

IFT 615 – Intelligence Artificielle

Été 2022

Application – vision artificielle

Professeur: Froduald Kabanza

Assistants: D'Jeff Nkashama & Jean-Charles Verdier

Sujets couverts

- Opérations bas niveau sur les images
 - ◆ détection de contour
 - ◆ calcul de gradients d'image
 - ◆ Corrélation 2D
 - ◆ Convolution 2D
- Aperçu d'un d'un réseau de neurones à convolution

Motivation

- La vue est un sens très utile à la survie d'un organisme
 - ◆ apporte beaucoup d'information sur son environnement (nourriture, prédateur, etc.)
- Presque toutes les créatures intelligentes sont dotées de vision
- Chez l'humain $\approx 25\%$ du cerveau sert à la vision
 - ◆ pour l'ouïe, c'est $\approx 8\%$
 - ◆ pour le touché, c'est $\approx 3\%$
- Ça donne une idée de la complexité de la tâche à résoudre...

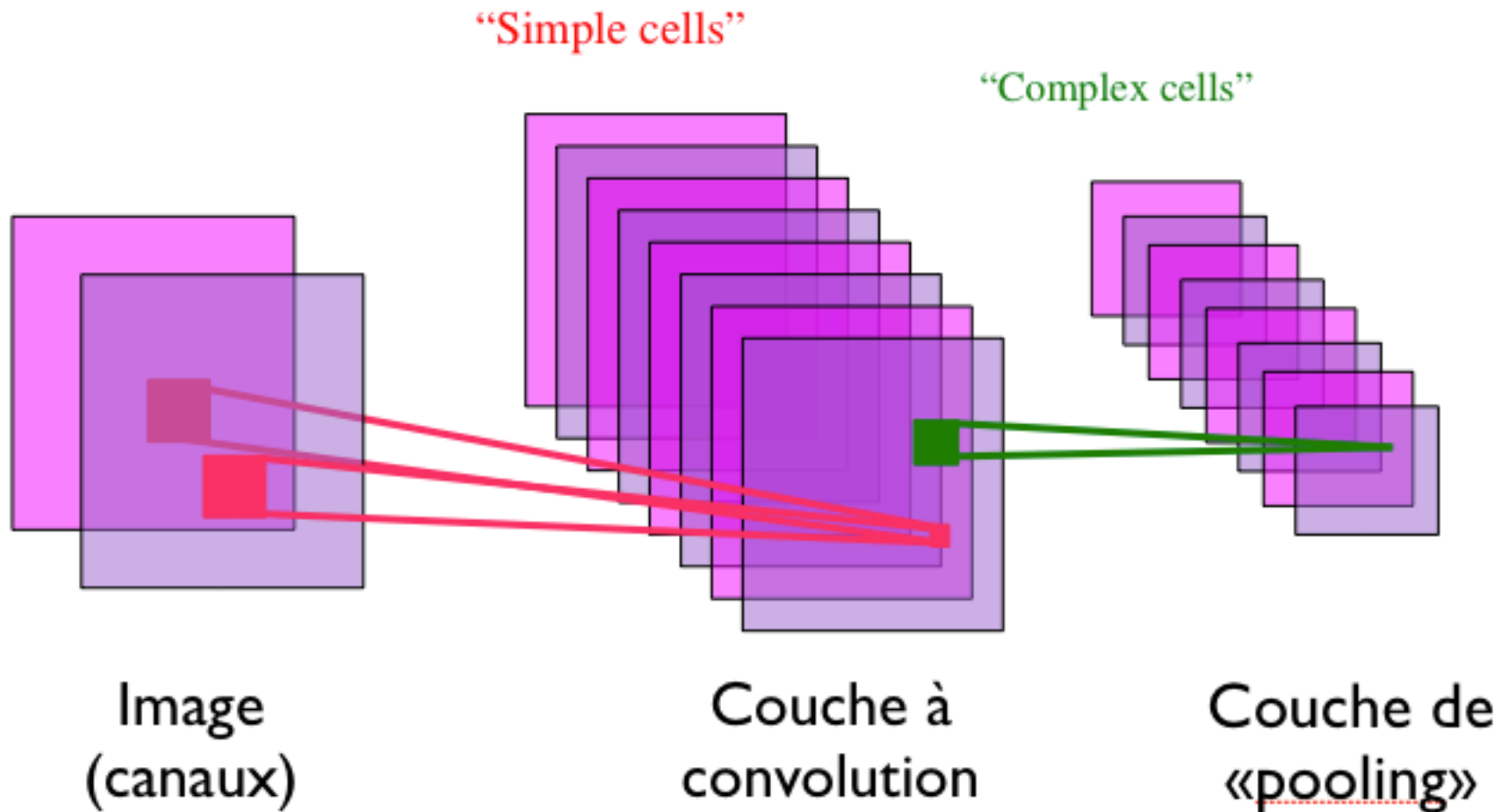
Dans ce cours...

- On va voir comment on manipule des images
 - ◆ quelle représentation de base utiliser
 - ◆ quel genre de prétraitements sont utiles
- L'objectif est d'avoir une vue d'ensemble des approches suivies en vision par ordinateur
- On va discuter des concepts fréquemment utilisés en vision
 - ◆ gradients d'image
 - ◆ convolution
 - ◆ « *pooling* »
 - ◆ *histogramme de gradients*

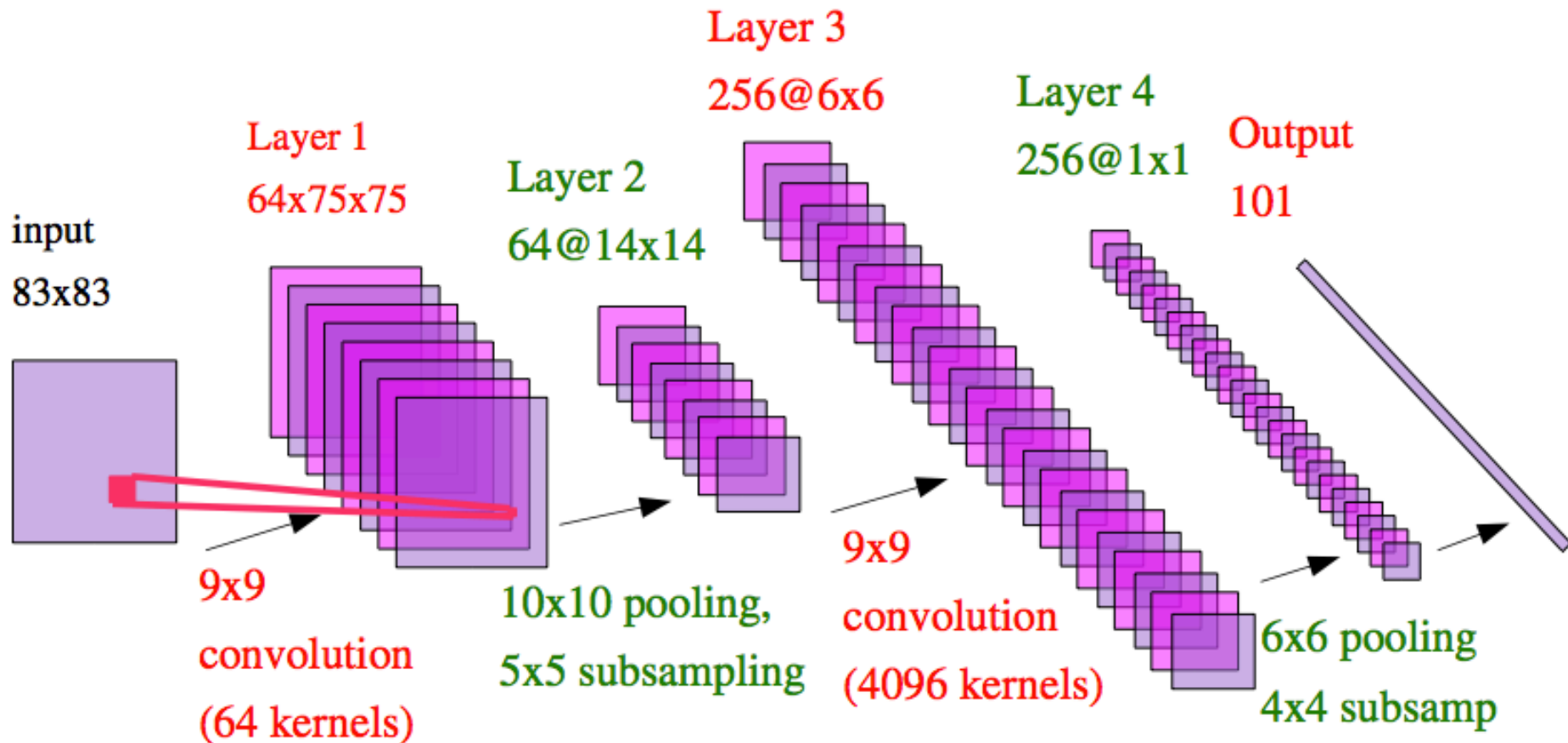
Histoire du CNN

- **1959 & 1962:** [David Hubel & Torsten Wiesel](#) : Fonctionnement de la vision chez les animaux https://www.youtube.com/watch?v=y_l4kQ5wjiw
- **1980s** : Kunihiro Fukushima – Architecture Neocognitron inspire des travaux de Hubel et Wiesel
 - ◆ [Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position](#)
- **1990s** : LeCun – CNN, inspiré par les travaux de Fukushima
 - ◆ [Gradient-Based Learning Applied to Document recognition](#)
- **2012** : Hinton et al. – AlexNet performe mieux sur ImageNet mieux que les approches traditionnelles
 - ◆ [ImageNet Classification with Deep Convolutional Neural Networks](#)

