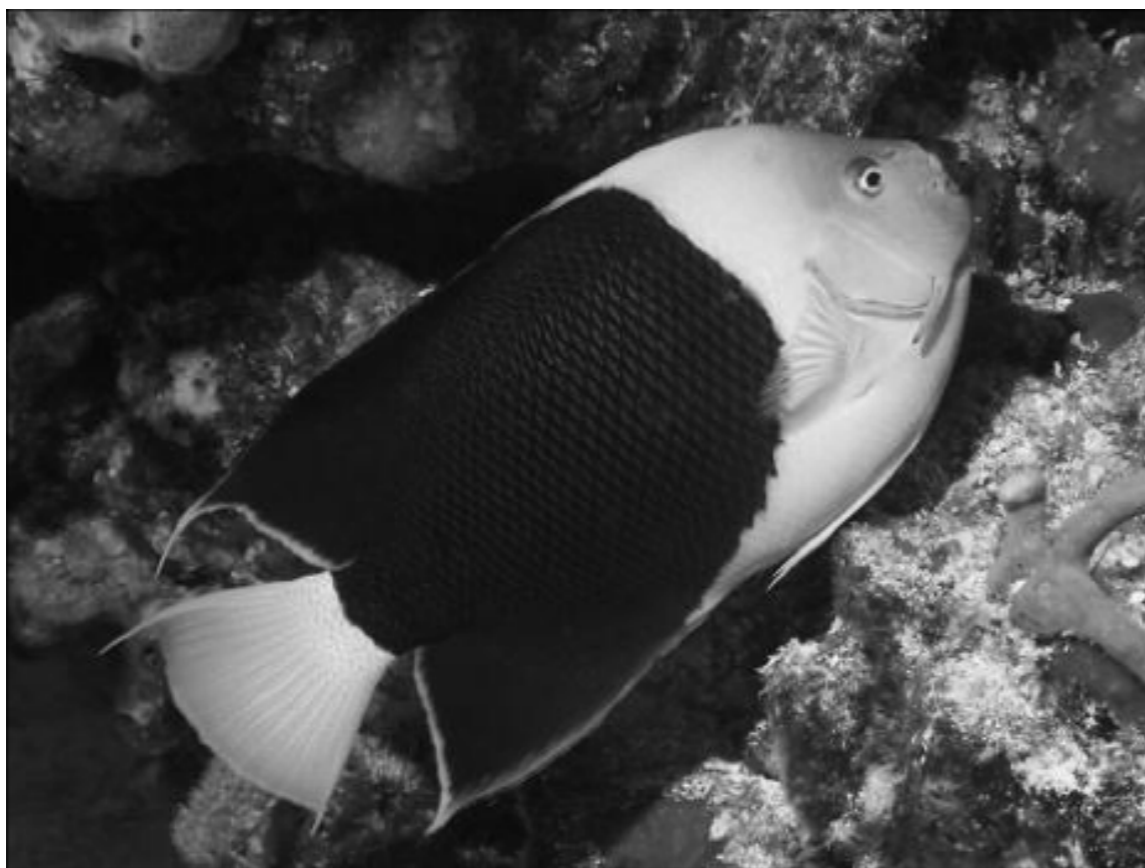
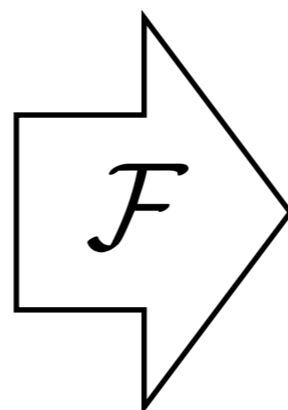


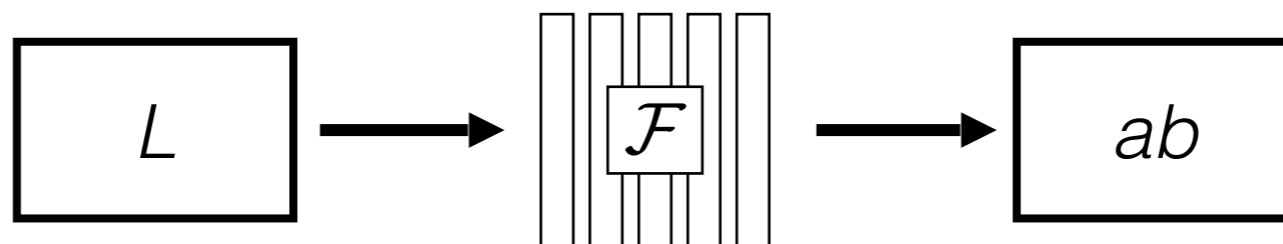
Image: canal L (de Lab)

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



Concaténation (L,ab), conversion vers RGB

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



Malheureusement, minimiser la somme des différences au carré ne fonctionne pas! 😞

Entrée



Sortie



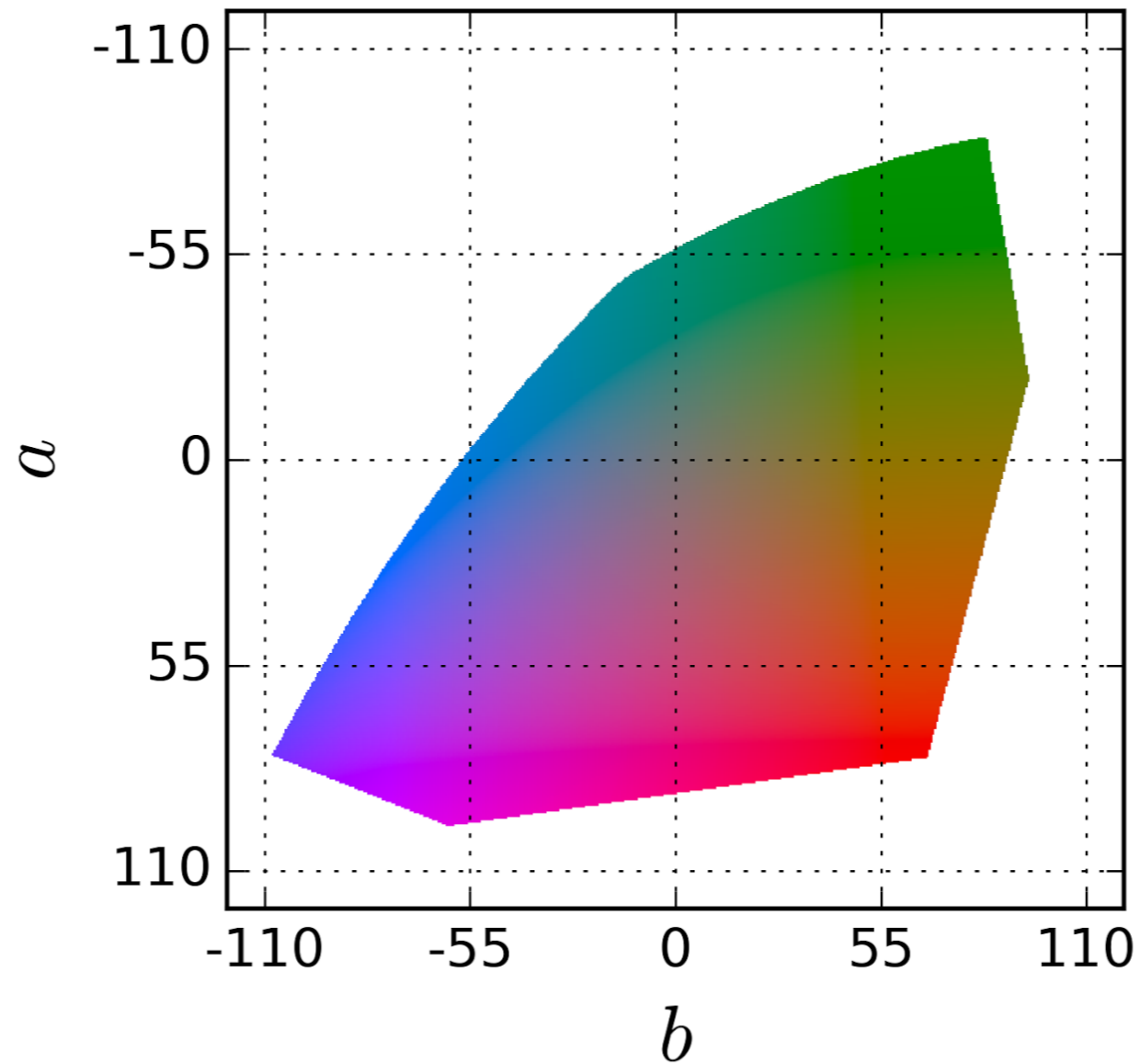
Vérité



$$L_2(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \hat{\mathbf{Y}}_{h,w}\|_2^2$$

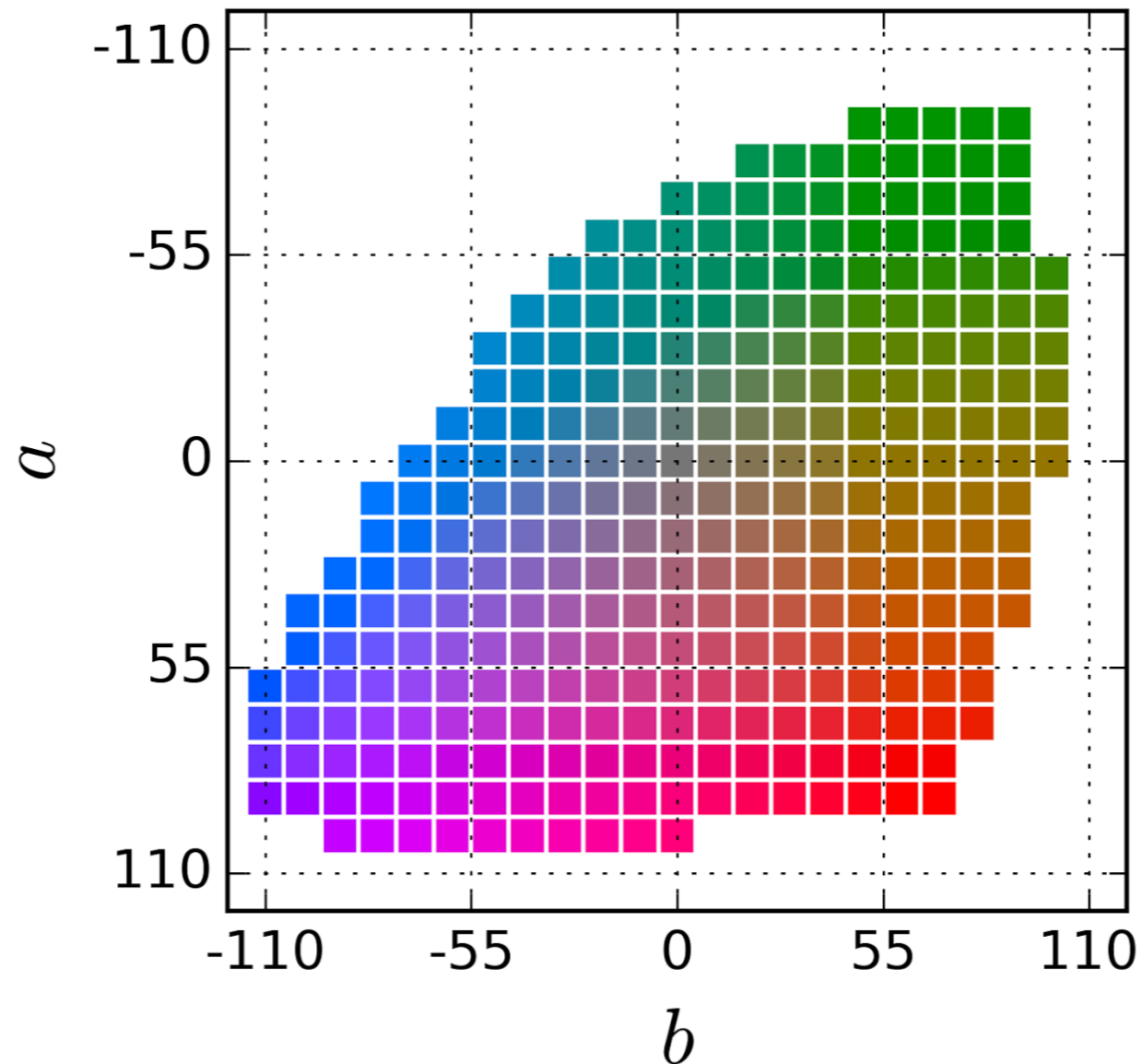
# Meilleure fonction de perte

Espace (continu) des couleurs ab



# Meilleure fonction de perte

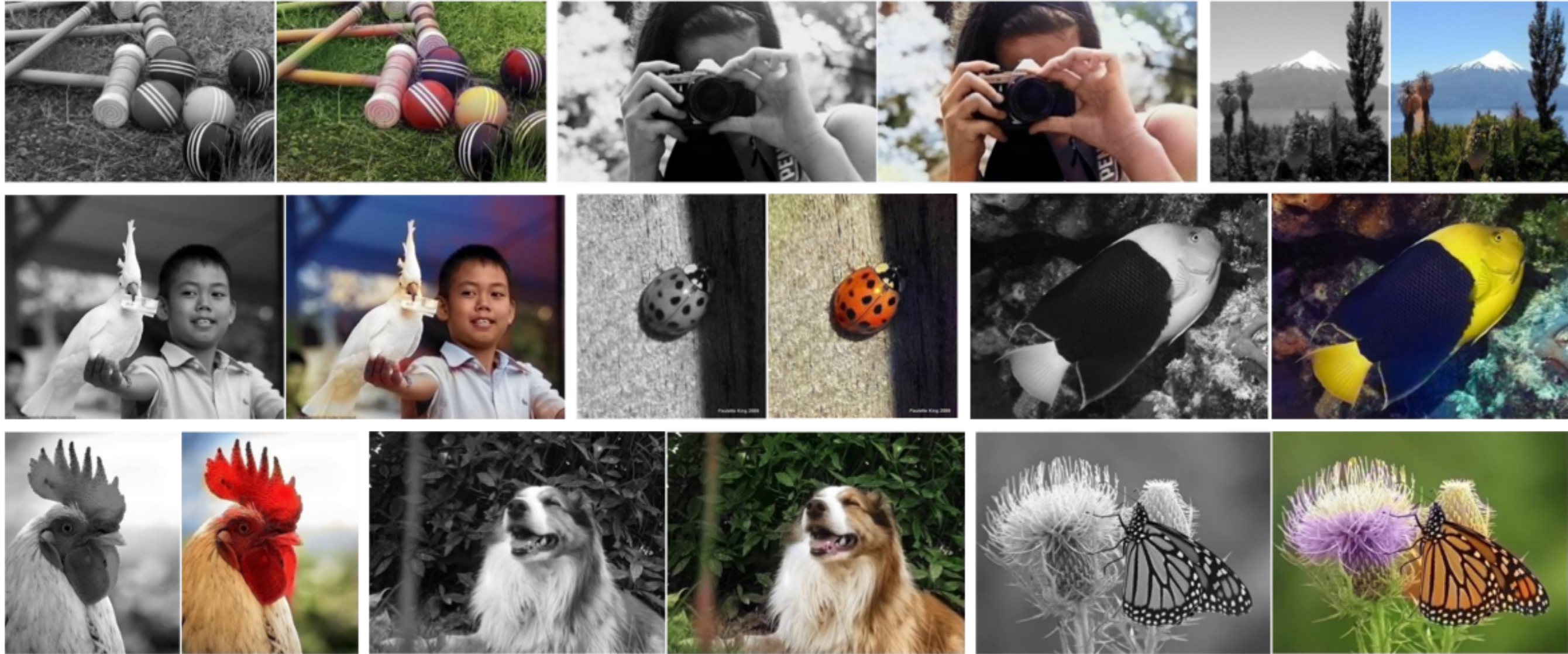
Discrétisation de l'espace de couleurs ab



Prédire la *distribution* sur l'espace  $ab$ !

Permet de modéliser les distributions multimodales (plusieurs couleurs sont plausibles)

# Bons résultats

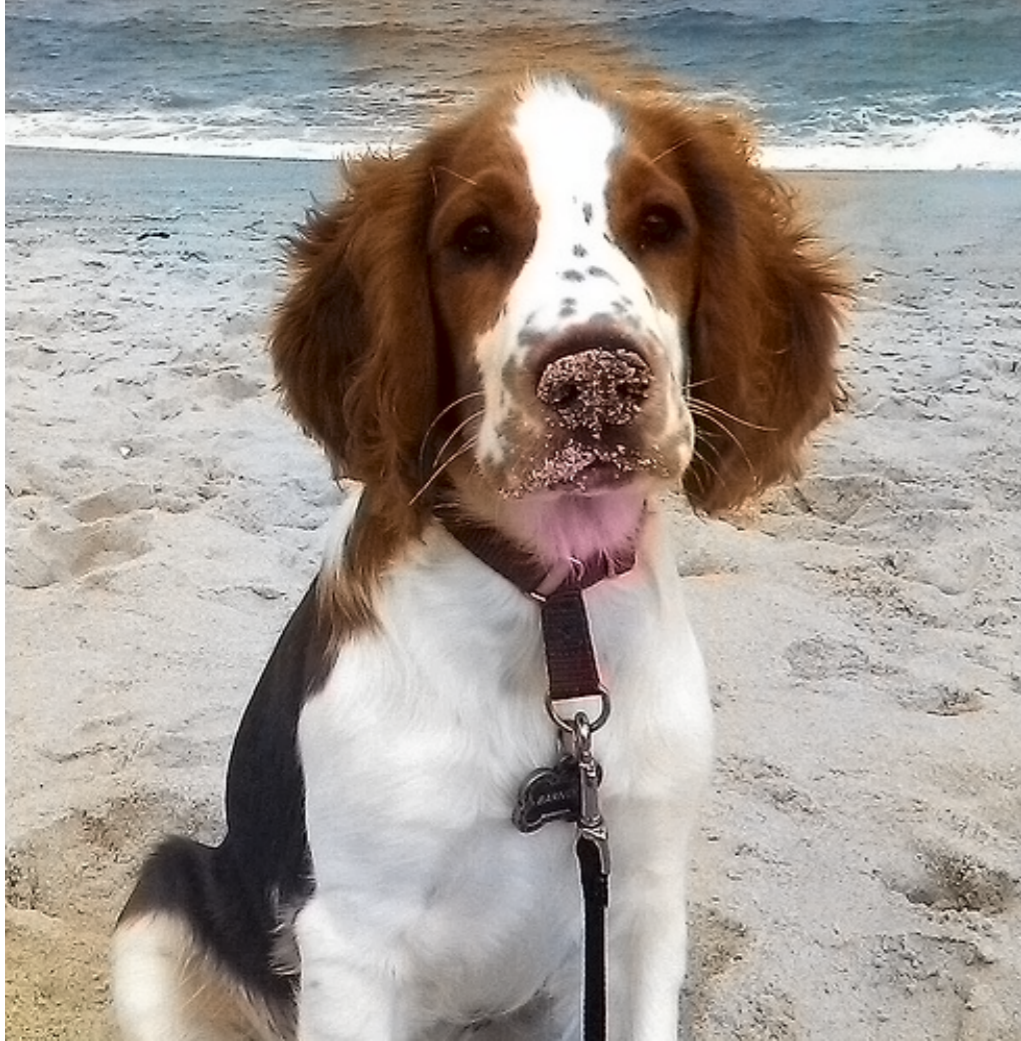




# Moins bons résultats



# Biais





Reddit /u/SherySantucci



Reddit ColorizeBot

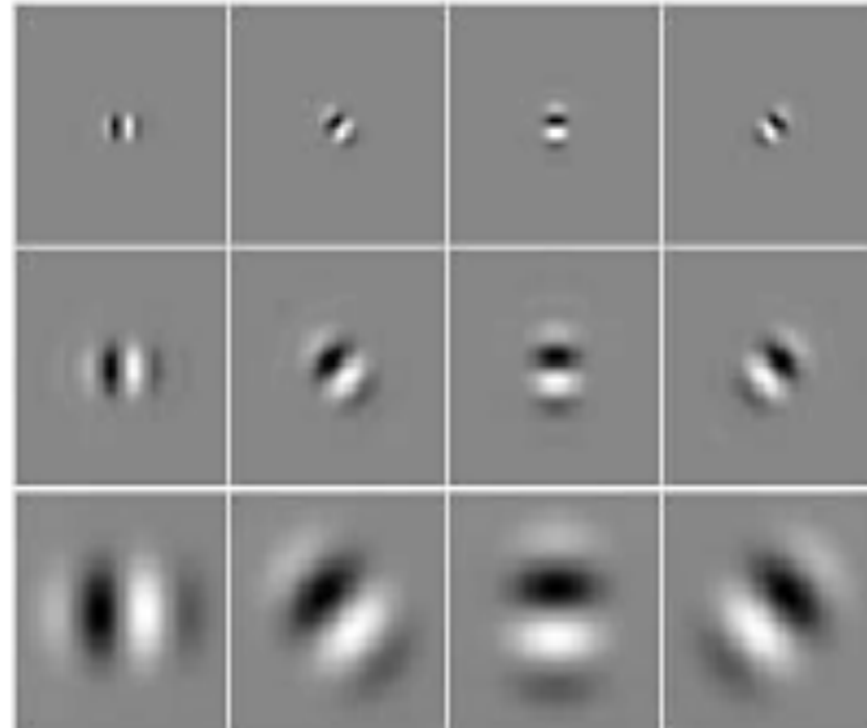


Photo de Reddit /u/Timteroo,  
Murale de Eduardo Kobra



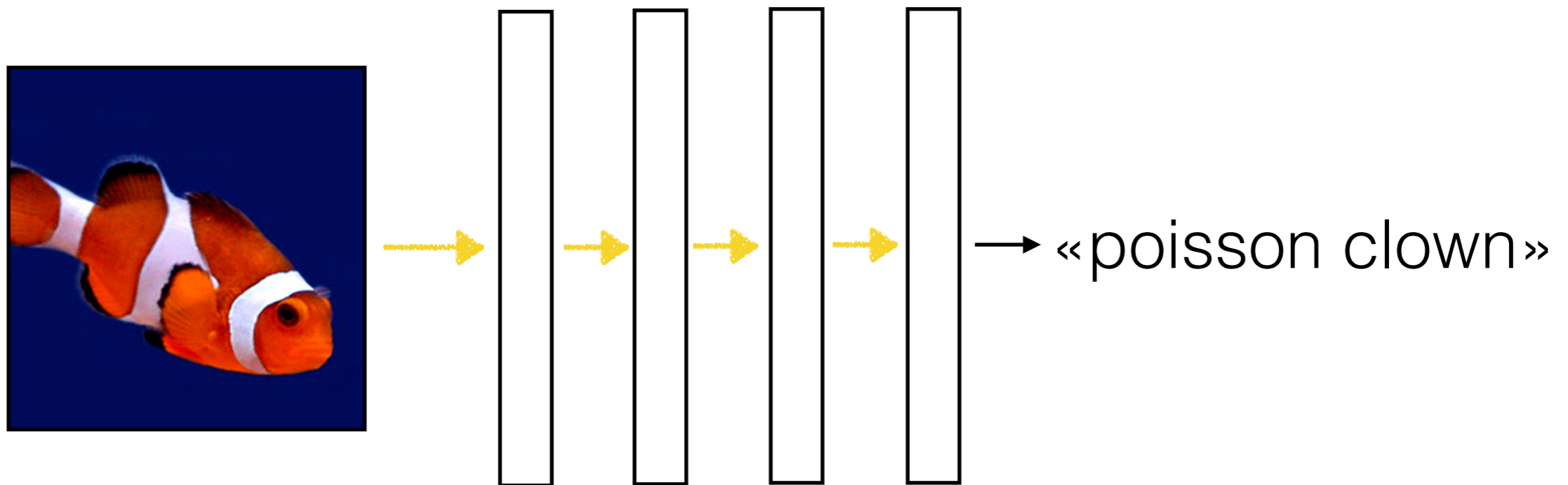
Reddit ColorizeBot

# Représenter les textures



Pourquoi *ces* caractéristiques exactement?

# Choix de la représentation

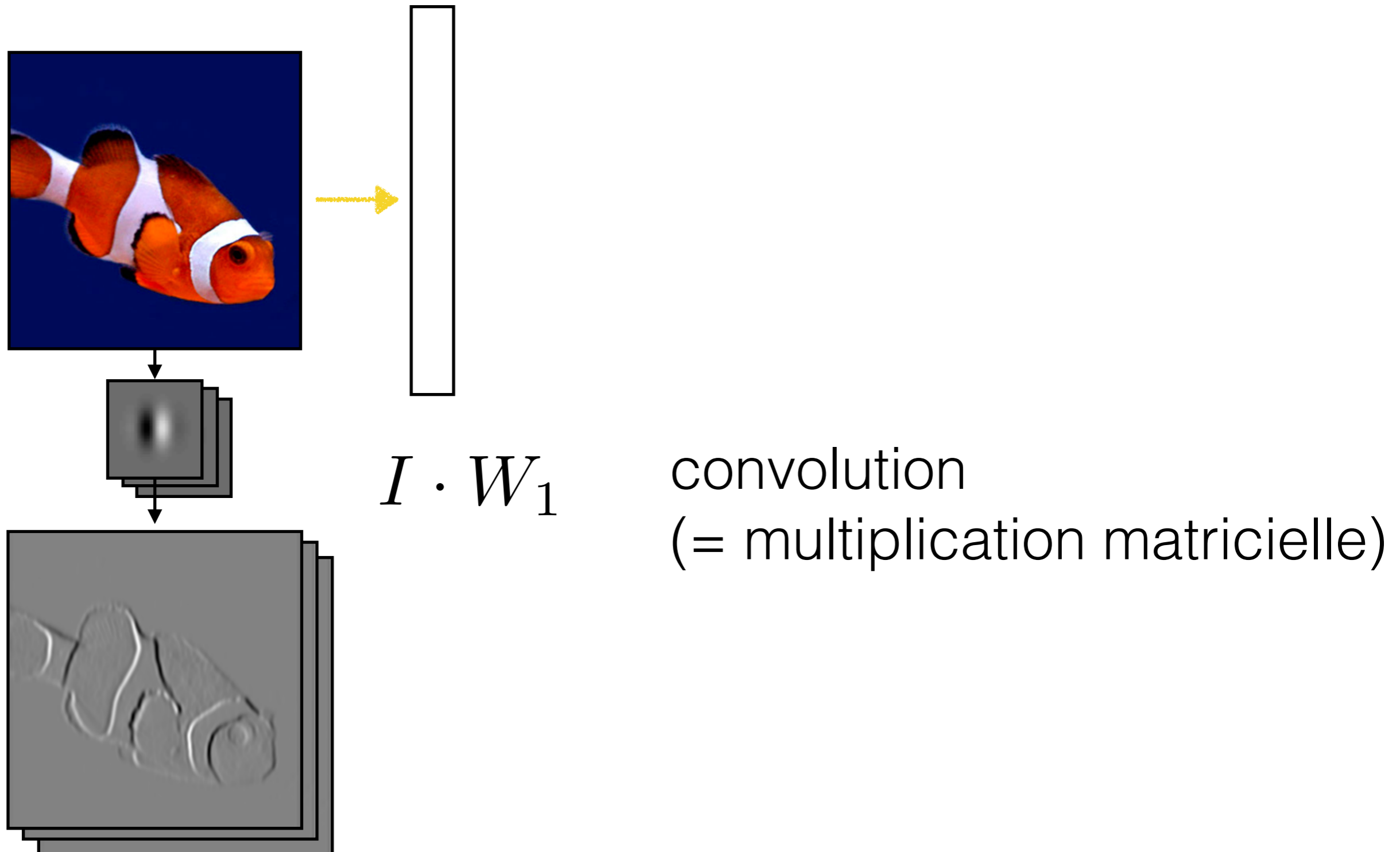


Au lieu de banques de filtres pré-déterminés, utilisons les caractéristiques apprises par un réseau de neurones!

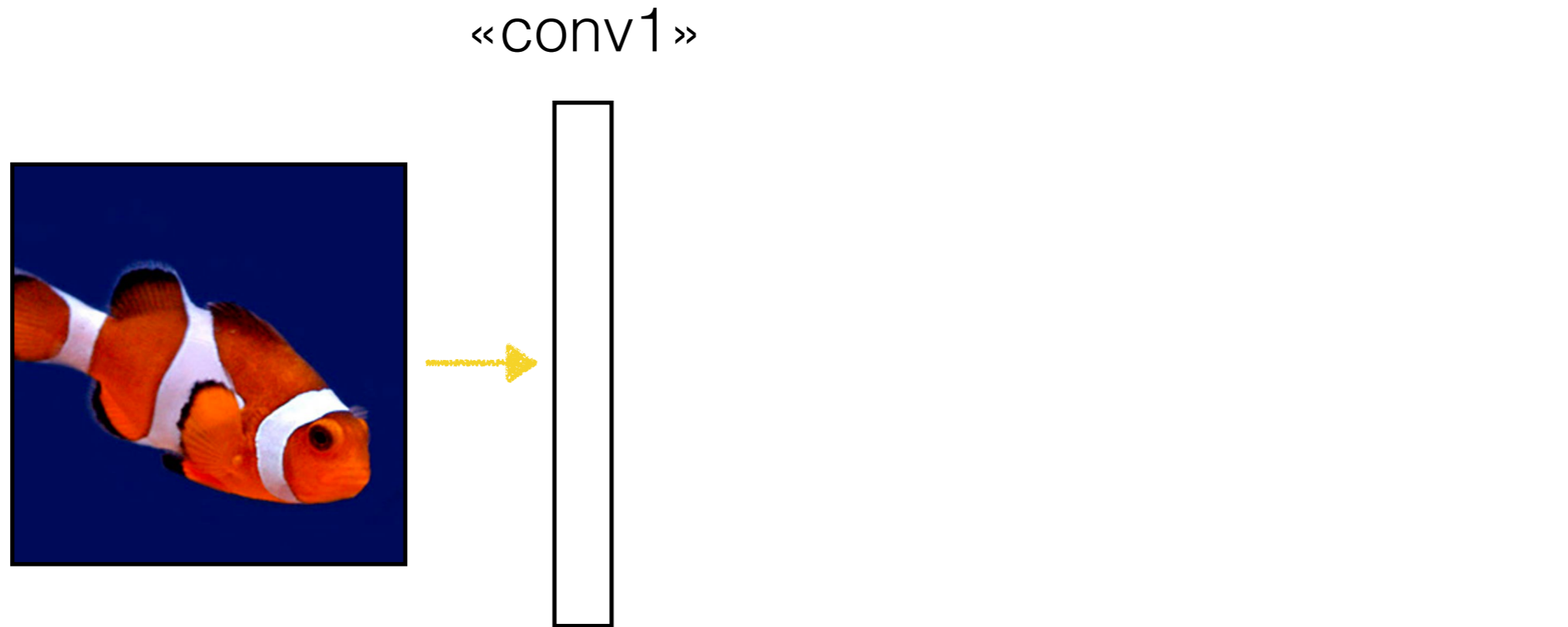
Un réseau de neurone entraîné à reconnaître les objets devient un calculateur de caractéristiques très robustes.