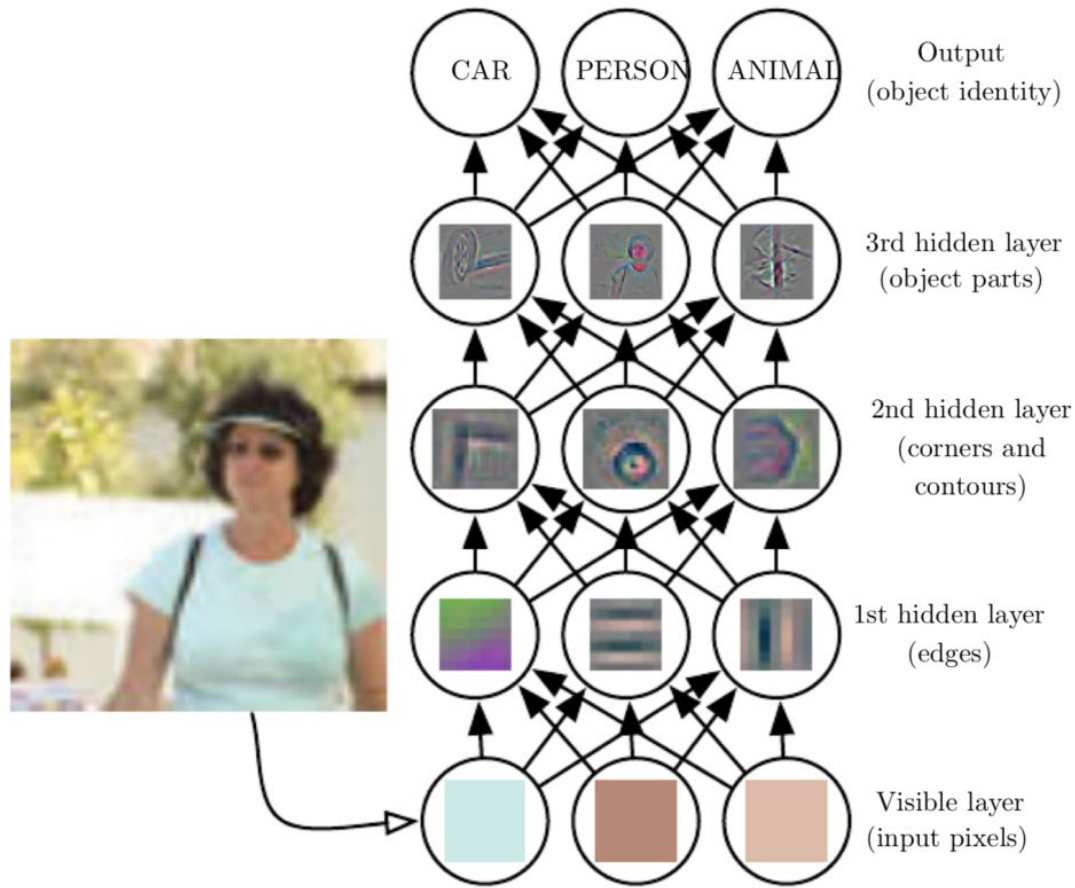
Réseau de neurones à convolution: structure des couches cachées



Réseau de neurones à convolution: réseau complet



Chaque couche apprend une abstraction



<https://bdtechtalks.com/2020/01/06/convolutional-neural-networks-cnn-convnets/>

Convolution

- Calcul d'une couche « simple cell »
 - ◆ première étape : calcul de la convolution

0	0.5
0.5	0

W

connexions
vers les neurones
cachés

0	0	0.5	255	0	0
0	0.5	0	255	0	0
0	0	255	0	0	0
0	255	0	0	0	0
255	0	0	0	0	0

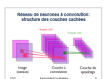
X

couche d'entrée

0	128	128	0
0	128	128	0
0	255	0	0
255	0	0	0

$X * W$

couche « simple cell »



Convolution

- Calcul d'une couche « simple cell »
 - ◆ première étape : calcul de la convolution
 - ◆ deuxième étape : calcul de la non-linéarité (ex.: $Logistic((x-200)/50)$)

0	0	255	0	0
0	0	255	0	0
0	0	255	0	0
0	255	0	0	0
255	0	0	0	0

X

couche d'entrée

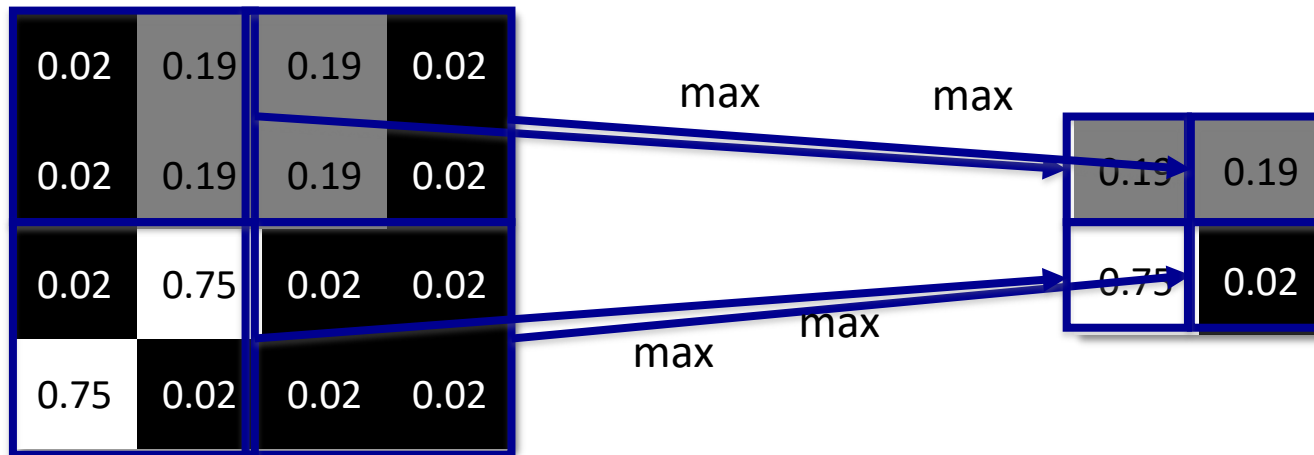
0.02	0.19	0.19	0.02
0.02	0.19	0.19	0.02
0.02	0.75	0.02	0.02
0.75	0.02	0.02	0.02

$Logistic((X * W - 200) / 50)$

couche « simple cell »

Max Pooling

- Calcul d'une couche « complex cell »
 - ◆ maximum dans plusieurs segments



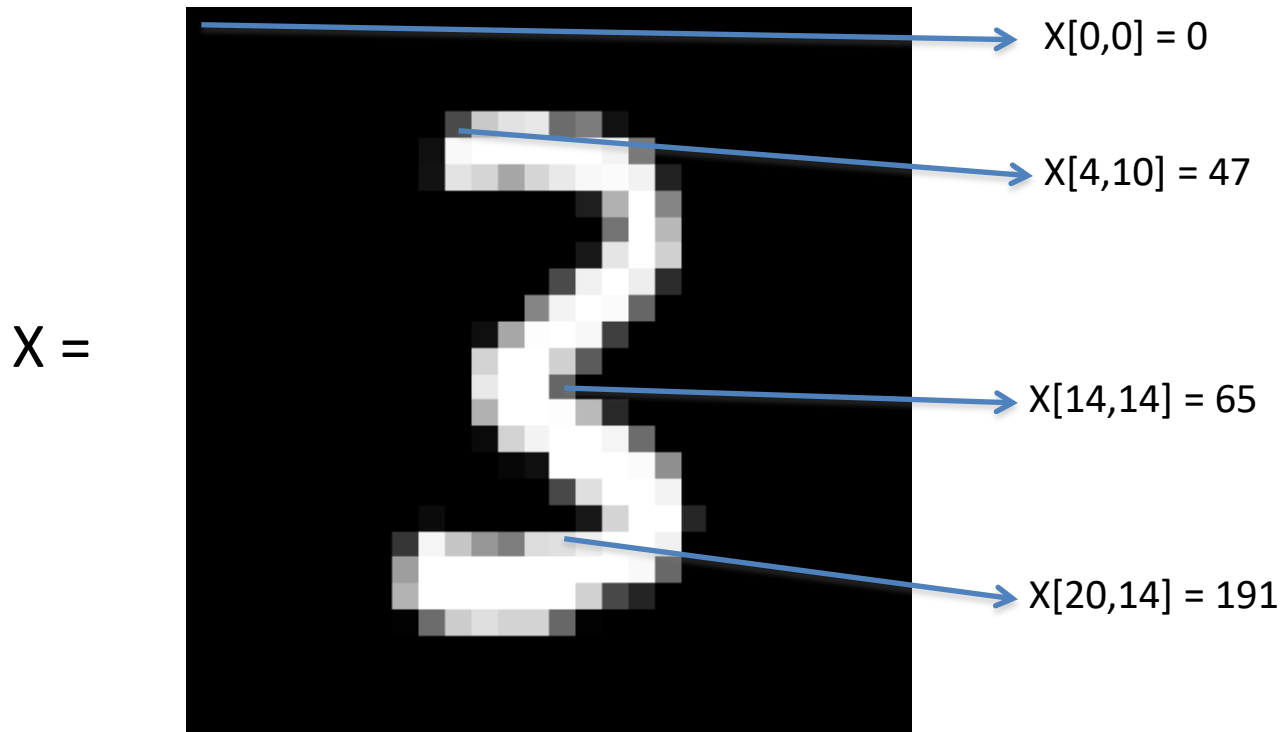
$g((X * W - 200) / 50)$ g : fonction d'activation

couche « simple cell »

couche « complex cell »