# Extraire les caractéristiques d'une image



# Extraire les caractéristiques d'une image

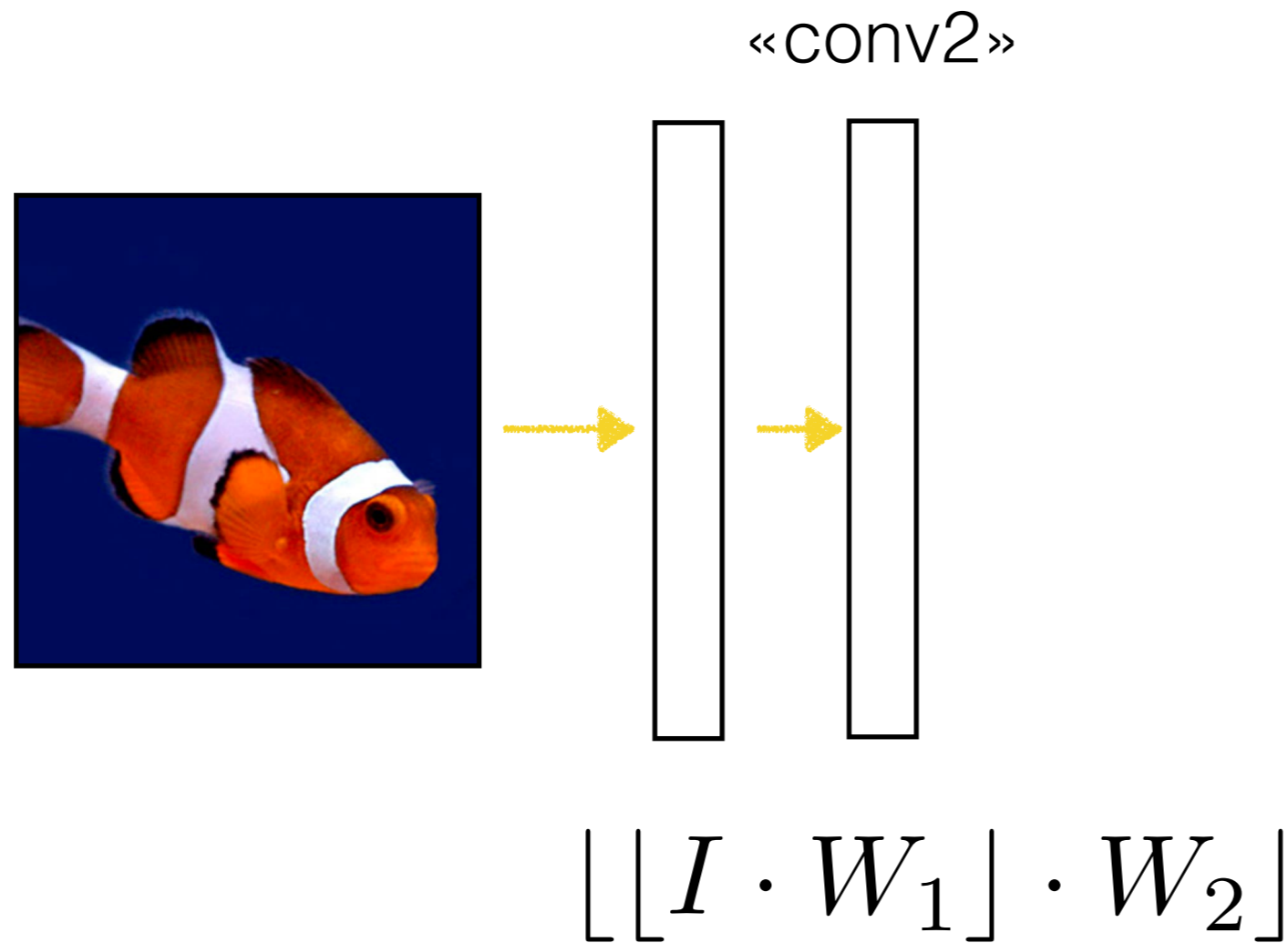


$[I \cdot W_1]$  convolution suivie d'une non-linéarité

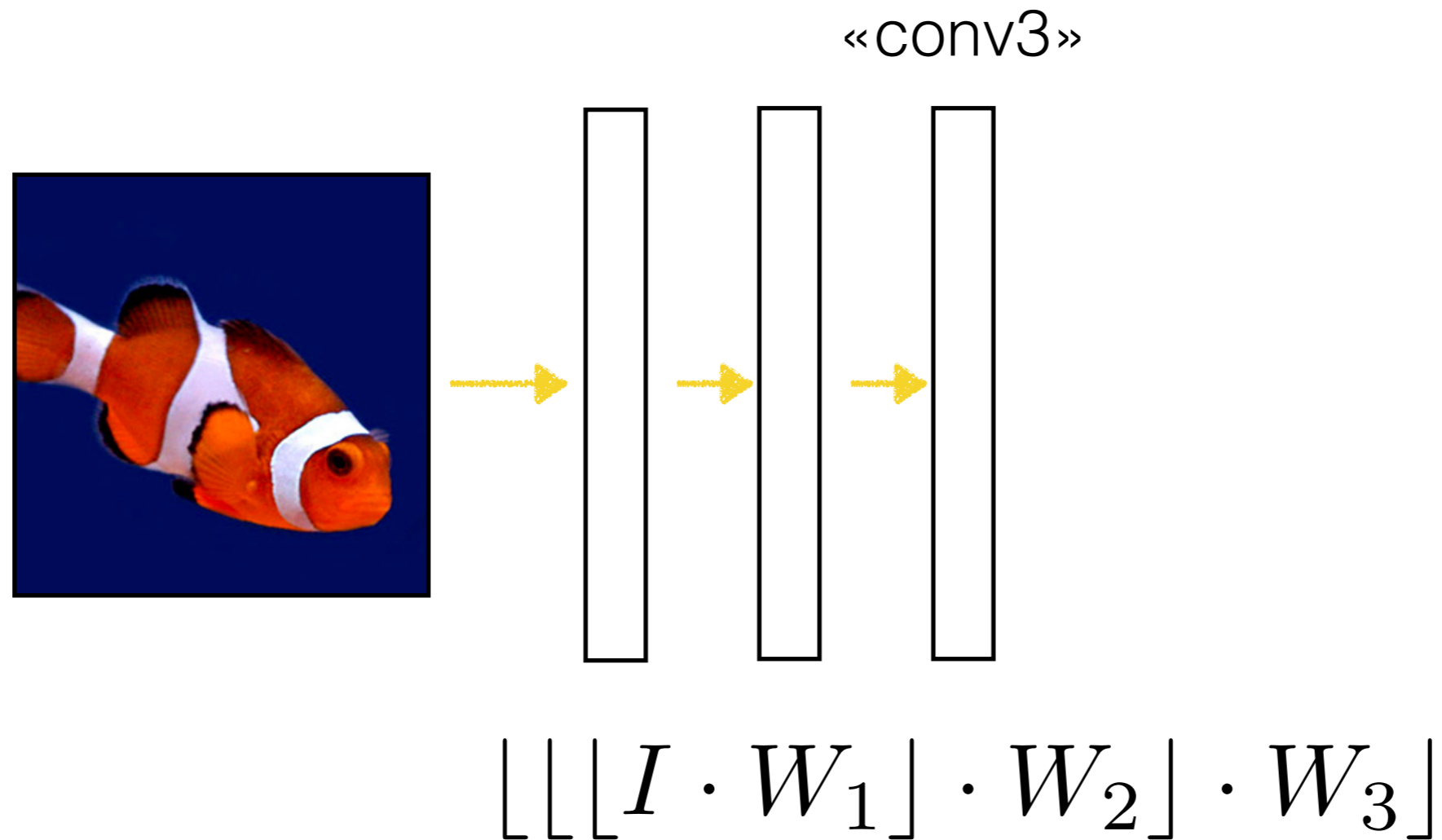
où  $[\vec{x}]_i = \max(x_i, 0)$

i.e. valeurs négatives = 0

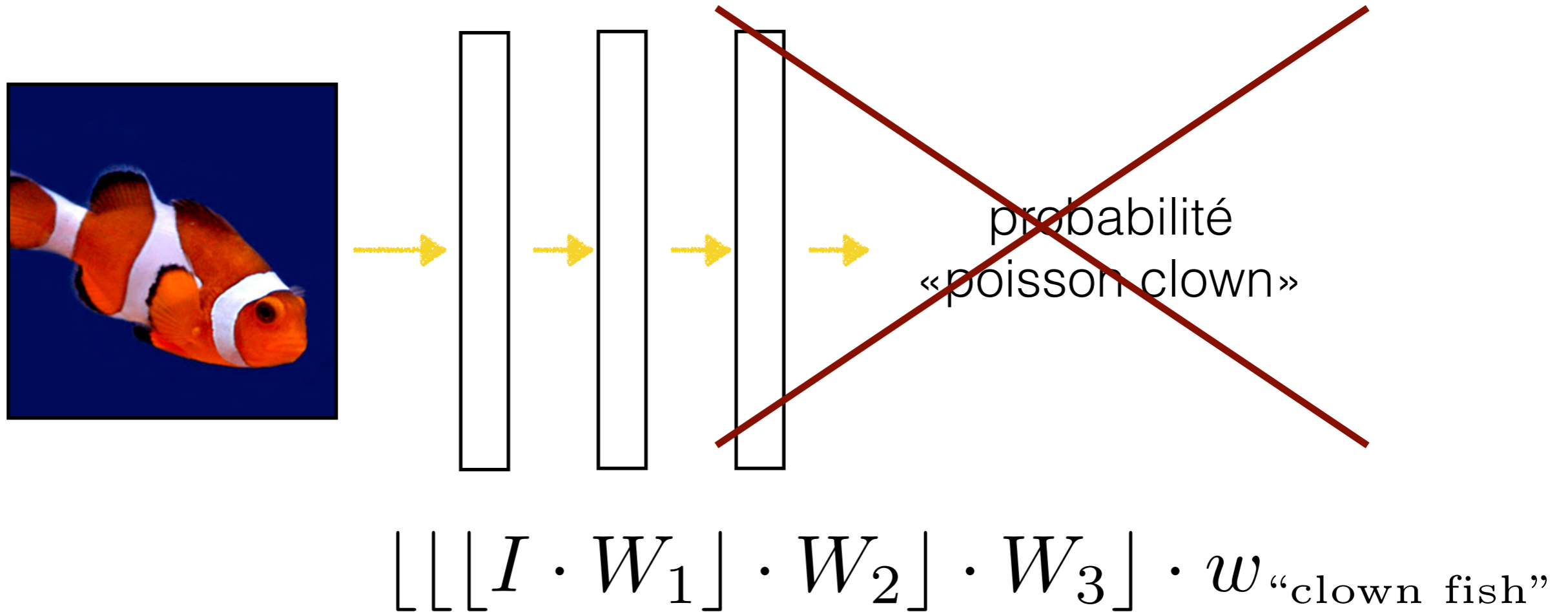
# Extraire les caractéristiques d'une image



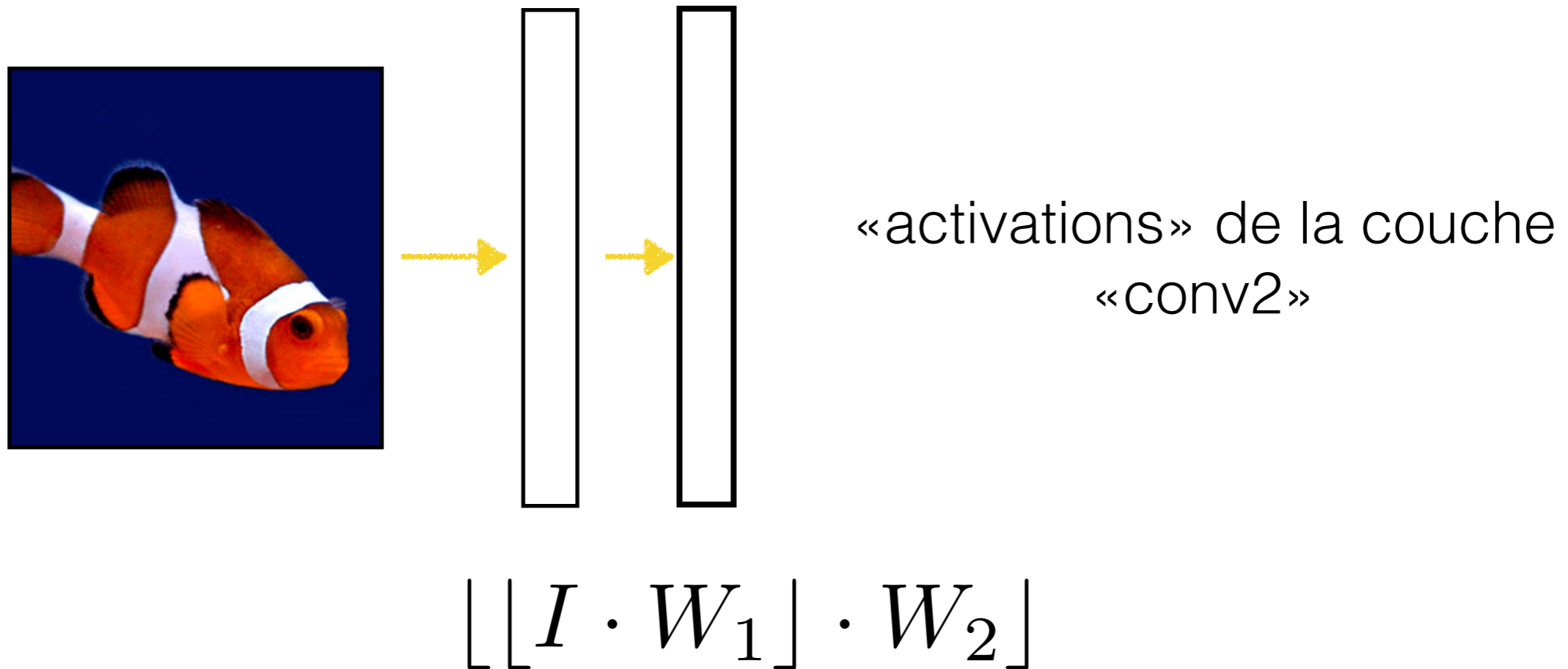
# Extraire les caractéristiques d'une image



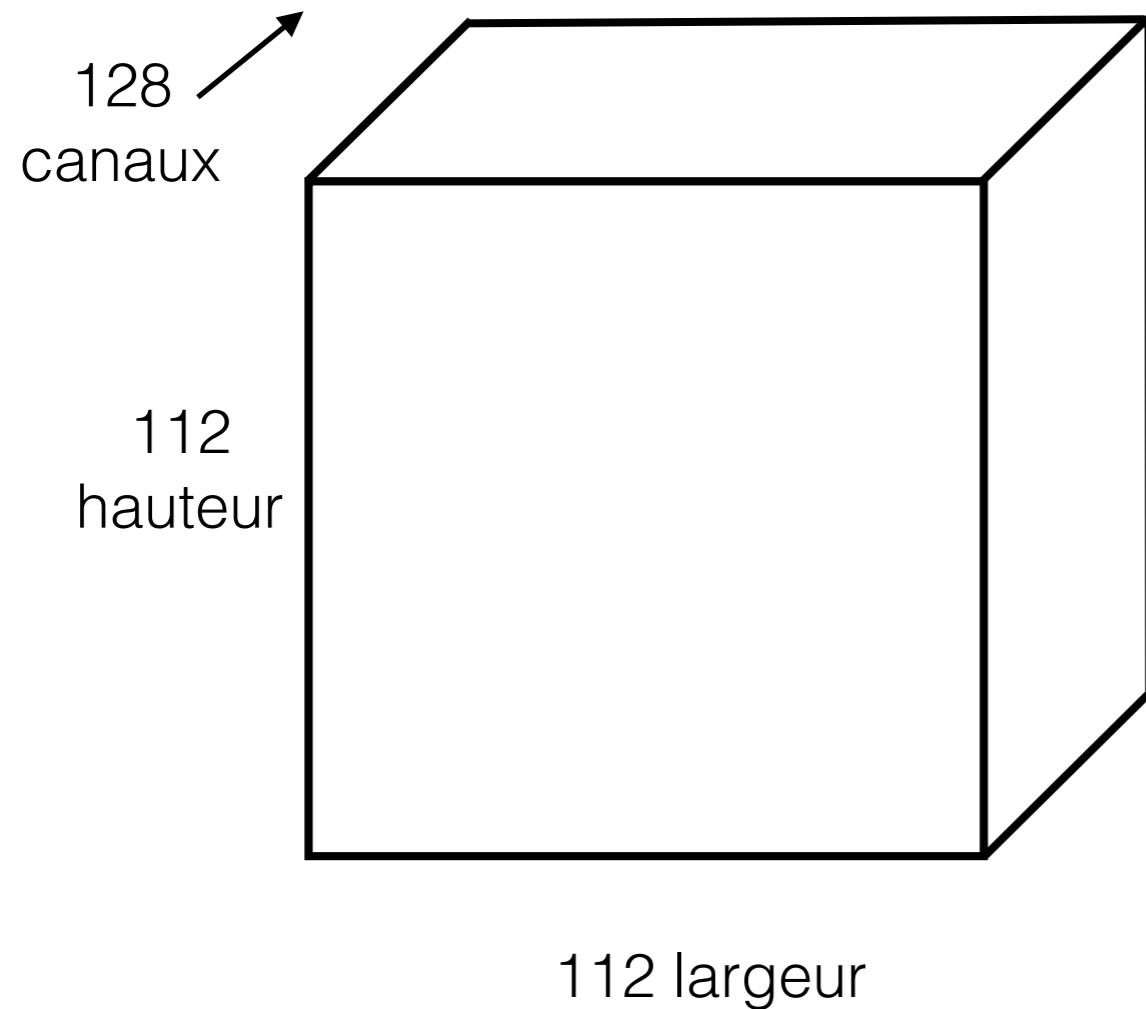
# Extraire les caractéristiques d'une image



# Extraire les caractéristiques d'une image



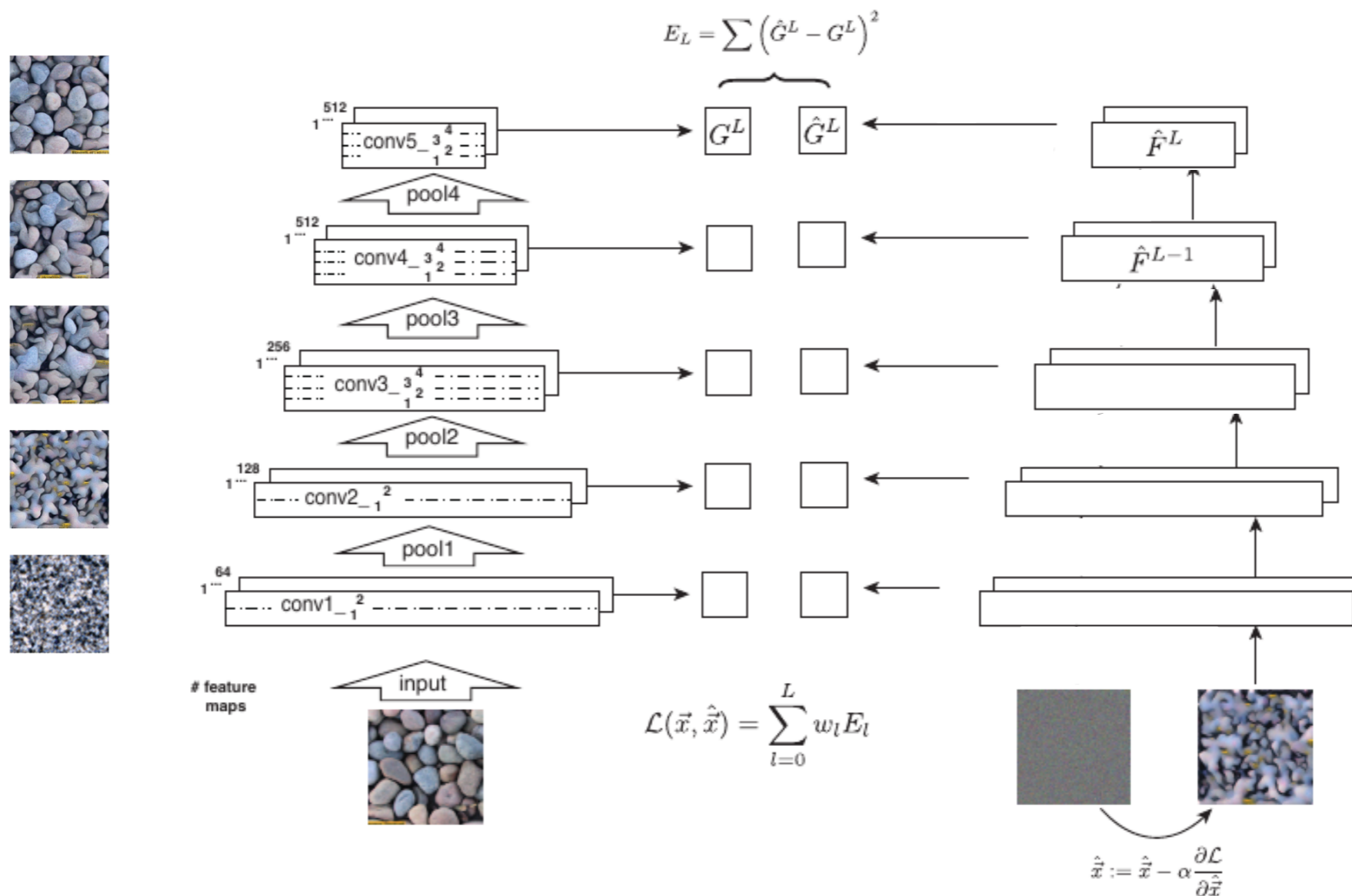
# Extraire les caractéristiques d'une image



activations de «conv2»

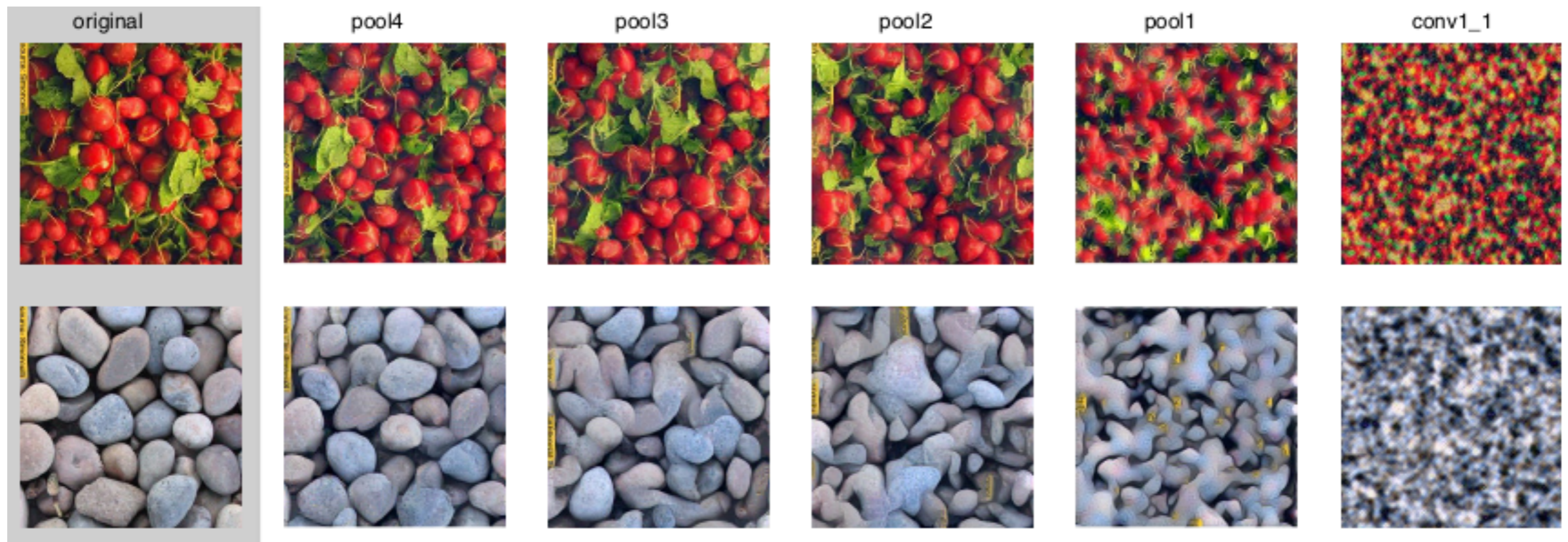
e.g. (112 x 112) x 128 pour conv2  
d'un modèle populaire (VGG19)

# Synthèse de textures par CNN





# Synthèse de textures par CNN



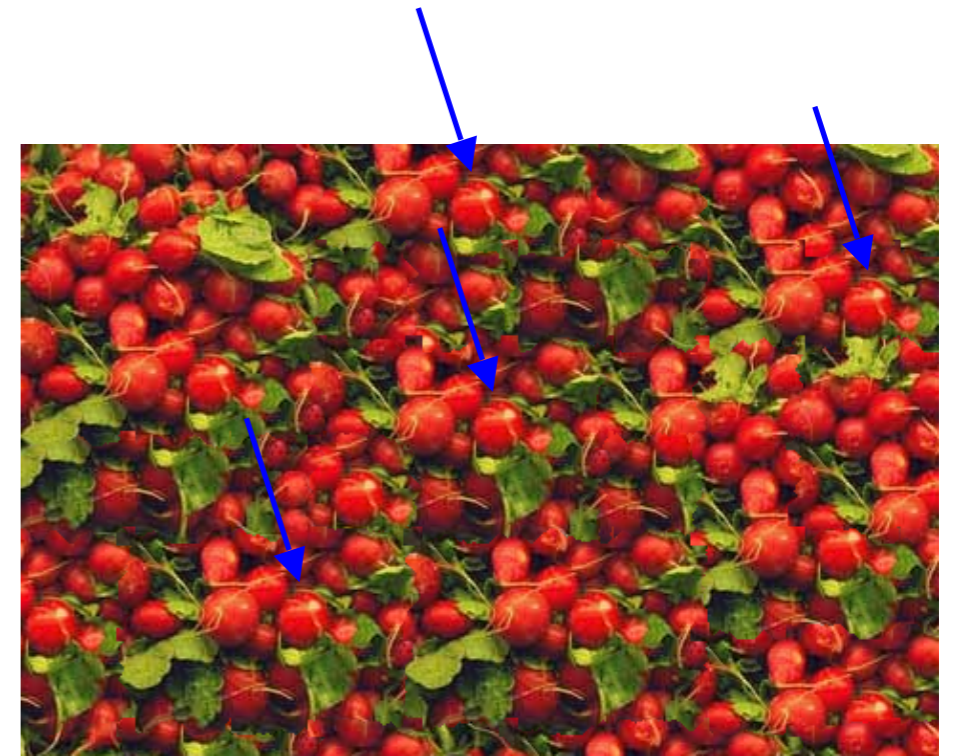
Haute  → Basse

# Statistiques de textures



# Synthèse de textures par CNN

Échantillonner la texture originale



# Synthèse de textures par CNN

Synthesis



Source

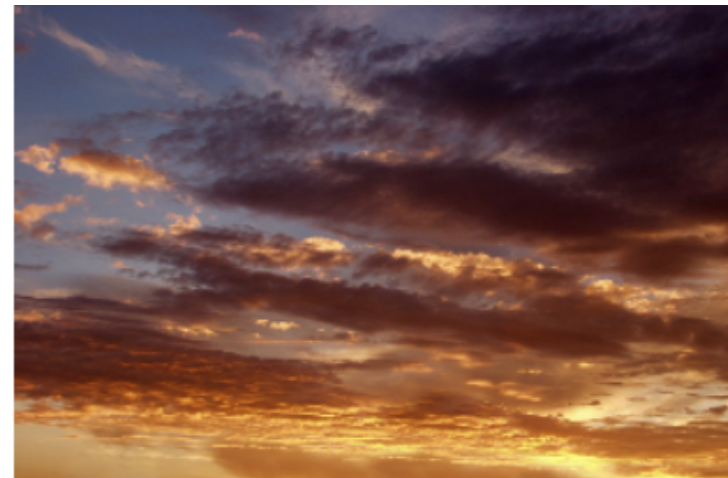
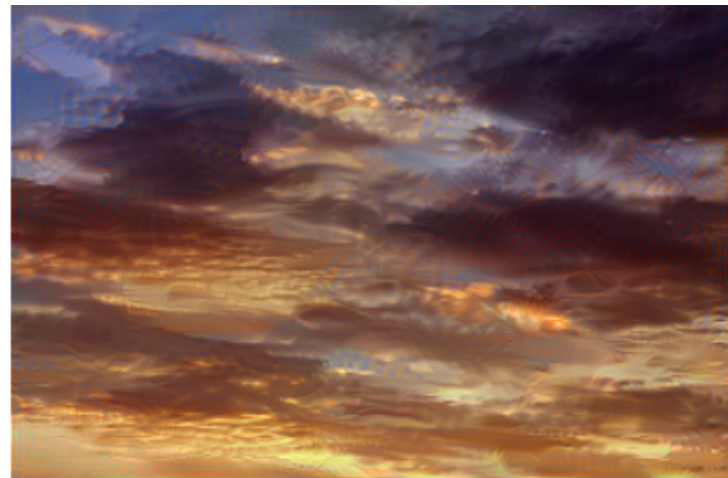