Représentation brute d'une image

- Image en niveau de gris: tableau 2D de pixels, entiers positifs de 8 bits

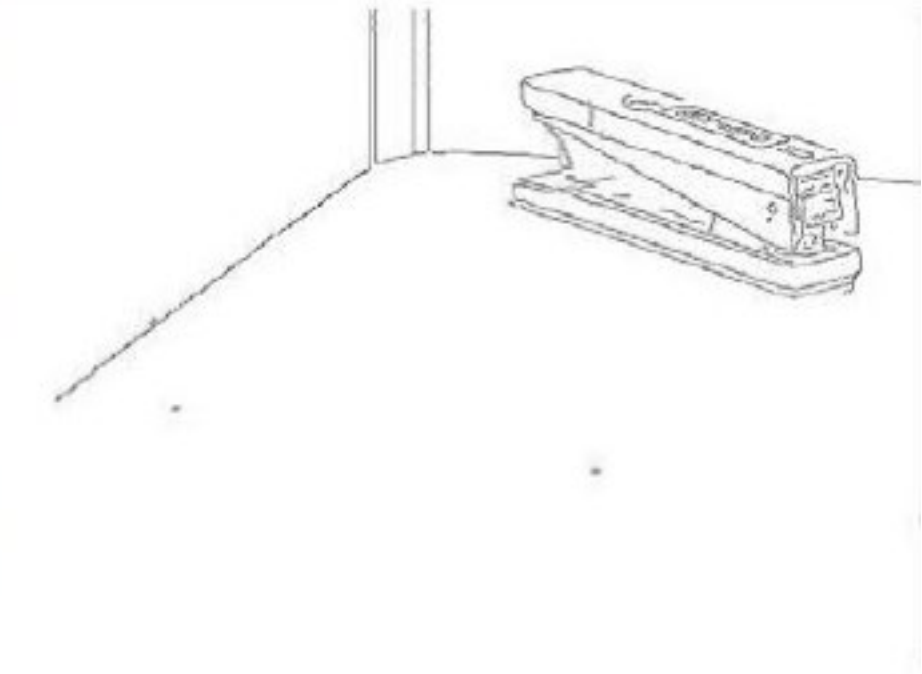


Contour

- Un contour est une changement soudain dans l'intensité/couleur de pixels adjacents



image originale

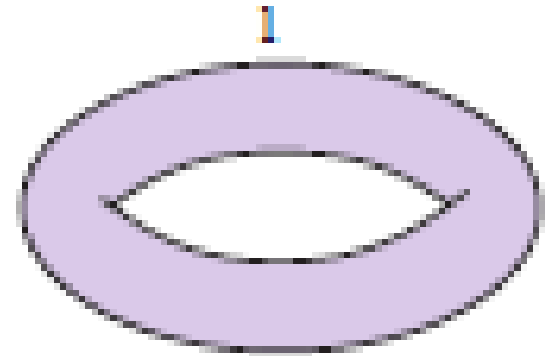
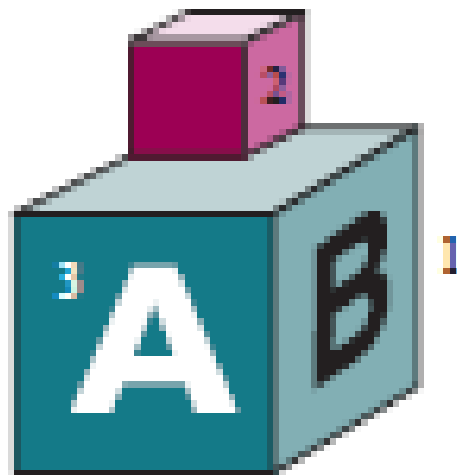
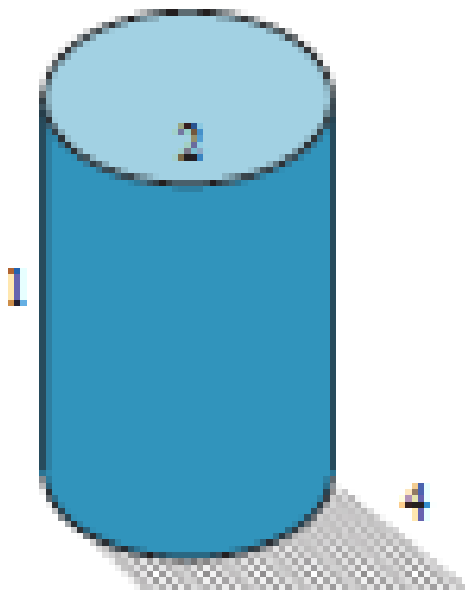


extraction des contours

Vision animale

- **1959 & 1962:** [David Hubel & Torsten Wiesel](#) : Fonctionnement de la vision chez les animaux https://www.youtube.com/watch?v=y_l4kQ5wjiw

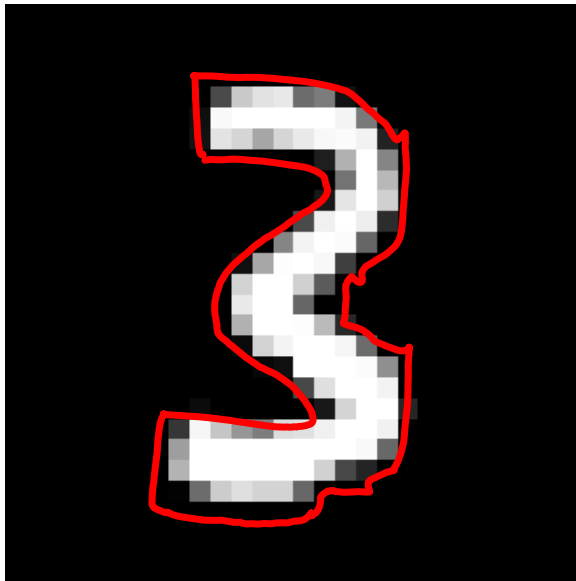
Types de contours d'images



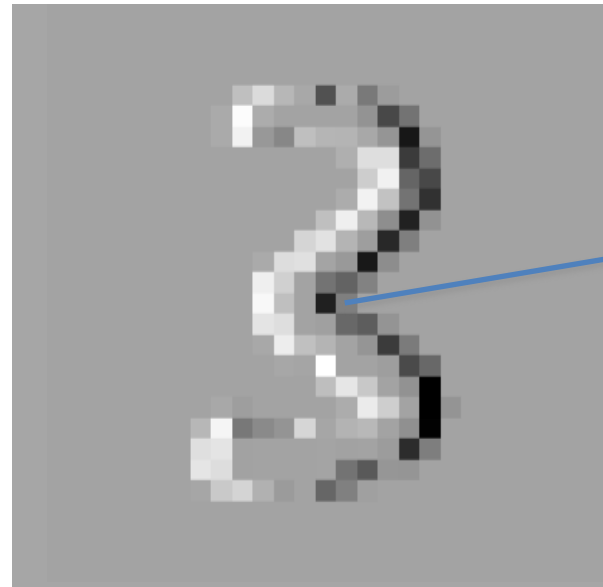
- 1- changement de profondeur
- 2- changement d'orientation de surface
- 3- changement de couleur (réflexion)
- 4- changement d'illumination

Gradient d'image

- Pour détecter si un pixel est sur la frontière d'un contour, on peut regarder la valeur relative des pixels autour de ce pixel
- Exemple: variation **horizontale** $H[i,j] = X[i,j+1] - X[i,j]$



X

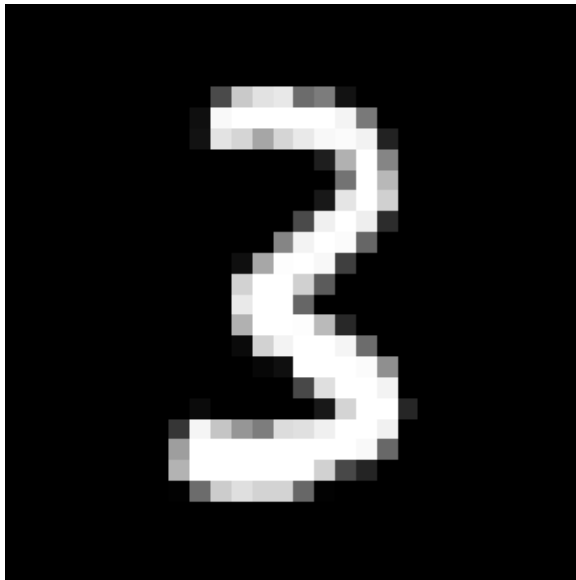


H

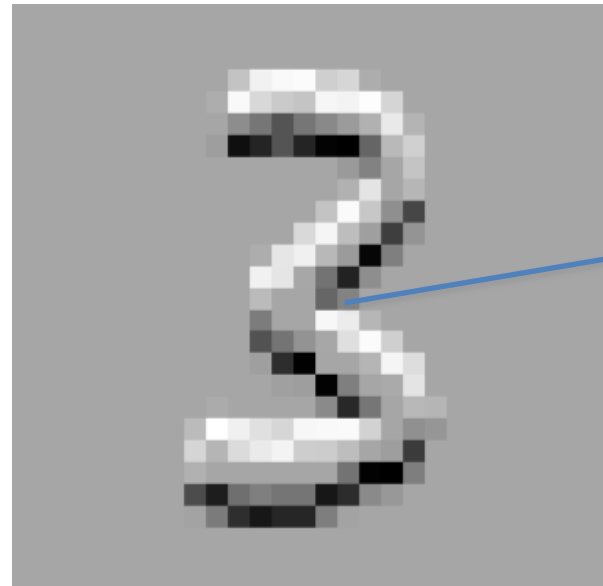
$H[14,14] =$
 $X[14,15] -$
 $X[14,14]$

Gradient d'image

- Pour détecter si un pixel est sur la frontière d'un contour, on peut regarder la valeur relative des pixels autour de ce pixel
- Exemple: variation **verticale** $V[i,j] = X[i+1,j] - X[i,j]$



X



V

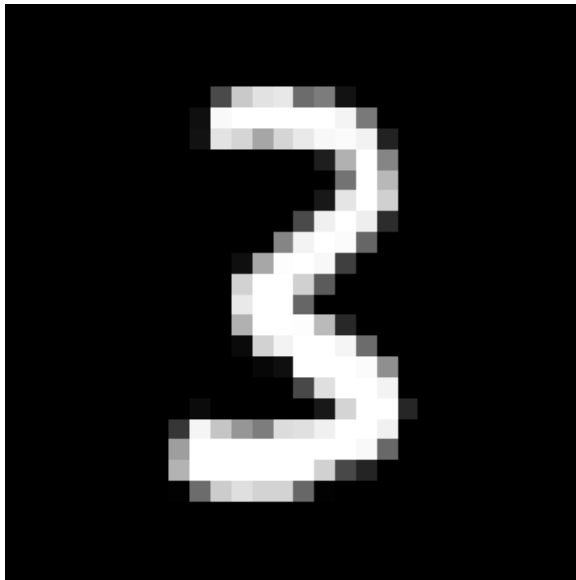
$$\begin{aligned} V[14,14] &= \\ X[15,14] &- \\ X[14,14] \end{aligned}$$