# Synthèse de textures par CNN

Synthesis



Source



# Capturer le style artistique

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



# Capturer le style artistique

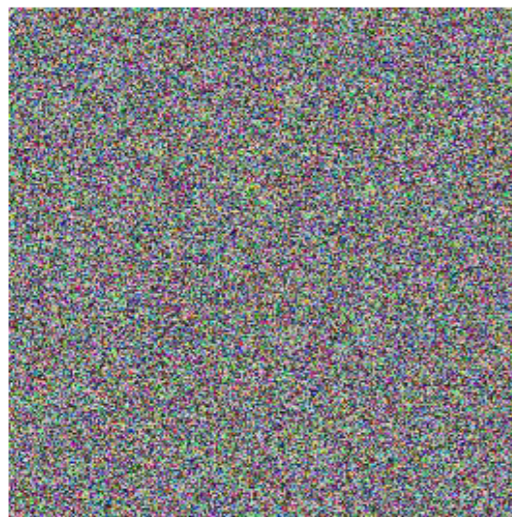
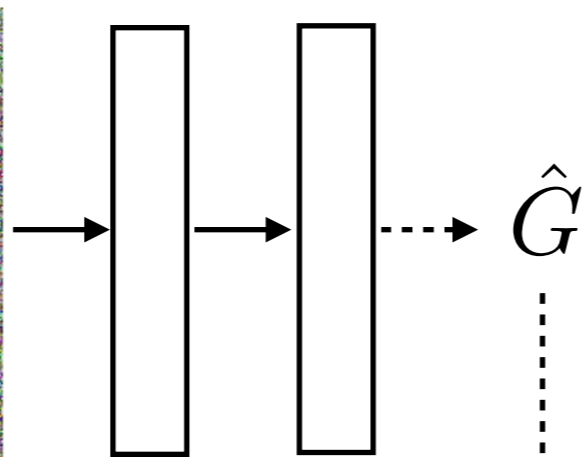


Image de synthèse



$\hat{G}$



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



Rajouter le contenu de la photo

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

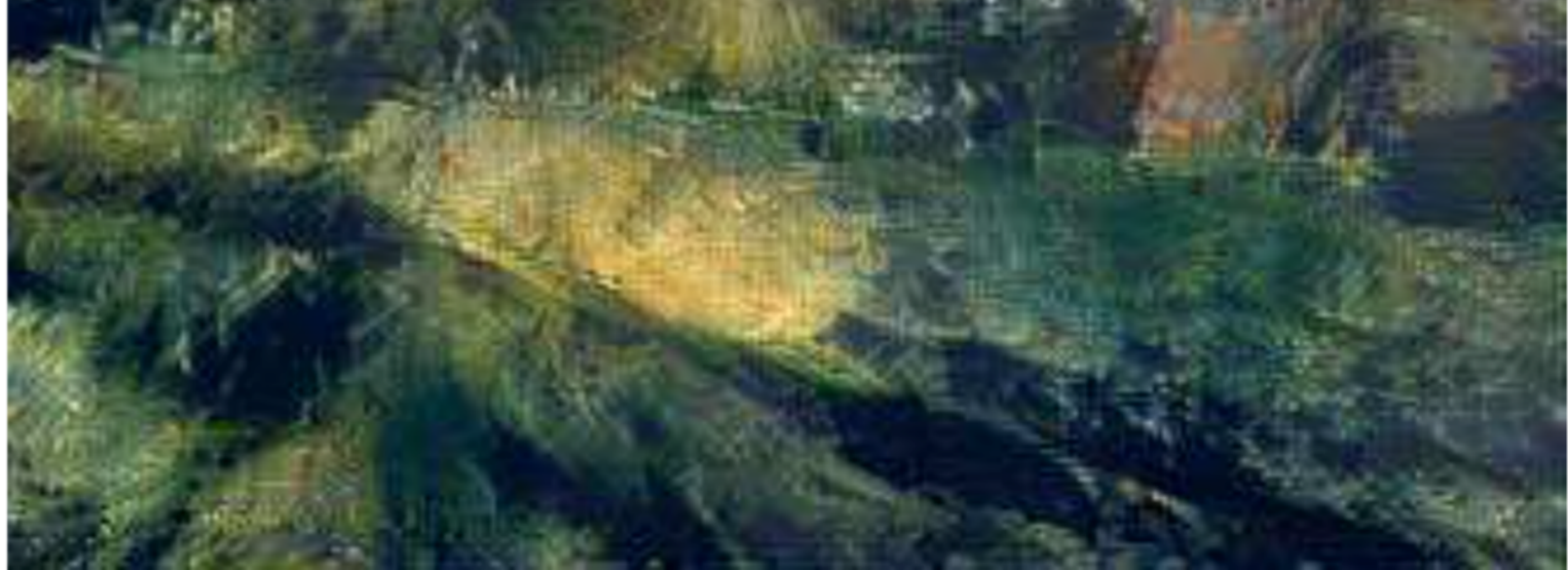
$$c_i(x,y)$$

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













Londres le jour



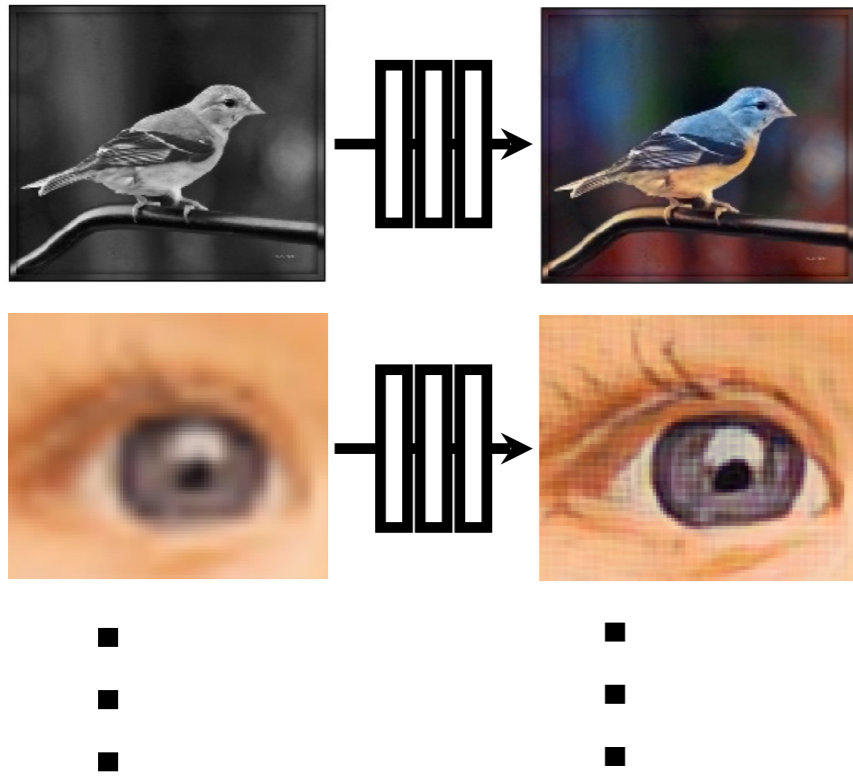
New York le soir

# Londres le soir?



# Revenons à la colorisation...

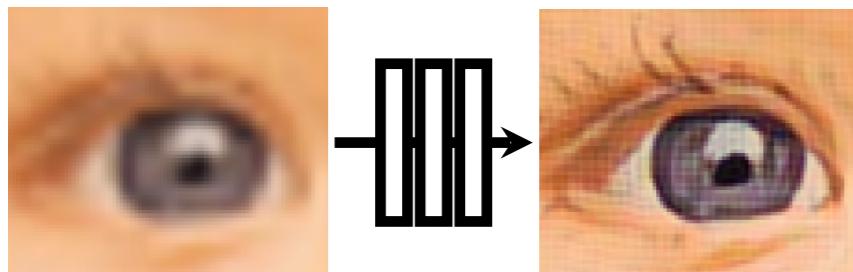
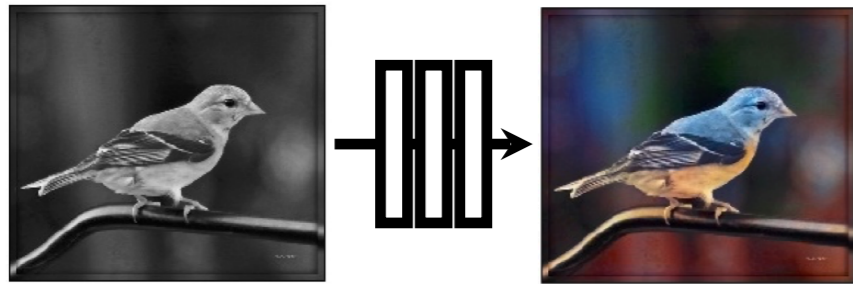




Quelle fonction de perte?



Image générées



■  
■  
■

■  
■  
■



Réseaux génératifs adversariaux  
Generative Adversarial Network  
(GANs)

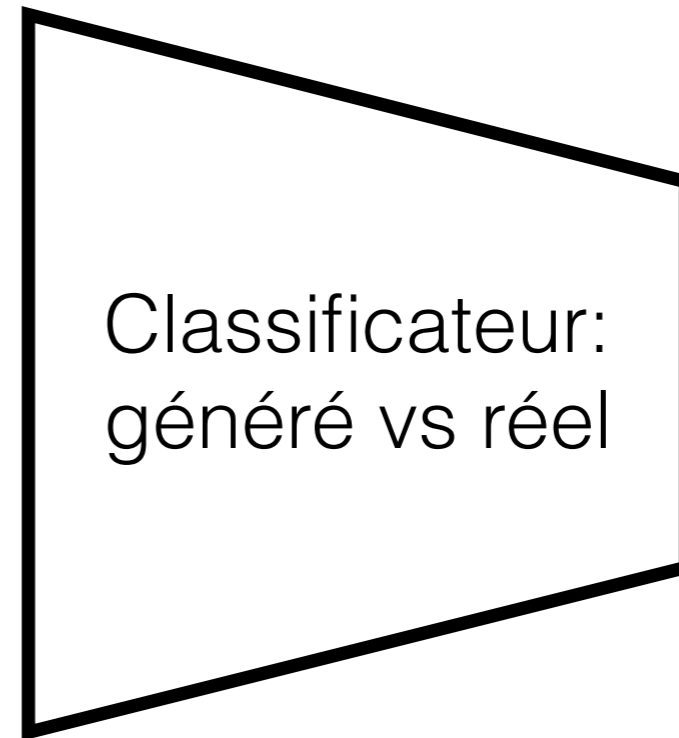
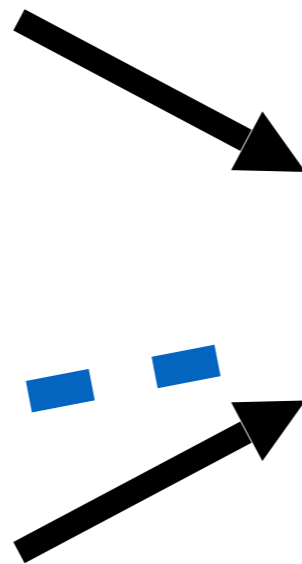


Image réelles



...

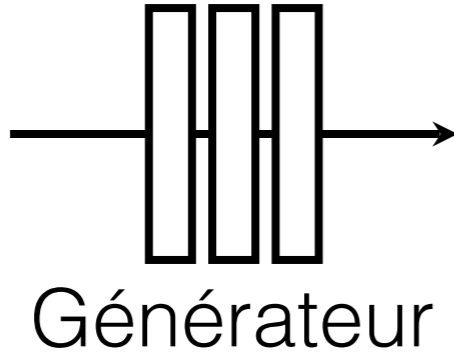


[Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, Bengio 2014]

$x$

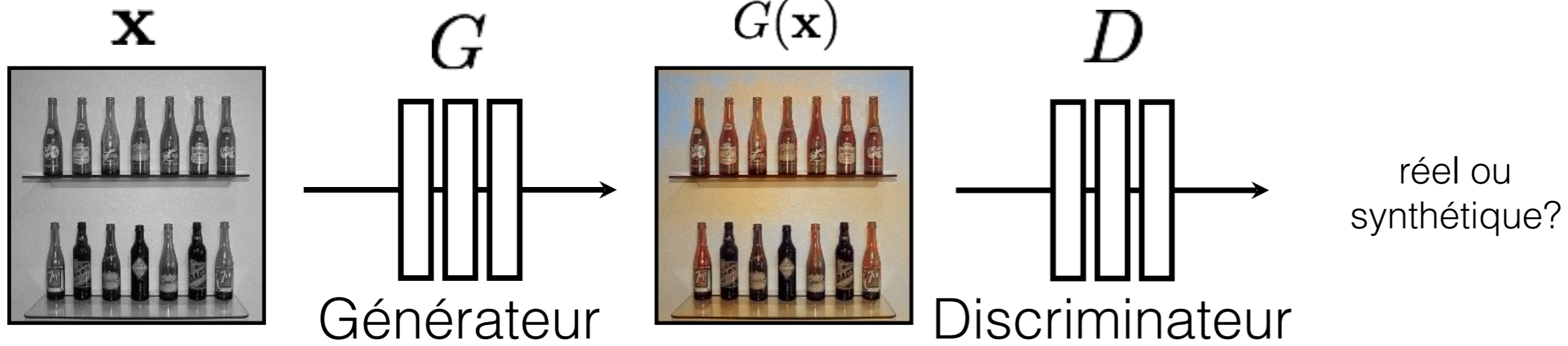


$G$



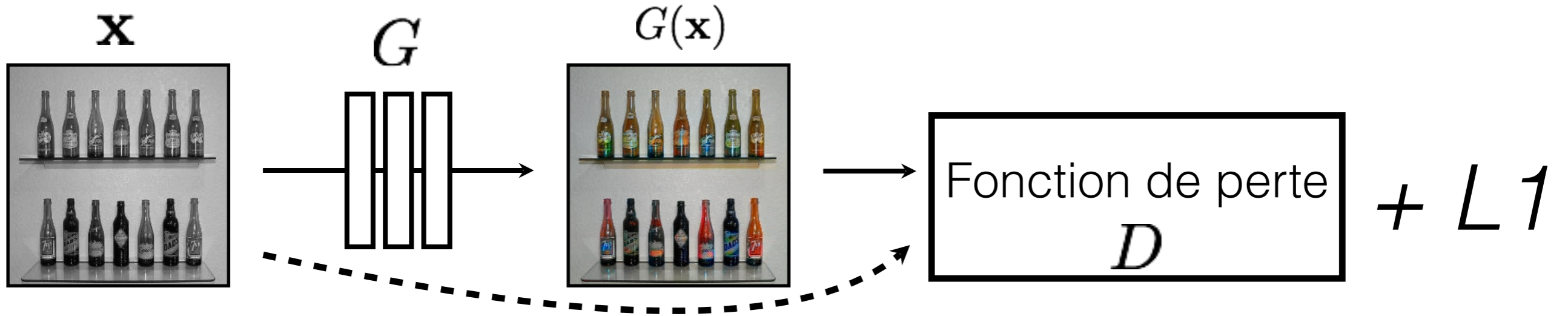
$G(x)$





**G** essaie de générer des images pour tromper **D**

**D** essaie d'identifier les images générées par **G**



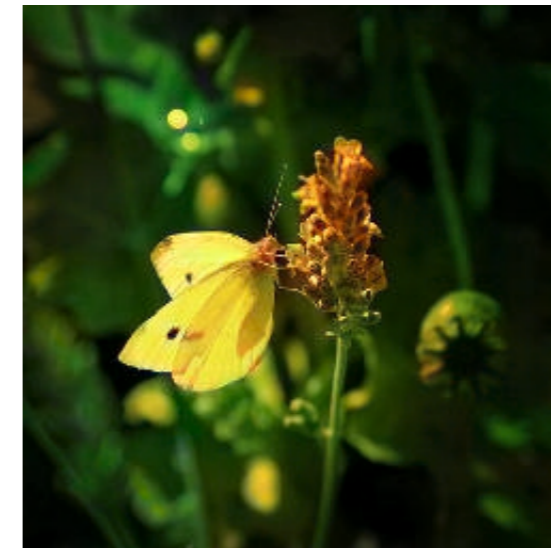
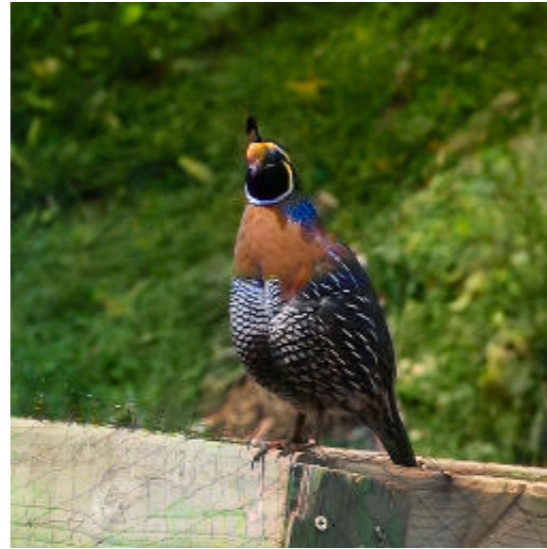
Du point de vue de  $\mathbf{G}$ ,  $\mathbf{D}$  est une fonction de perte.

Au lieu d'être déterminée à la main (L2, etc.), elle est *apprise*.

# Gris vers couleur



# Gris vers couleur

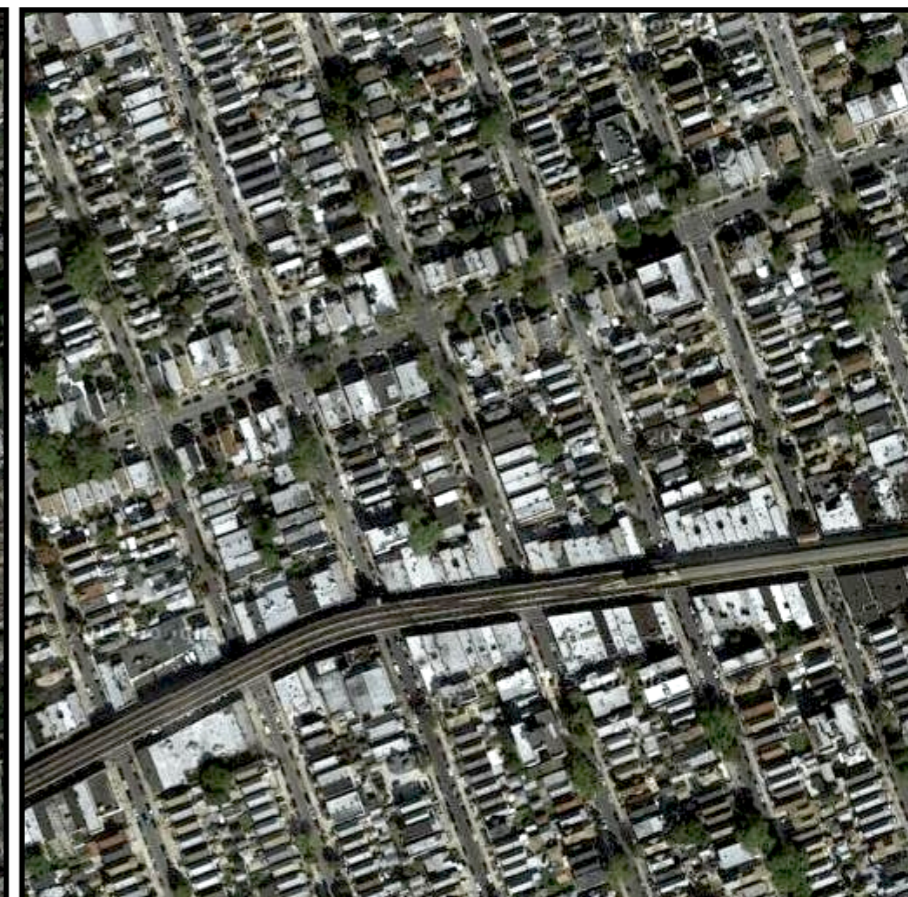
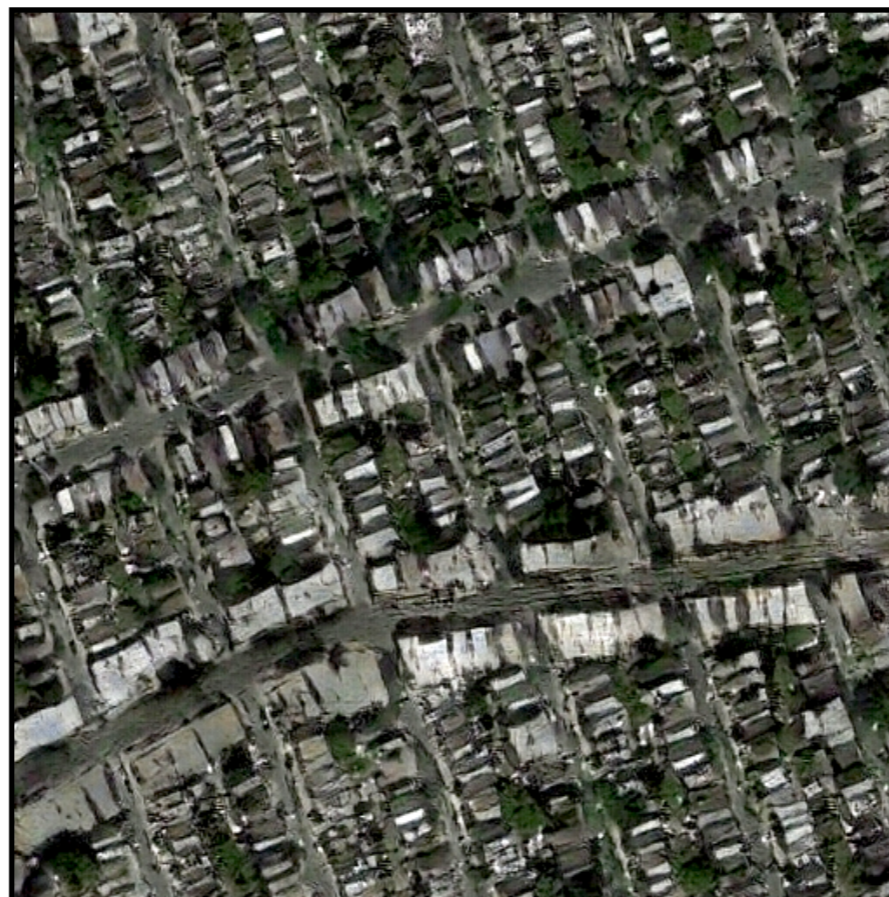


# Carte vers satellite

Input

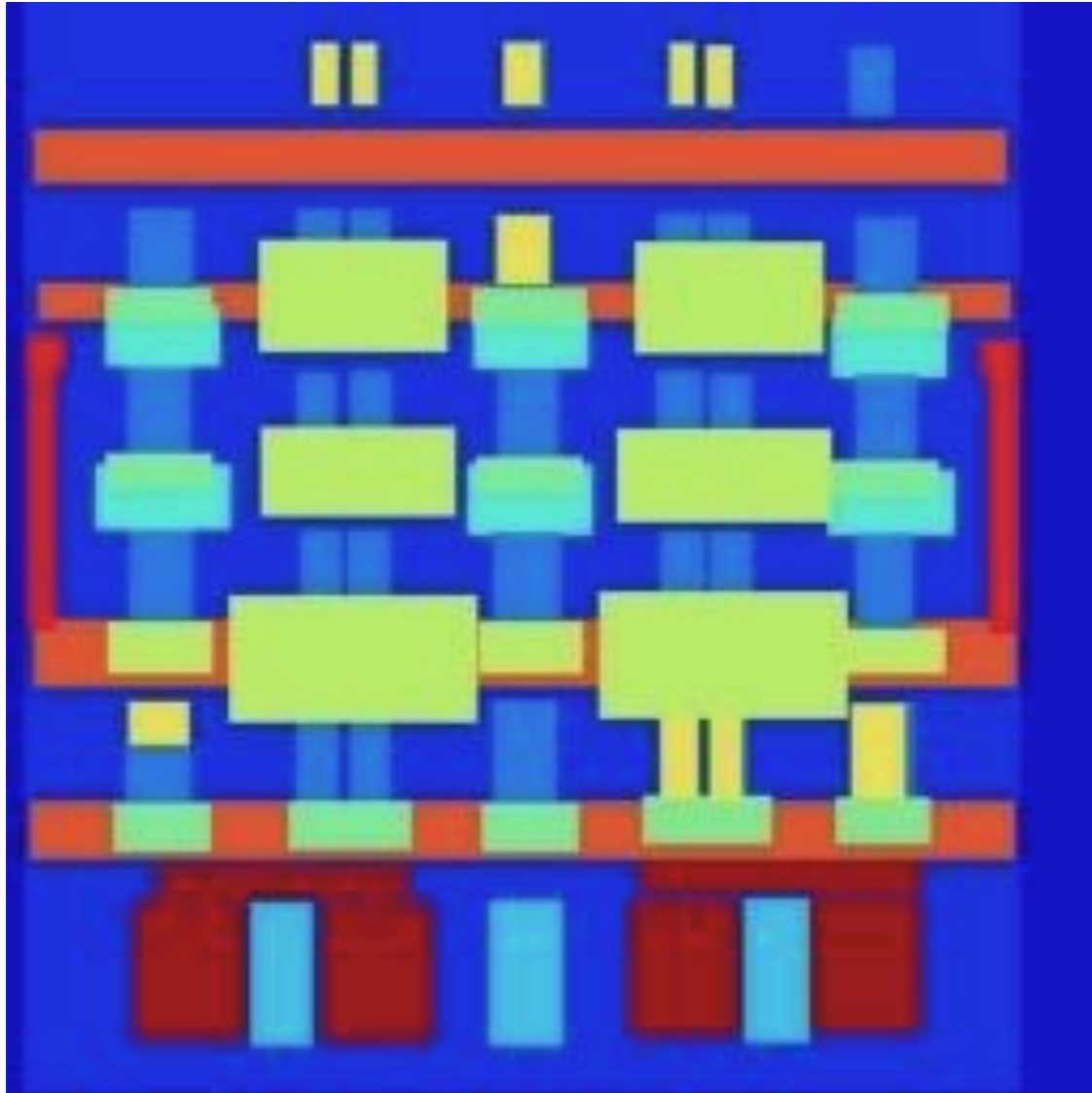
Output

Groundtruth



# Étiquette vers façade

Entrée

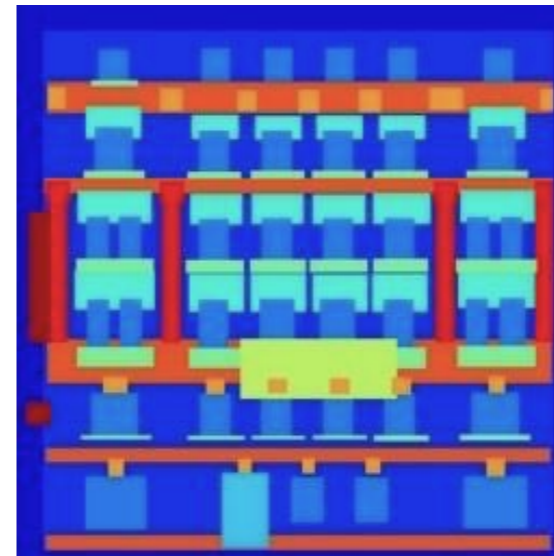
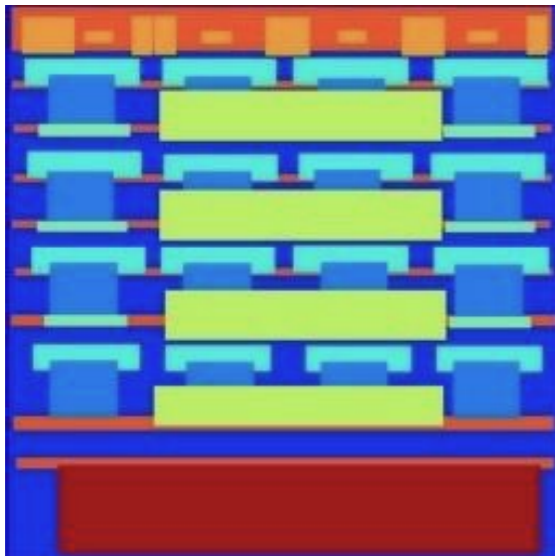
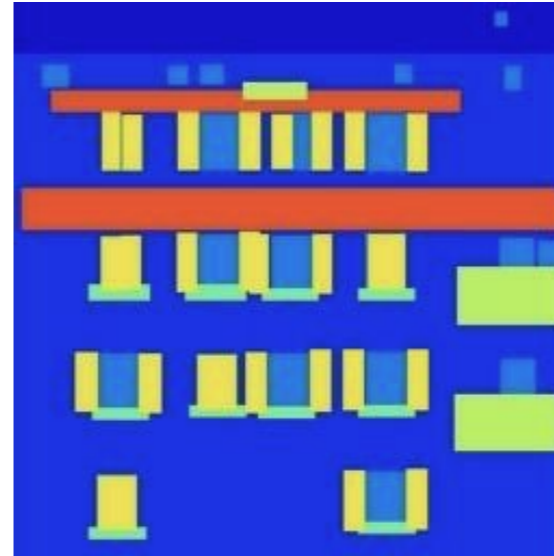
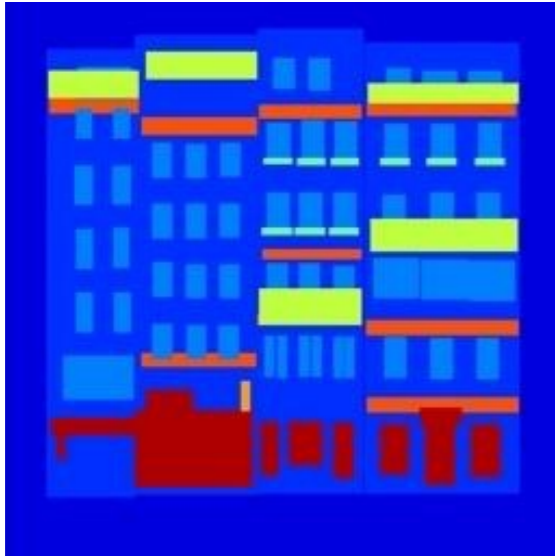