Détecter un contours à partir des gradients d'image

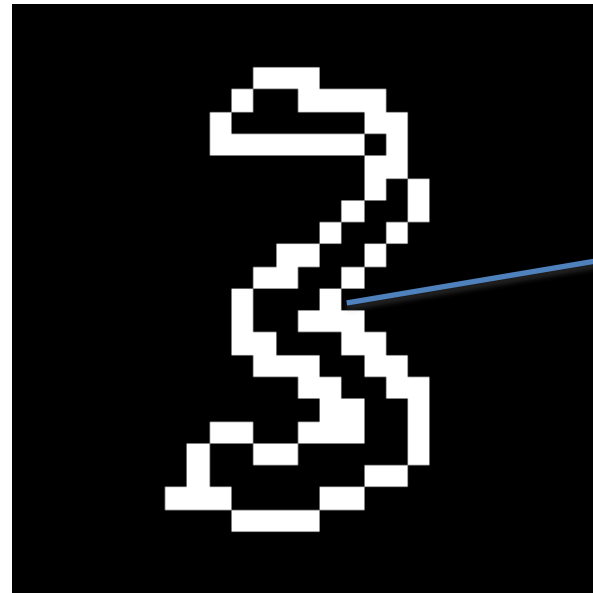
- Un pixel ferait partie d'un contour si la somme des variations (positive ou négative) horizontale et verticale est élevée

$$E[i,j] = \text{sqrt}(V[i,j]**2 + H[i,j]**2)$$

- On applique un seuil pour déterminer si contour ou pas



X



$E > 128$

$E[14,14] > 128$

Gradient d'image

- On peut voir le calcul des variations comme des dérivées partielles
- La « fonction » $f(a, b)$ serait la valeur de l'image à la position (a, b)

$$\frac{\partial f(a, b)}{\partial b} = \lim_{\Delta \rightarrow 0} \frac{f(a, b + \Delta) - f(a, b)}{\Delta} \approx \underbrace{X[i, j+1] - X[i, j] = H[i, j]}_{\Delta = 1}$$

$$\frac{\partial f(a, b)}{\partial a} = \lim_{\Delta \rightarrow 0} \frac{f(a + \Delta, b) - f(a, b)}{\Delta} \approx \underbrace{X[i+1, j] - X[i, j] = V[i, j]}$$

Gradient d'image

- Si $H[i,j]$ et $V[i,j]$ sont les dérivées partielles de l'image, alors

$$G[i,j,:] = [H[i,j], V[i,j]]$$

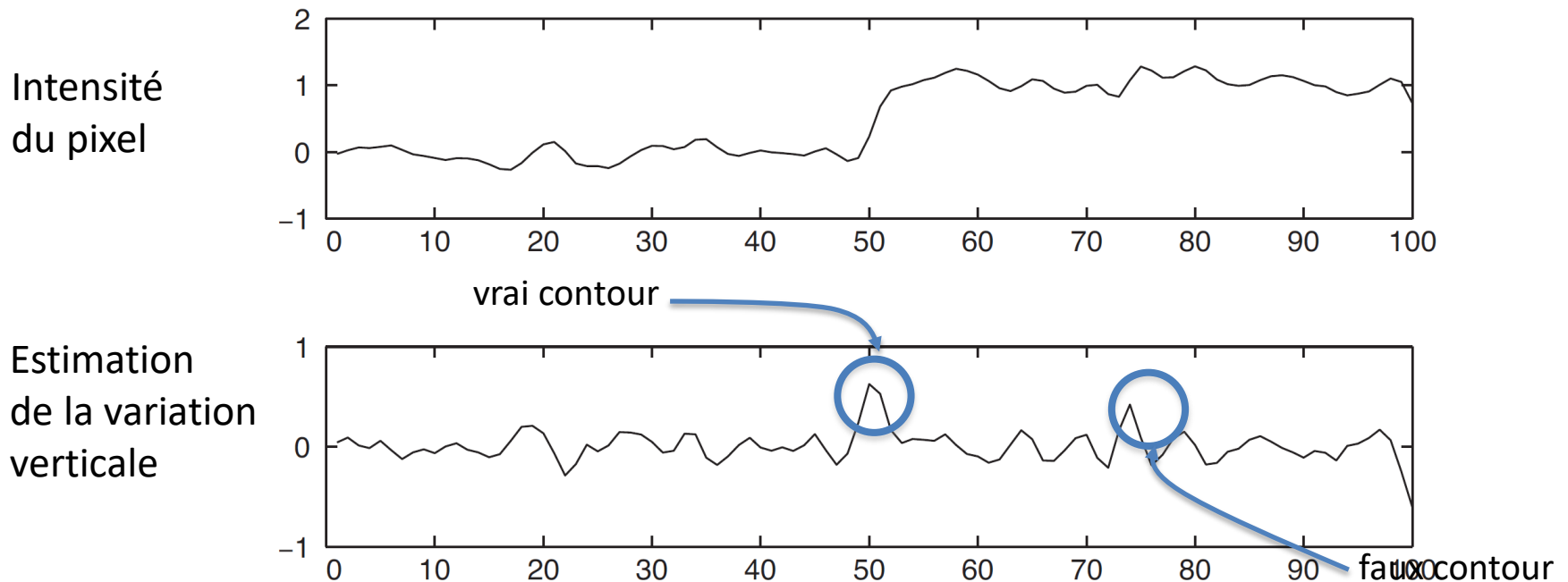
est le **gradient de l'image**, à la position (i,j)

- Pour détecter contours, l'idée serait de calculer donc la norme euclidienne de ces gradients et voir où les gradients changent significativement

$$E[i,j] = \sqrt{V[i,j]^2 + H[i,j]^2} = \underbrace{\sqrt{\text{sum}(G[i, j, :]^2)}}_{\text{norme du vecteur } G[i,j,:]}$$

Exemple de calcul du gradient bruité

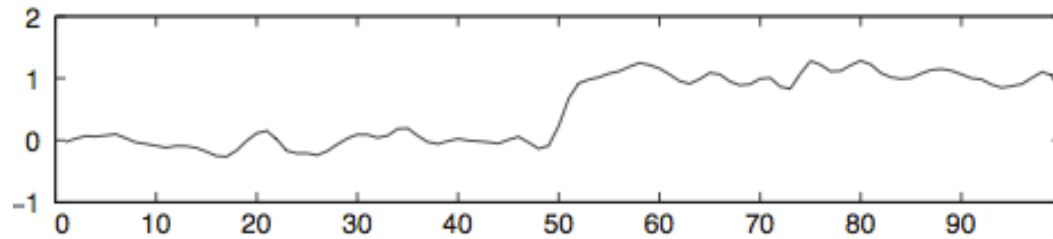
- Des vraies images sont bruitées et donc variations des gradients vont l'être aussi



- Pour éliminer la détection de faux contours, on peut lisser l'image

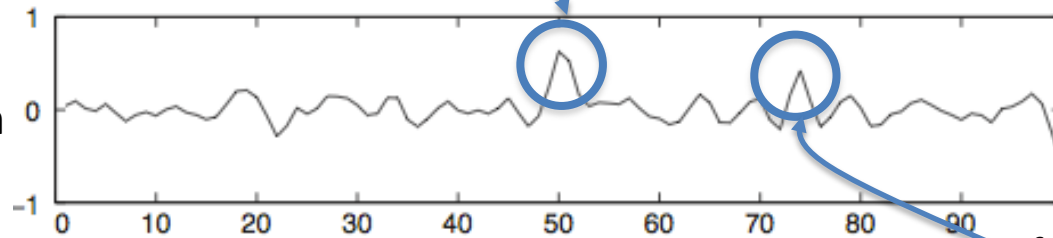
Calcul gradient d'image lissé

Intensité
du pixel



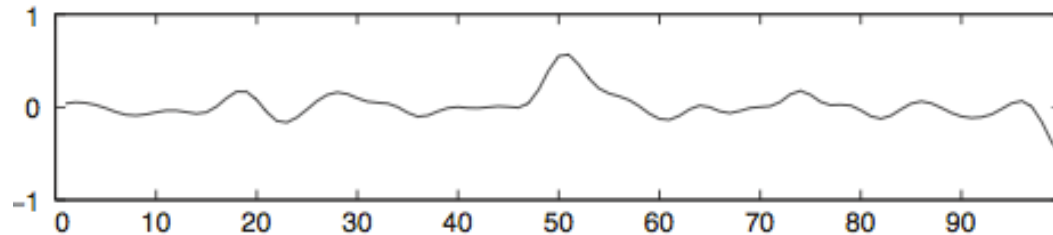
vrai contour

Estimation
de la variation
verticale



faux contour

Estimation
de la variation
Verticale d'une
version lissée de
l'intensité



Lissage Gaussien d'une image

0	0	255	0	0
0	0	255	0	0
0	0	255	0	0
0	255	0	0	0
255	0	0	0	0

X

U(0,0)	U(0,1)	U(0,2)	U(0,3)	U(0,4)
U(1,0)	U(1,1)	U(1,2)	U(1,3)	U(1,4)
U(2,0)	U(2,1)	U(2,2)	U(2,3)	U(2,4)
U(3,0)	U(3,1)	U(3,2)	U(3,3)	U(3,4)
U(4,0)	U(4,1)	U(2,2)	U(4,3)	U(4,4)

U

$$U(i_0, j_0) = \sum_i \sum_j X[i, j] G_\sigma(d),$$

d étant la distance de (i_0, j_0) à (i, j) .

$$G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}$$

Lissage Gaussien d'une image

0	0	255	0	0
0	0	255	0	0
0	0	255	0	0
0	255	0	0	0
255	0	0	0	0

X

U(0,0)	U(0,1)	U(0,2)	U(0,3)	U(0,4)
U(1,0)	U(1,1)	U(1,2)	U(1,3)	U(1,4)
U(2,0)	U(2,1)	U(2,2)	U(2,3)	U(2,4)
U(3,0)	U(3,1)	U(3,2)	U(3,3)	U(3,4)
U(4,0)	U(4,1)	U(2,2)	U(4,3)	U(4,4)

C

On somme seulement sur (i,j)
dans un voisinage inférieur à 3σ

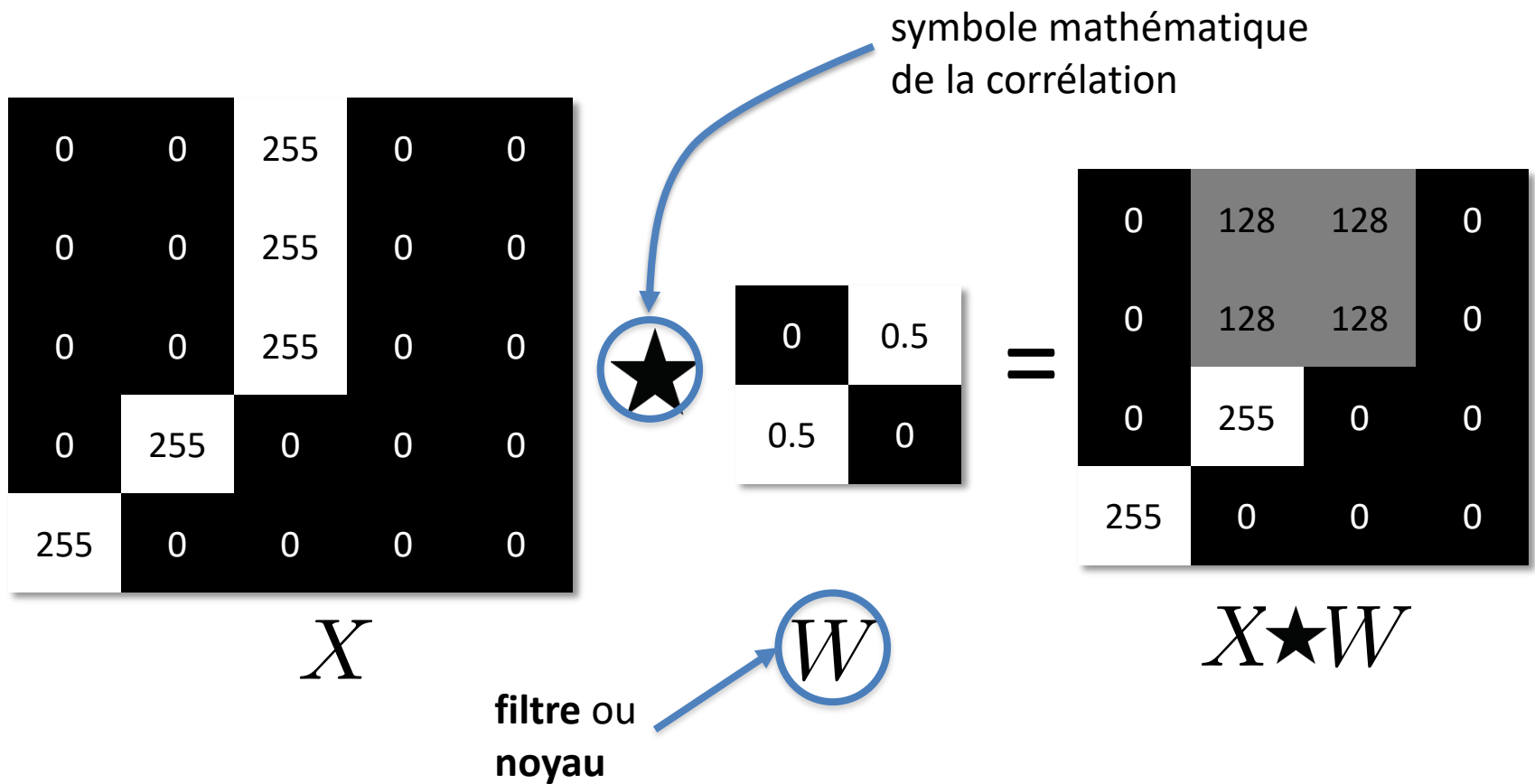
$$C(i_0, j_0) = \sum_i \sum_j X[i,j] G_\sigma(d),$$

d étant la distance de (i_0, j_0) à (i, j) .

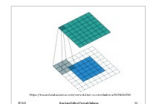
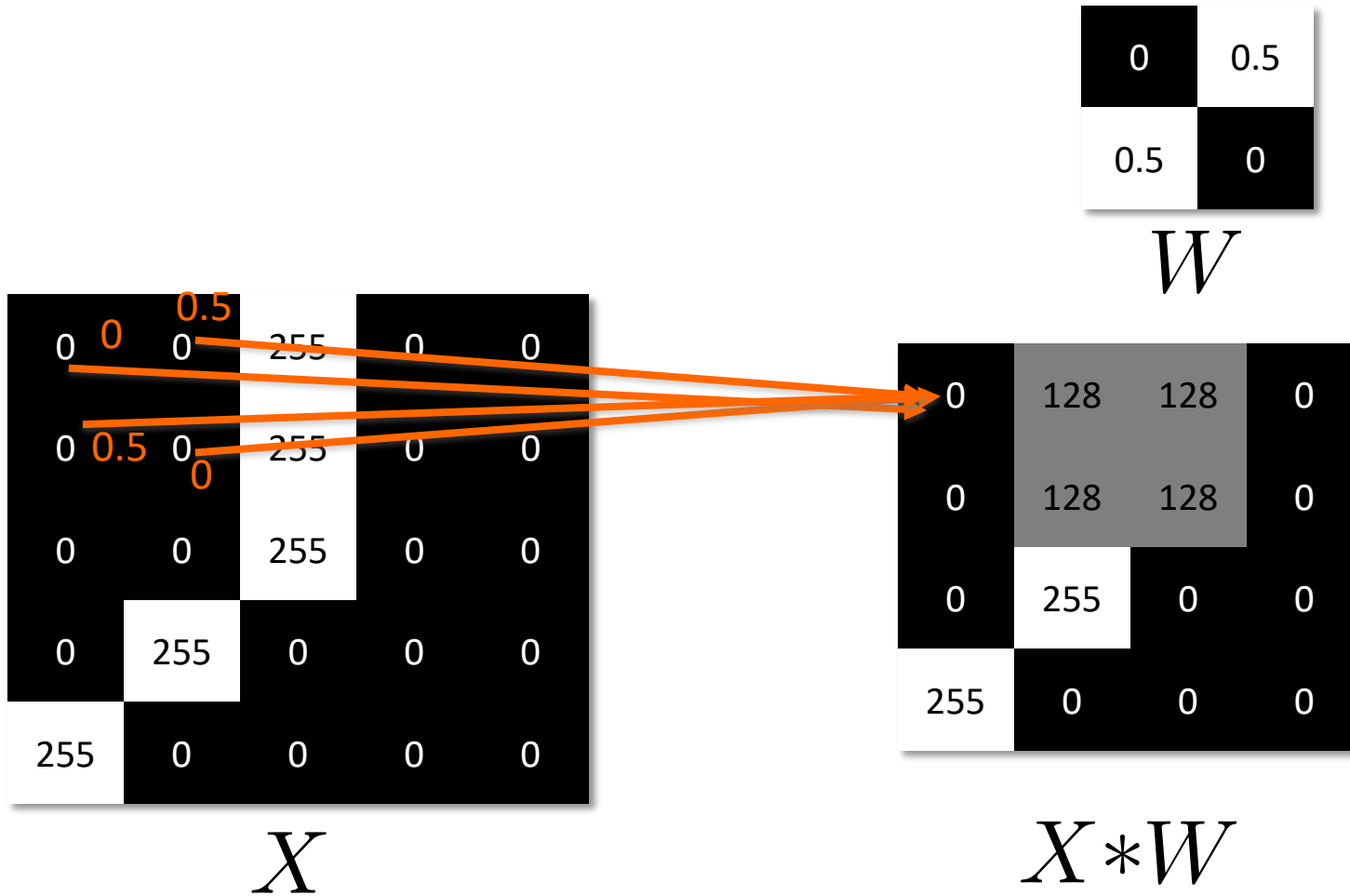
$$G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}$$

Corrélation 2D

- Appliquer un filtre gaussien dans un voisinage limité peut être vu comme l'application d'une **corrélation 2D**



Correlation 2D



Corrélation 2D

- Calculer H est l'équivalent de faire une corrélation avec le filtre $W = \text{array}([-1,1])$

The diagram illustrates the 2D correlation operation. It shows the input matrix X , the kernel W , and the resulting output matrix $X \star W$.

Input Matrix X :

0	0	255	0	0
0	0	255	0	0
0	0	255	0	0
0	255	0	0	0
255	0	0	0	0

Kernel W :

\star

-1	1
----	---

Output Matrix $X \star W$:

0	255	-255	0	0
0	255	-255	0	0
0	255	-255	0	0
255	-255	0	0	0
-255	0	0	0	0

Corrélation 2D

- Calculer V est l'équivalent de faire une corrélation avec le filtre $W = \text{array}([-1],[1])$

The diagram illustrates the 2D correlation operation. It shows the input matrix X , the kernel W , and the resulting output matrix $X \star W$.

Input Matrix X (5x5):

0	0	255	0	0
0	0	255	0	0
0	0	255	0	0
0	255	0	0	0
255	0	0	0	0

Kernel W (1x2):

-1
1

Output Matrix $X \star W$ (5x5):

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	255	-255	0	0
255	-255	0	0	0

Corrélation 2D sur un signal unitaire

						Padded f							
↙	Origin f												
	0	0	0	0	0		0	0	0	0	0	0	0
	0	0	0	0	0		0	0	0	0	0	0	0
	0	0	0	0	0	w	0	0	0	1	0	0	0
	0	0	1	0	0	1 2 3	0	0	0	0	0	0	0
	0	0	0	0	0	4 5 6	0	0	0	0	0	0	0
	0	0	0	0	0	7 8 9	0	0	0	0	0	0	0

(a) (b)