Sortie





# Étiquette vers façade



# C'est le jour et la nuit!

Input

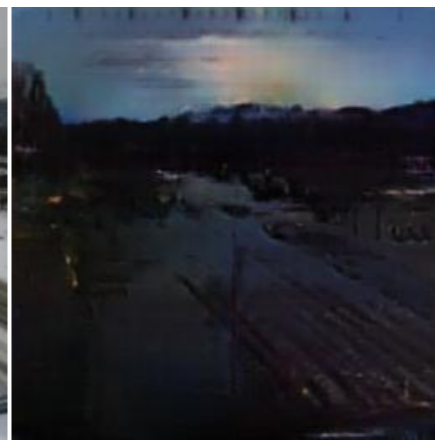
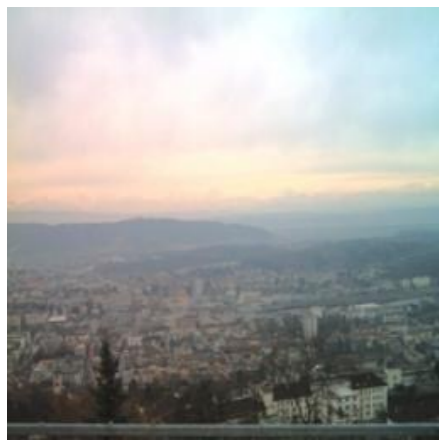
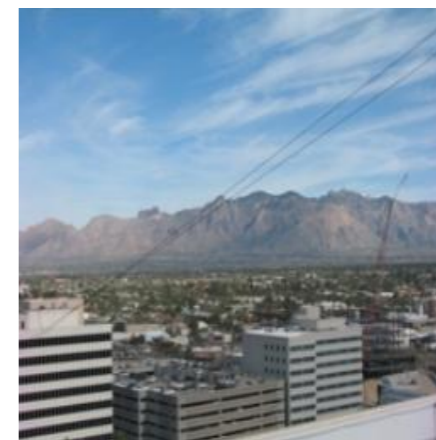
Output

Input

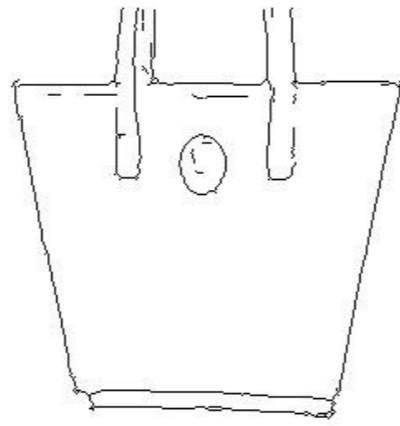
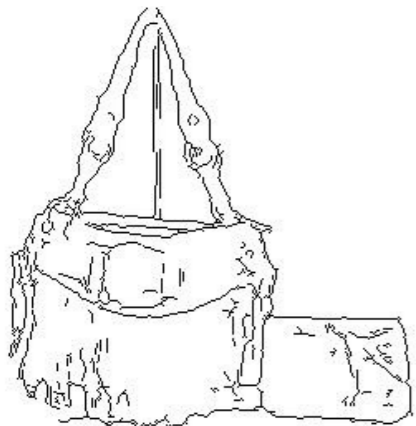
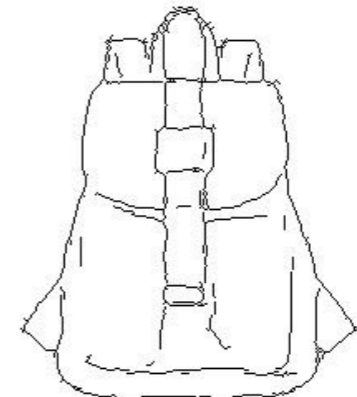
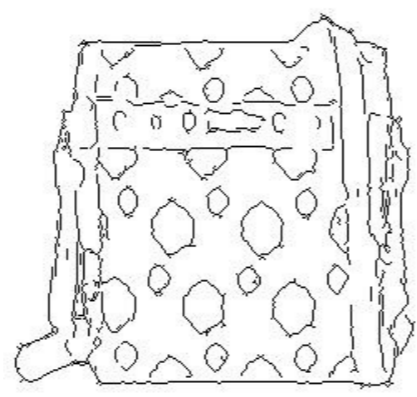
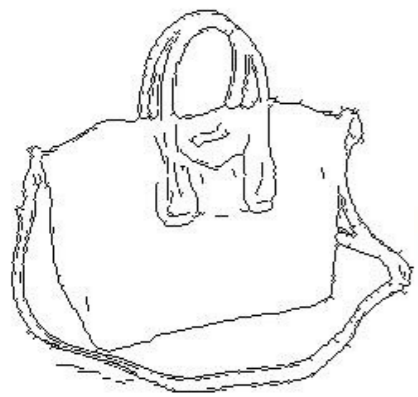
Output

Input

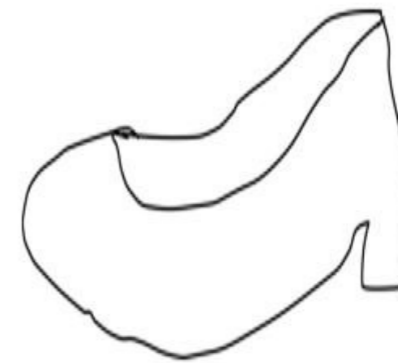
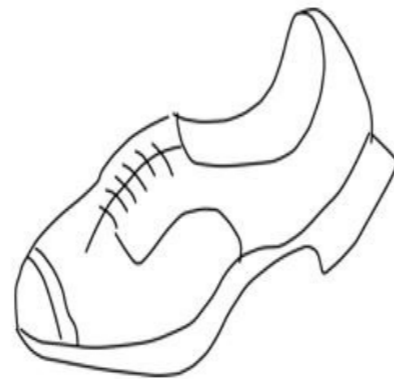
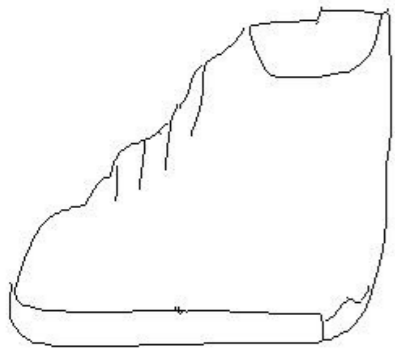
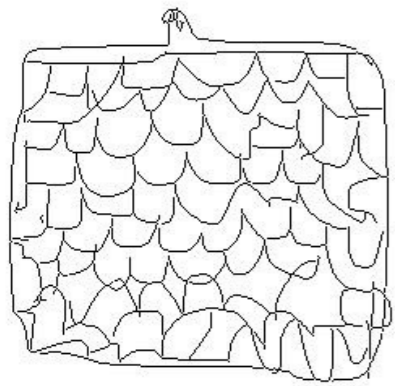
Output



# Arêtes vers images



# Dessins vers images



junyanz / pytorch-CycleGAN-and-pix2pix Watch 176 Star 4,351 Fork 982

Code Issues 28 Pull requests 3 Projects 0 Insights

Image-to-image translation in PyTorch (e.g., horse2zebra, edges2cats, and more)

- pytorch gan cyclegan pix2pix deep-learning computer-vision computer-graphics image-manipulation image-generation
- generative-adversarial-network gans

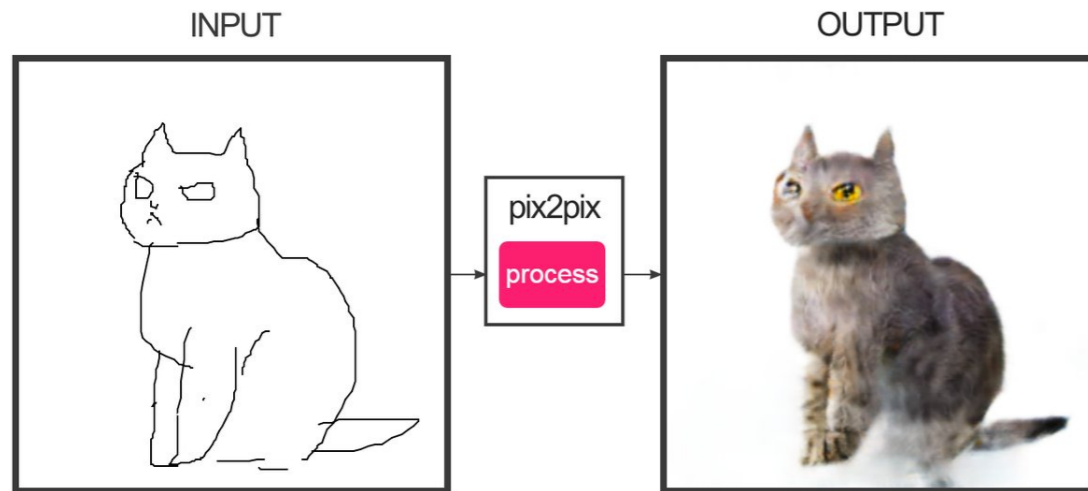
223 commits 3 branches 0 releases 26 contributors

Branch: master New pull request Find file Clone or download

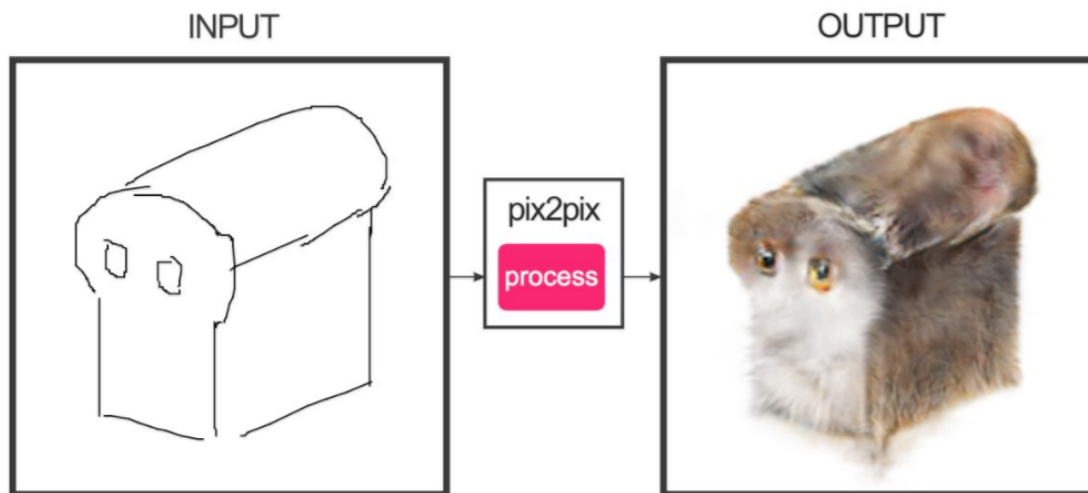
taesung89 Update README.md Latest commit 6d9e173 10 Jun 13, 2018, 12:04 AM PDT

data	1. datasets are now configured automatically based on dataset_mode op...	10 days ago
datasets	Multiple changes regarding option management. See below.	15 days ago
imgs	add edges2cats demo	a year ago
models	TestModel now supports model_suffix option that can change the name o...	10 days ago