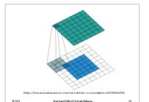
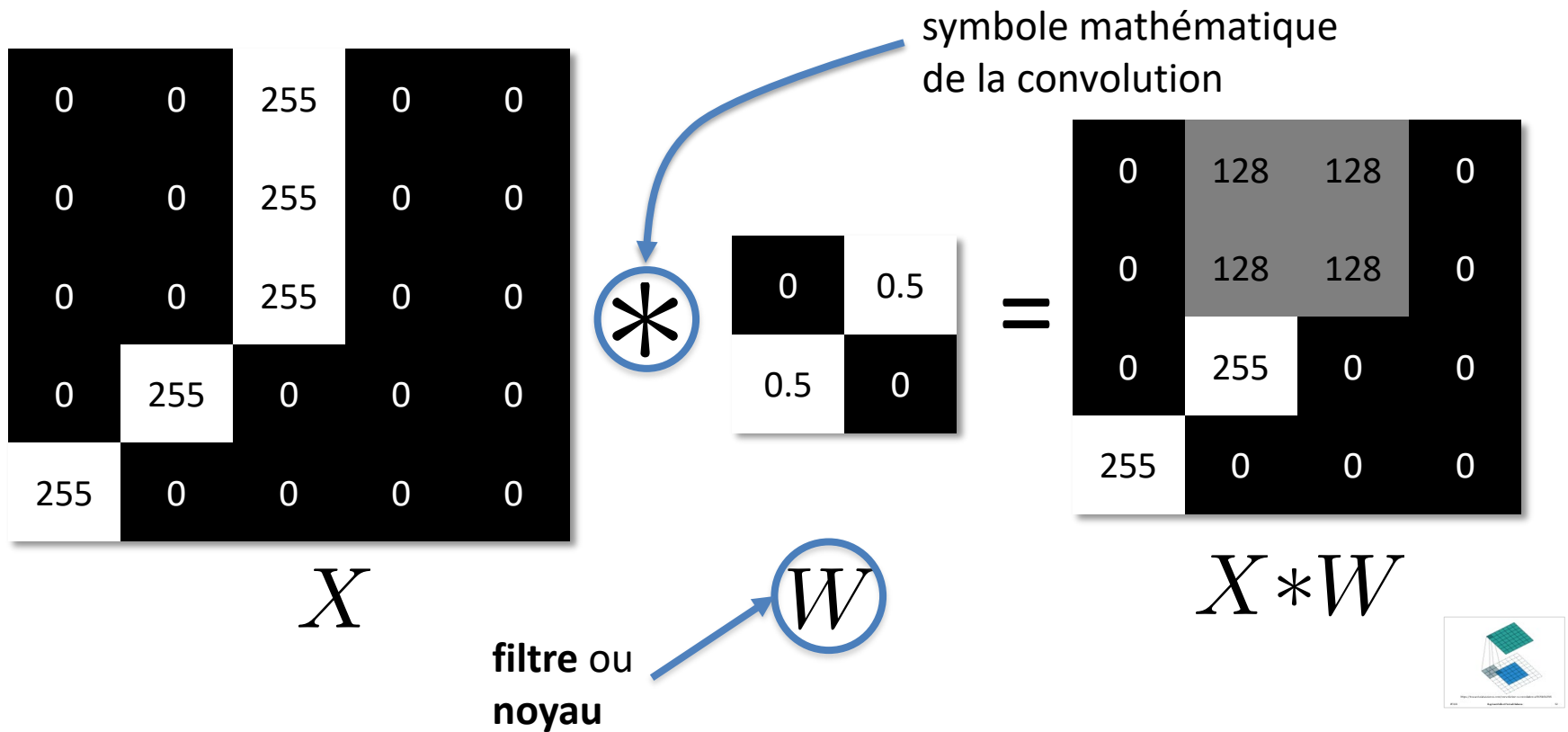
↙ Initial position for w						Correlation result					Full correlation result				
1 2 3	0	0	0	0	0						0	0	0	0	0
4 5 6	0	0	0	0	0		0	0	0	0	0	0	0	0	0
7 8 9	0	0	0	0	0		0	9 8 7	0		0	0	9 8 7	0	0
0	0	0	1	0	0	0	0	6 5 4	0		0	0	6 5 4	0	0
0	0	0	0	0	0	0	0	3 2 1	0		0	0	3 2 1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(c) (d) (e)

<https://towardsdatascience.com/convolution-vs-correlation-af868b6b4fb5>

Convolution 2D

- La **convolution 2D** revient à une corrélation qu'on appliquerait en prenant comme point de référence des indexes du filtre la dernière rangée et la dernière colonne



Conclusion

- Les technologies de vision par ordinateur sont de plus en plus performantes et de plus en plus répandues
 - ◆ Reconnaissance de caractères
 - ◆ Reconnaissance d'images
 - ◆ Recherche d'images
 - ◆ Conduite autonome
 - ◆ Analyse vidéo
 - ◆ Surveillance
 - ◆ Bien d'autres applications
- Plusieurs algorithmes basés sur des CNN : *ResNet*, *YOLO* et plus
- Plusieurs plateformes supporte ces algorithmes: *Anaconda*, *Scikit Learn*, *Amazon SageMaker*, *Microsoft Azure ML*, *Google Vertex AI* et plus

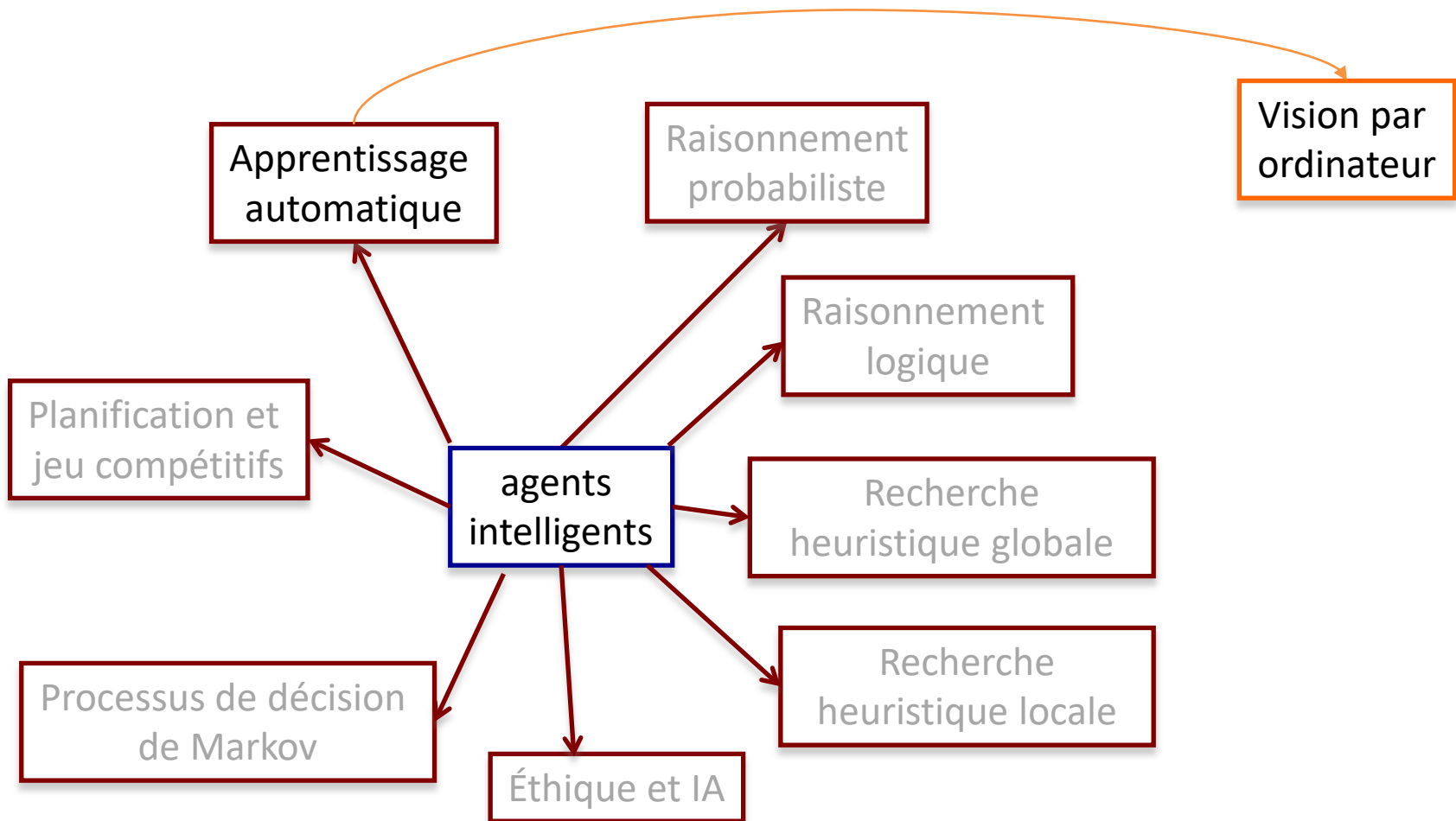
Conclusion

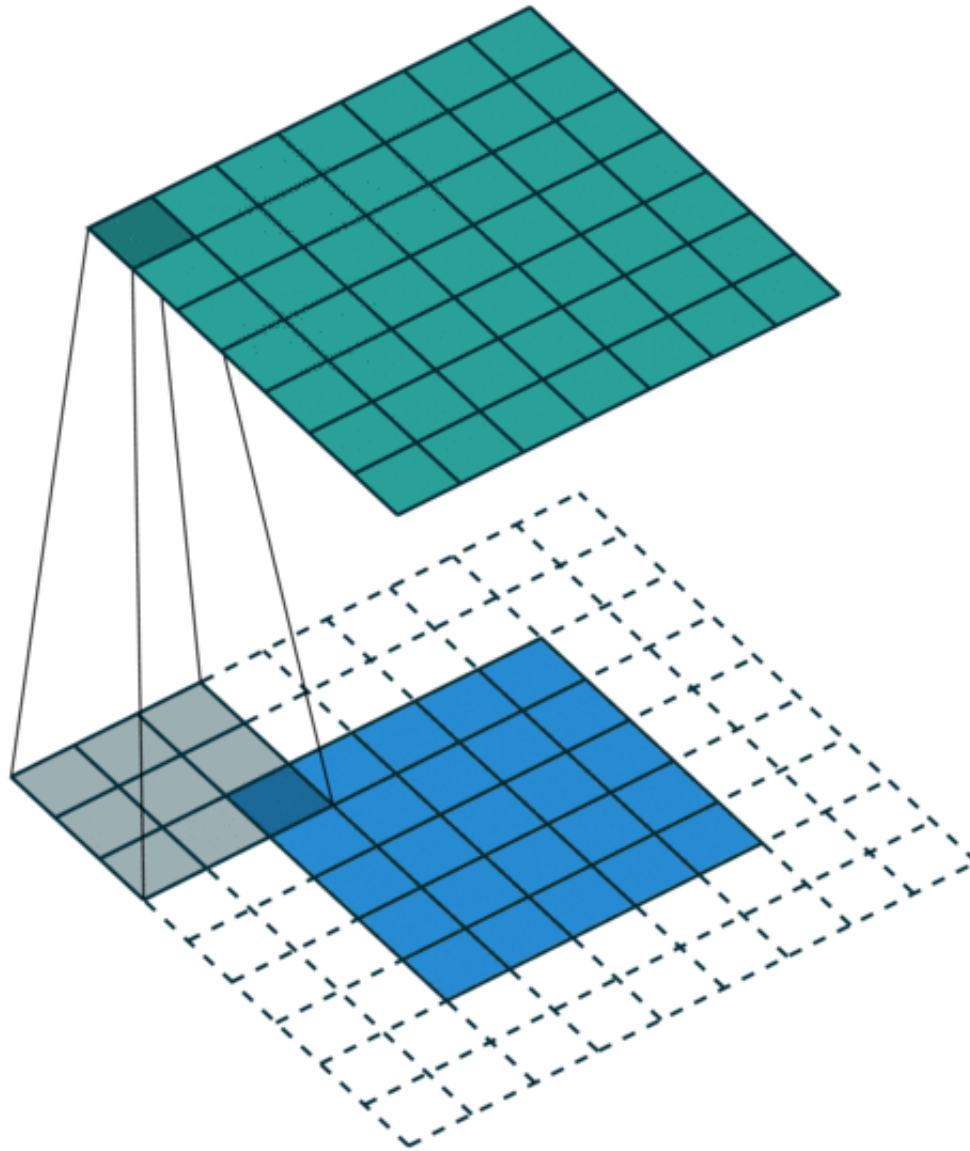
- Ce cours ne donne qu'une introduction à la vision par ordinateur
- Cours plus avancés:
 - ◆ Le bacc en imagerie offre plusieurs cours sur le sujet (ex.: **IMN 559 - Vision par ordinateur**)
 - » ces cours peuvent être suivis à la maîtrise...
 - ◆ **IFT 603 – Techniques d'apprentissage**
 - ◆ **IFT 725 – Réseaux neuronaux** : couvre les réseaux à convolution avec plus de détails (cours de maîtrise)

Sujets couverts

Concepts et algorithmes

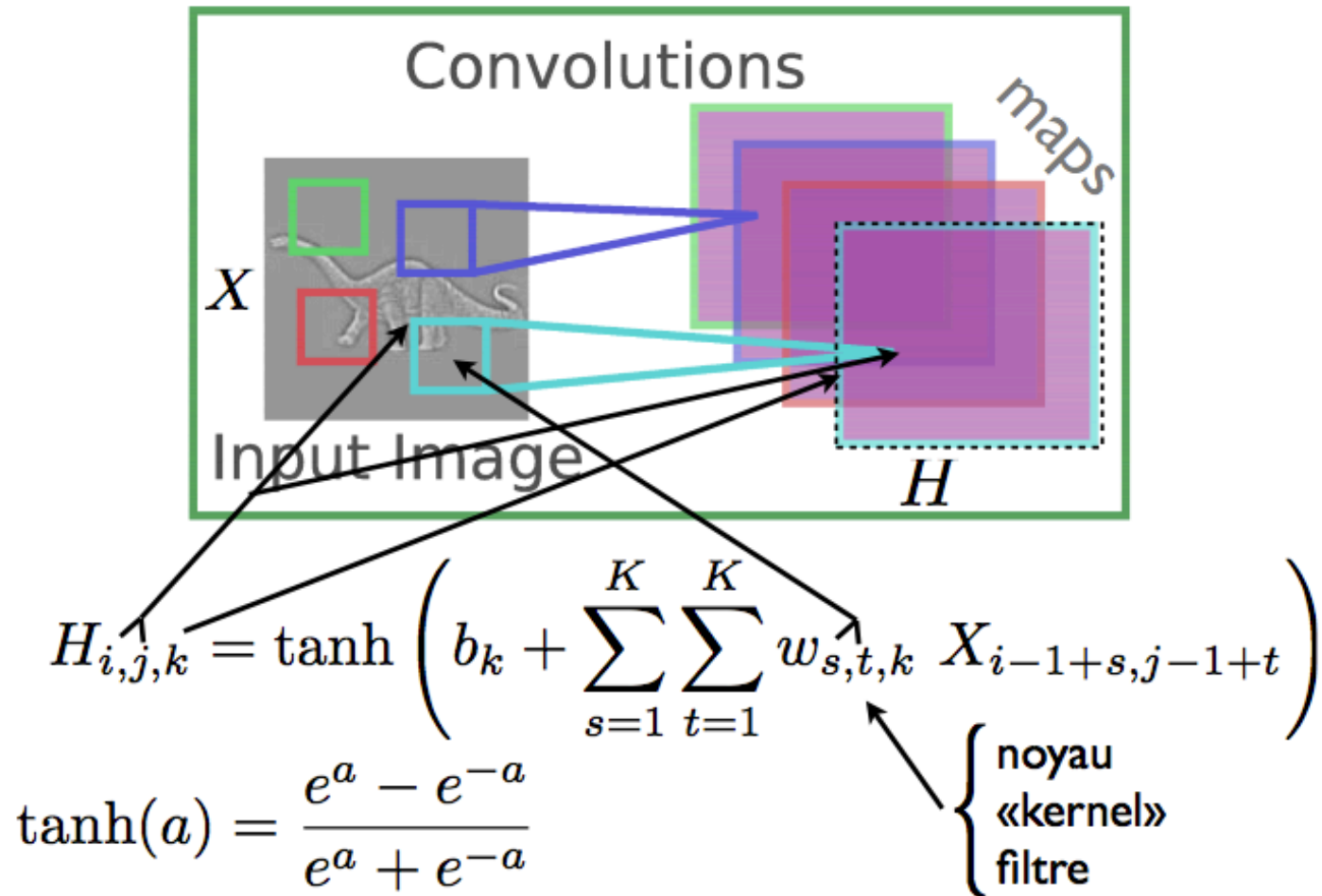
Applications



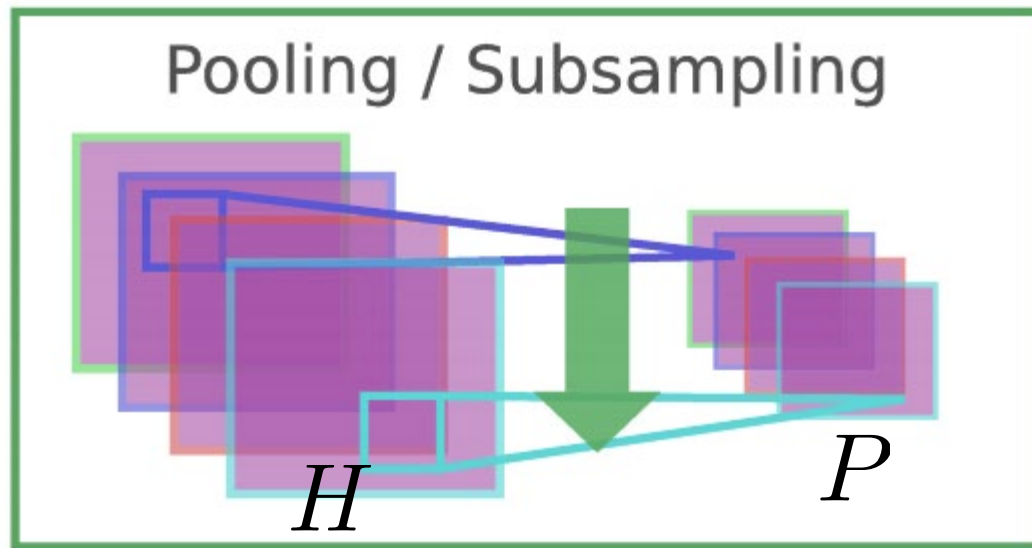


<https://towardsdatascience.com/convolution-vs-correlation-af868b6b4fb5>

Réseau de neurones à convolution: couche à convolution



Réseau de neurones à convolution: couche à « pooling »

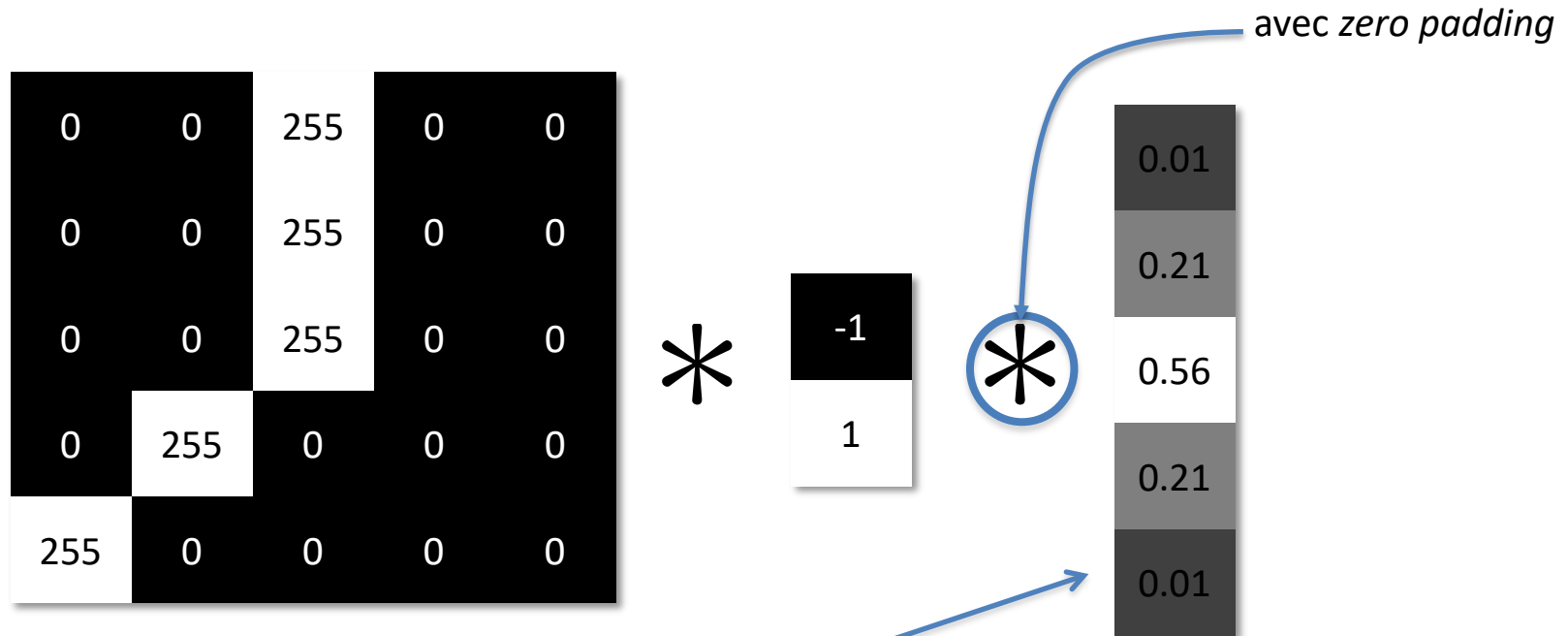


- Deux étapes

- ◆ «max pooling»:
$$P_{i,j,k} = \max_{(i',j') \in N(i,j)} H_{i',j',k}$$
- ◆ «downsampling»: garde seulement une fraction des neurones