# #edges2cats [Christopher Hesse]



@gods\_tail



Ivy Tasi @ivymyt

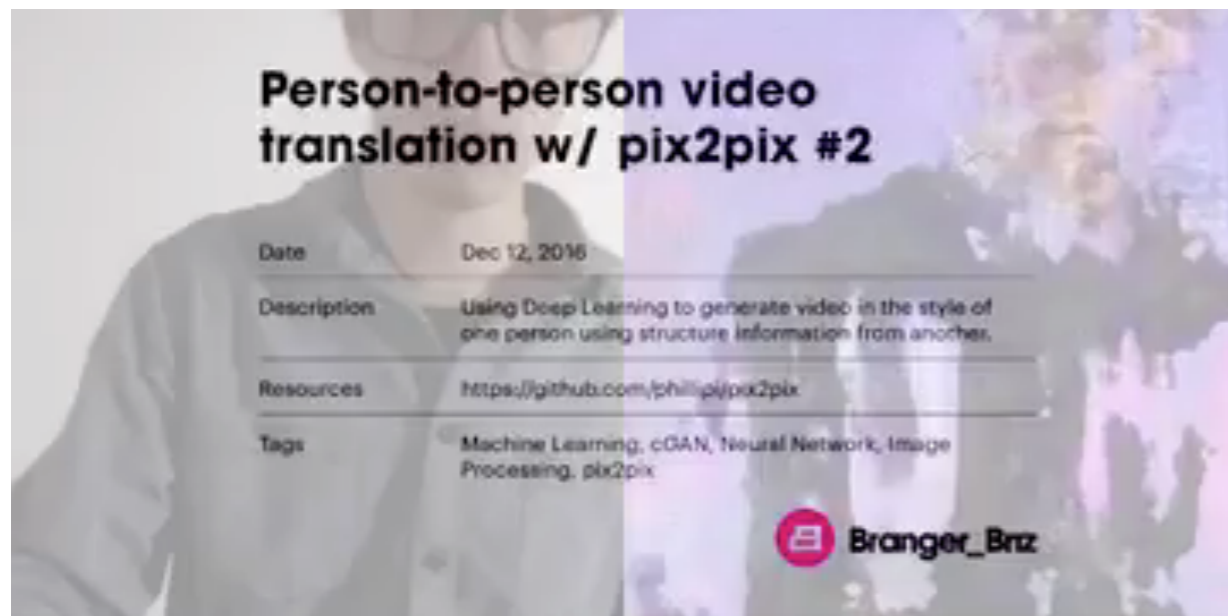


Vitaly Vidmirov  
@vvid

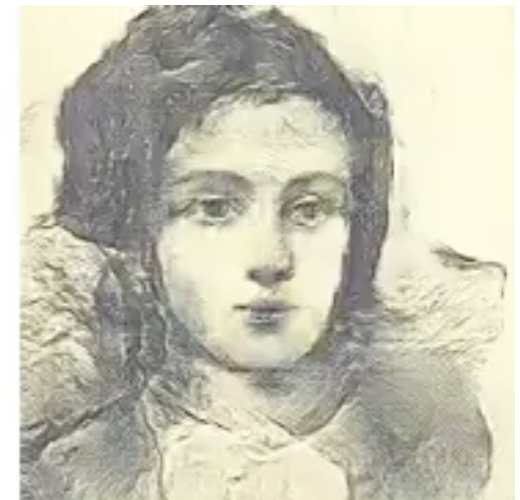


@ka92

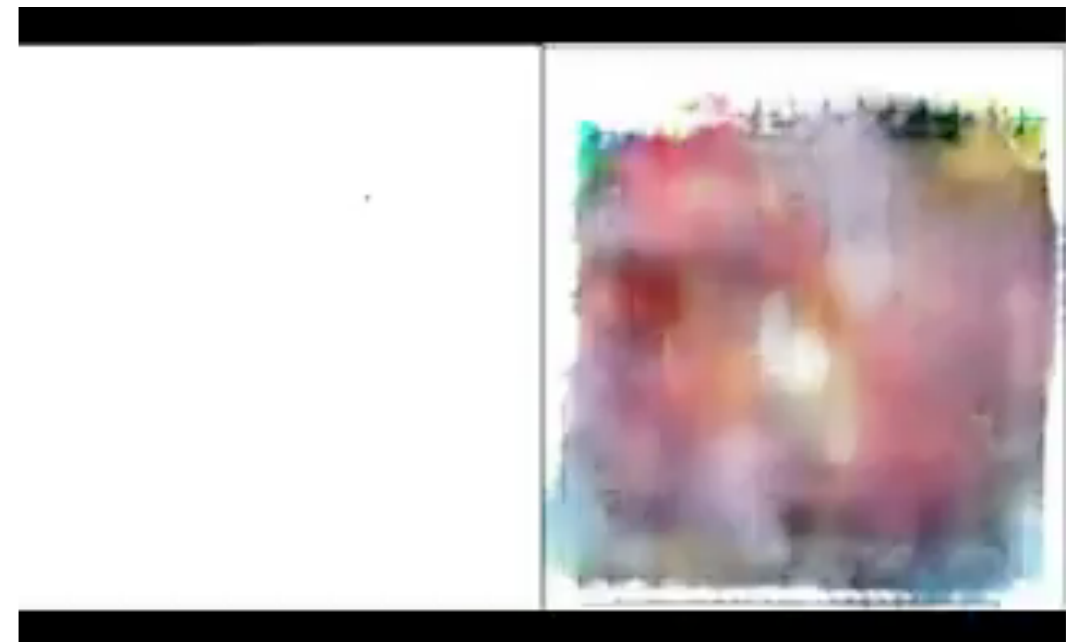
# Twitter-driven research: #pix2pix



Brannon Dorsey @brannondorsey



Mario Klingemann @quasimondo



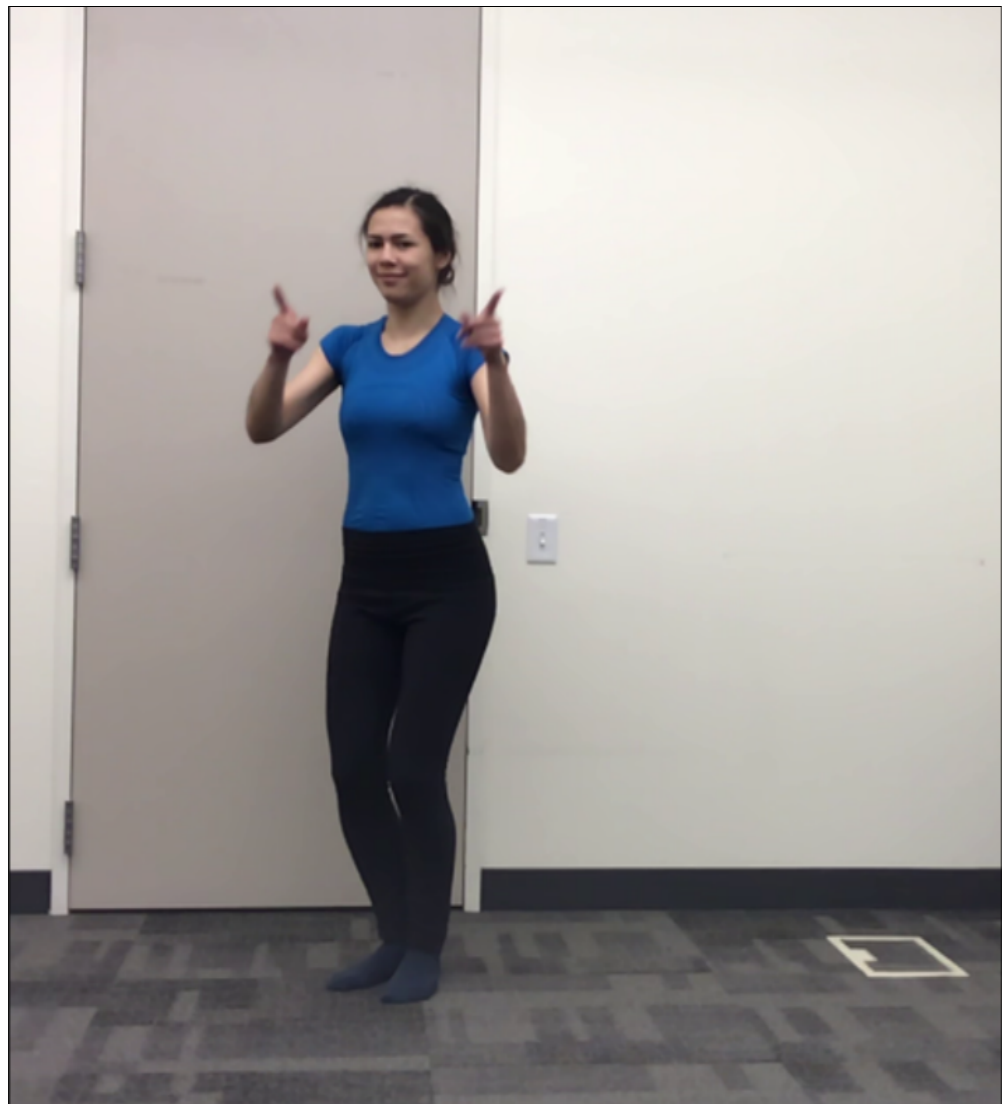
Bertrand Gondouin @bgondouin



© Memo Akten, "Learning to See: Gloomy Sunday"

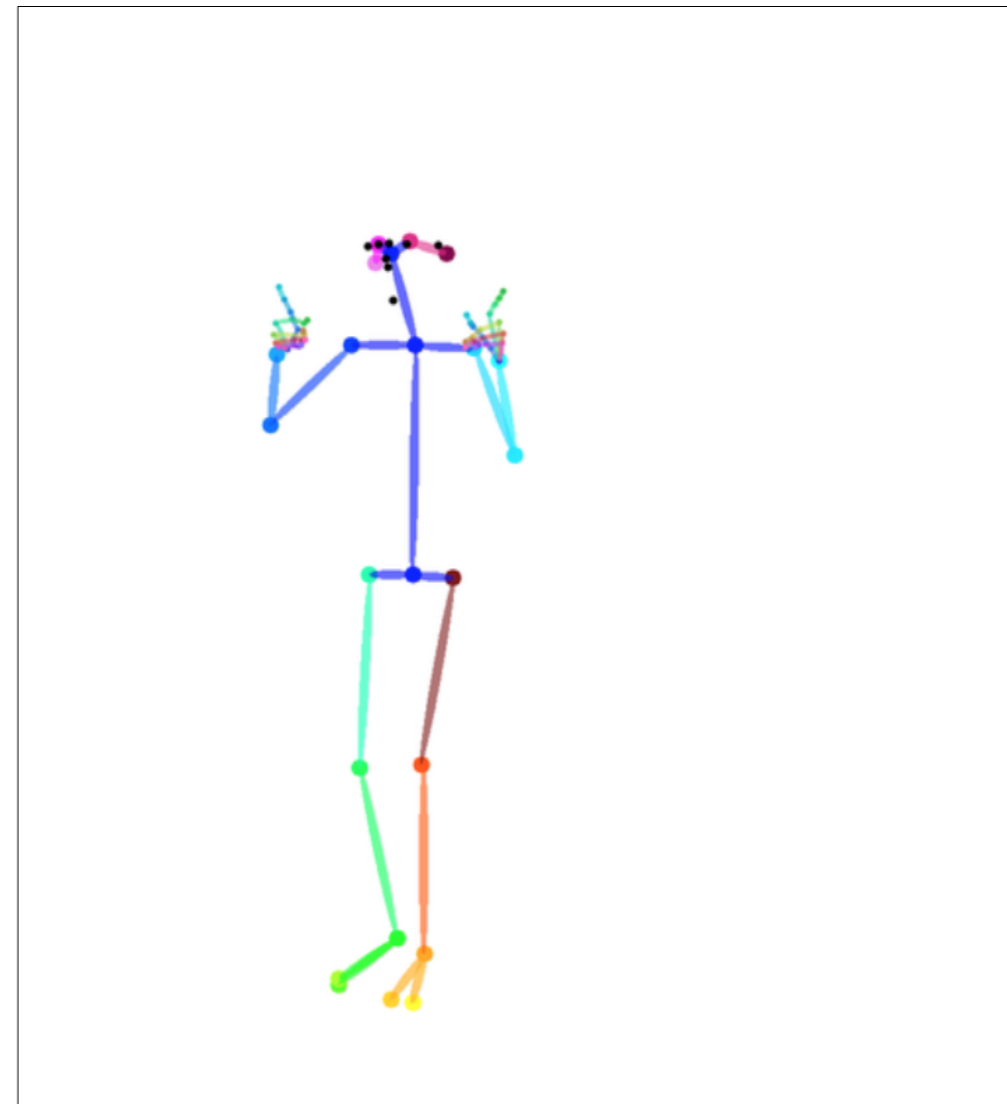
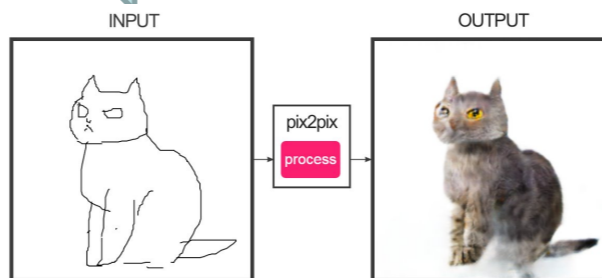


# «Fais comme moi»



OpenPose

pix2pix

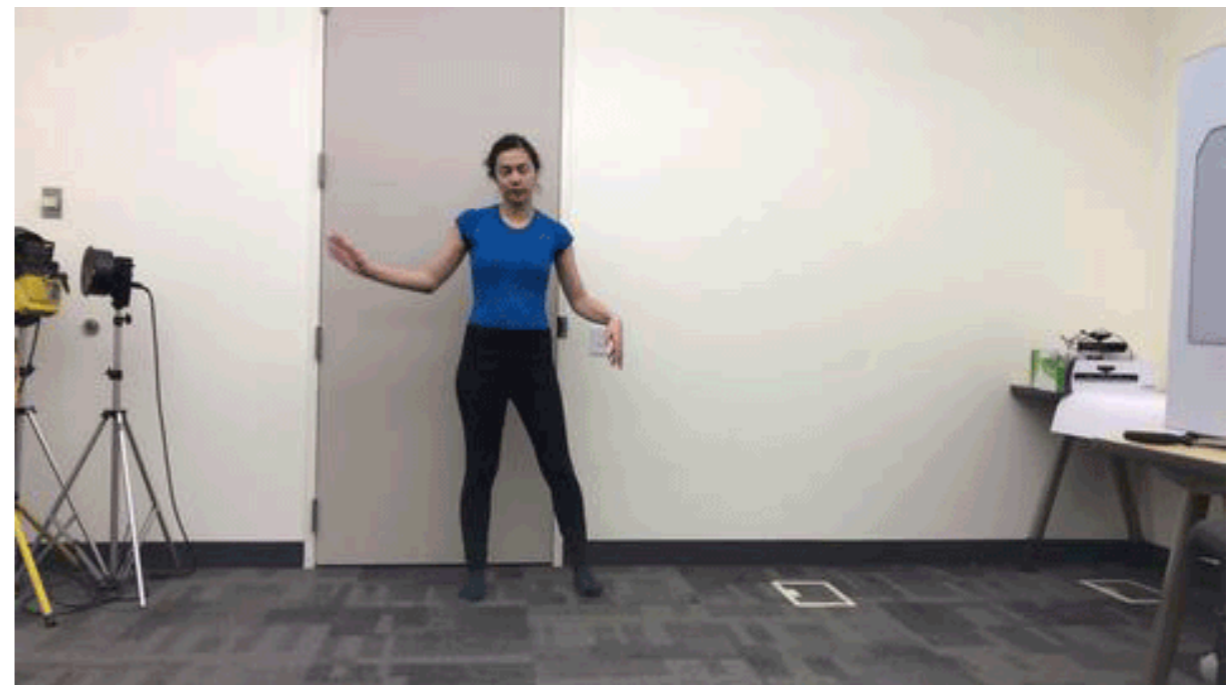


# Everybody Dance Now

Caroline Chan, Shiry Ginosar, Tinghui Zhou, Alexei A. Efros  
UC Berkeley

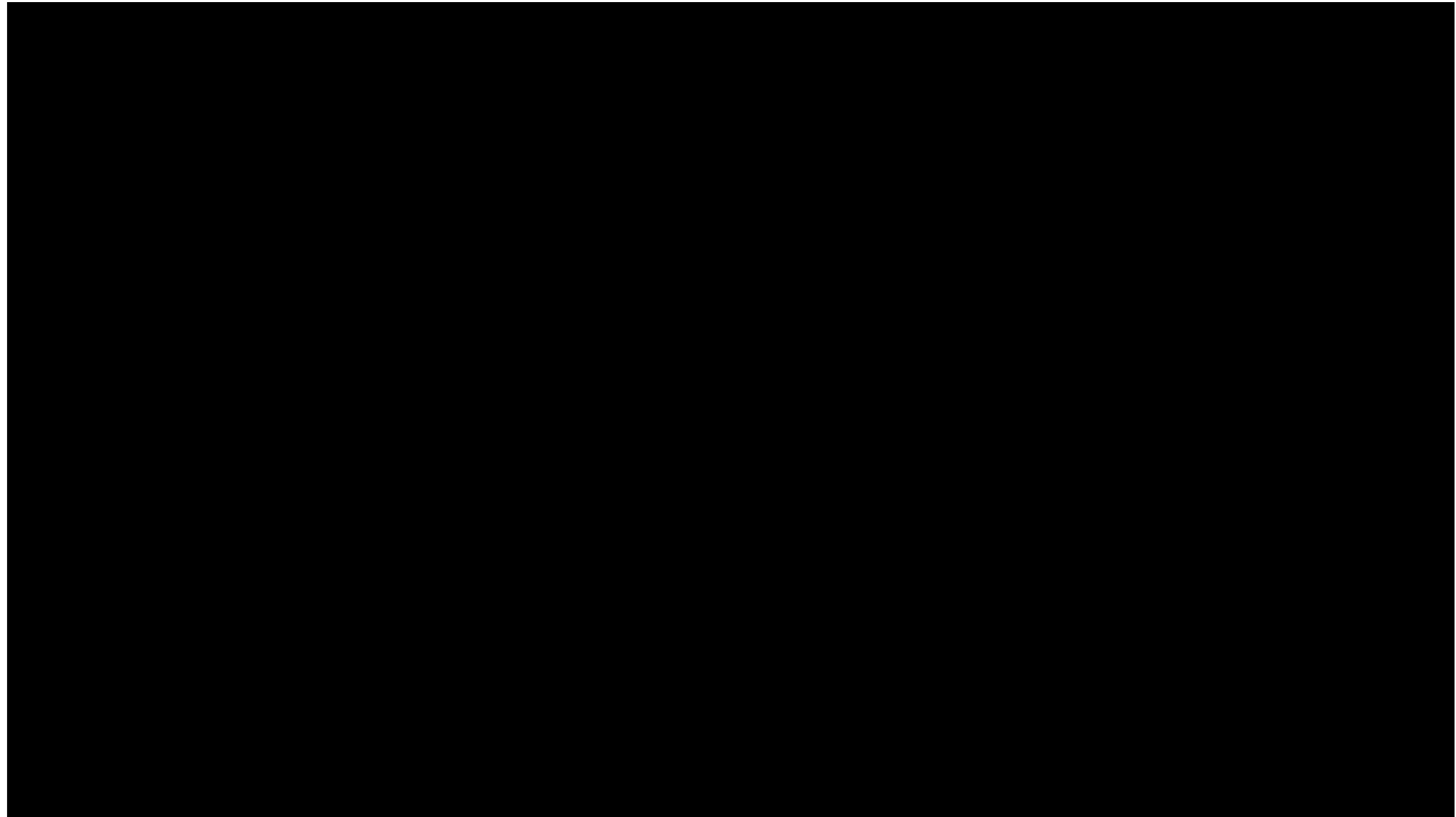


Source



Destination

# Résultats



<https://www.youtube.com/watch?v=PCBTZh41Ris&feature=youtu.be>