Bruit dans le calcul du gradient d'image

- Pour éliminer la détection de ces faux contours, on applique une deuxième convolution pour **lisser** le résultat



- Le filtre utilisé est appelé **filtre gaussien**

Bruit dans le calcul du gradient d'image

- Pour éliminer la détection de ces faux contours, on applique une deuxième convolution pour **lisser** le résultat

0.01	$= \exp(-2^2) / Z$
0.21	$= \exp(-1^2) / Z$
0.56	$= \exp(-0^2) / Z$
0.21	$= \exp(-1^2) / Z$
0.01	$= \exp(-2^2) / Z$

constante de normalisation

$$Z = \exp(-2^2) + \exp(-1^2) + \exp(-0^2) + \exp(-1^2) + \exp(-2^2)$$

Bruit dans le calcul du gradient d'image

- Pour éliminer la détection de ces faux contours, on applique une deuxième convolution pour **lisser** le résultat

0.01	= $\exp(-2^2) / Z$
0.21	= $\exp(-1^2) / Z$
0.56	= $\exp(-0^2) / Z$
0.21	= $\exp(-1^2) / Z$
0.01	= $\exp(-2^2) / Z$

Formule générale du filtre gaussien

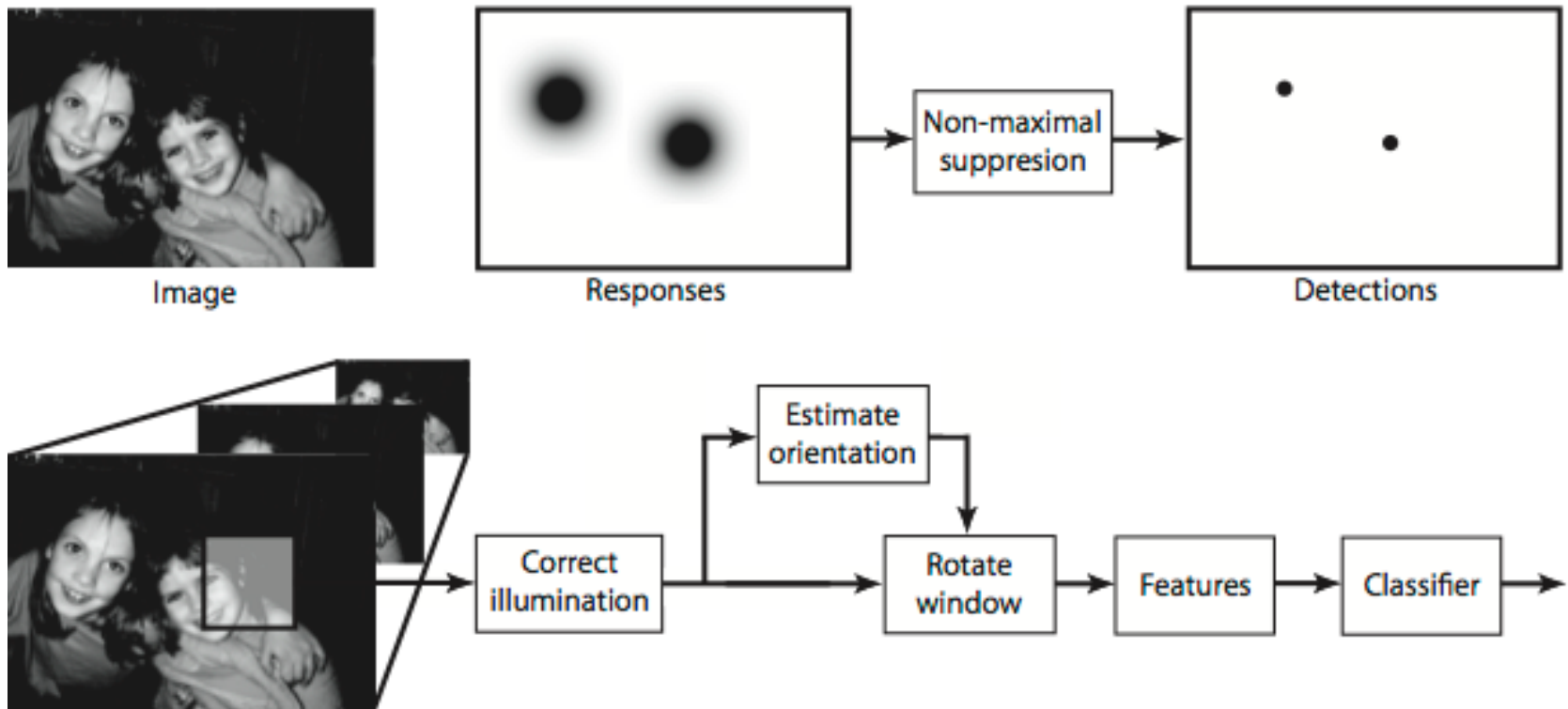
$$W[i,j] = \exp(-d(i,j)^2/\sigma^2)/Z$$

$d(i,j)$ = distance p/r au centre du filtre

σ = paramètre de lissage
(plus il est grand, plus on lisse)

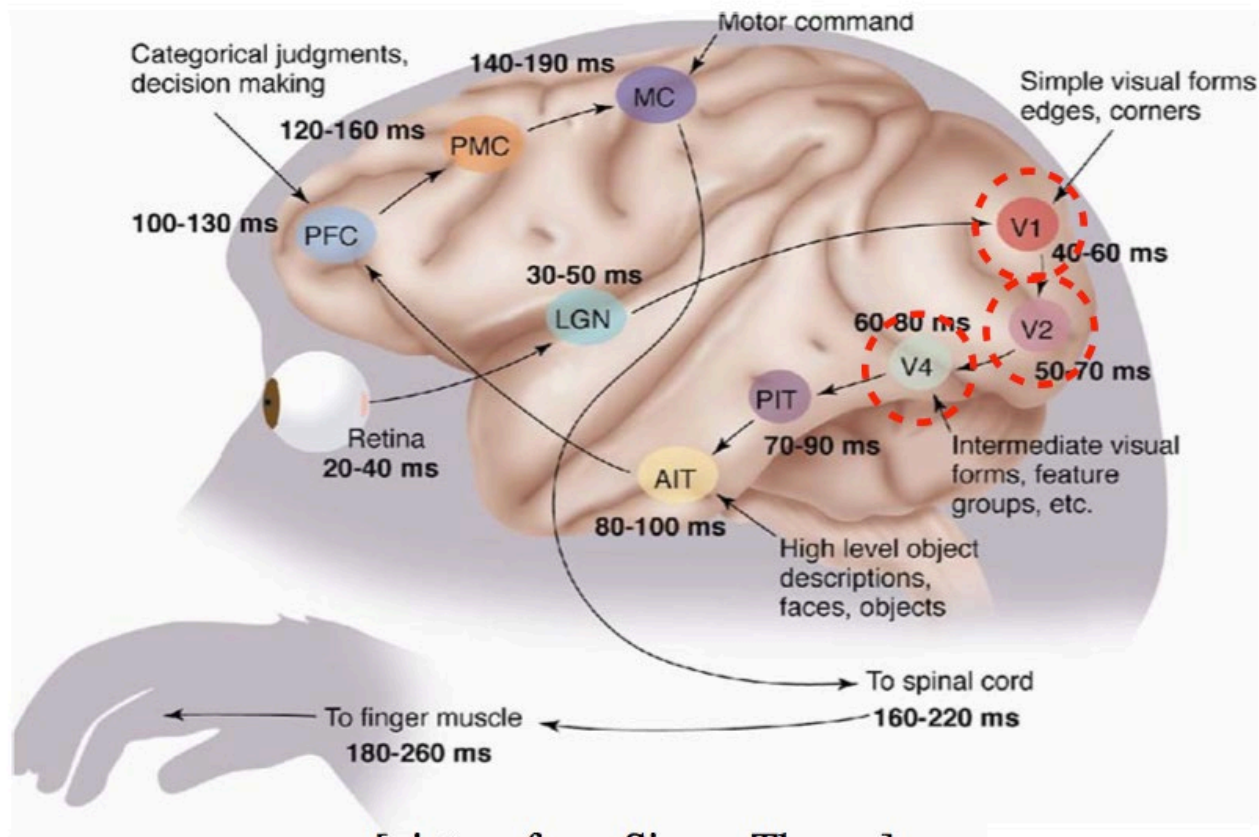
De la classification à la détection

- Quoi faire si ce que l'on cherche n'est pas au centre de l'image?
- **Idée générale:** on applique le même classifieur à plusieurs positions et échelles dans l'image

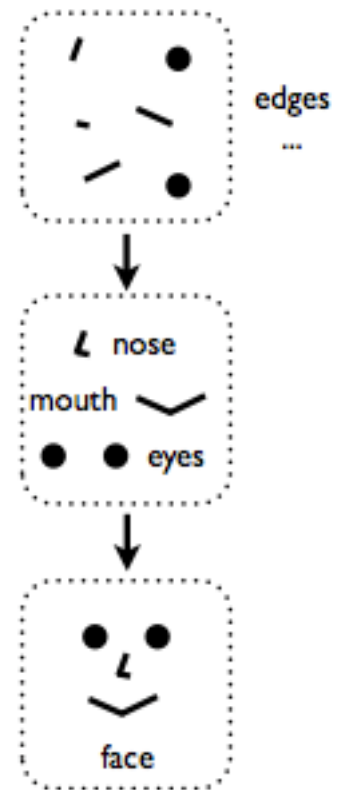


System visuel humain

- Pourquoi ne pas s'inspirer du cerveau pour faire de la vision!



[picture from Simon Thorpe]



Caractéristiques d'images

- En plus de servir à détecter des contours, les gradients d'image peuvent servir à **extraire des caractéristiques** d'une image
- On a vu que l'orientation des gradients ne varie pas en fonction de l'intensité
 - ◆ on pourrait utiliser cette propriété pour obtenir des caractéristiques invariantes p/r à l'intensité d'une image (ex.: l'illumination)

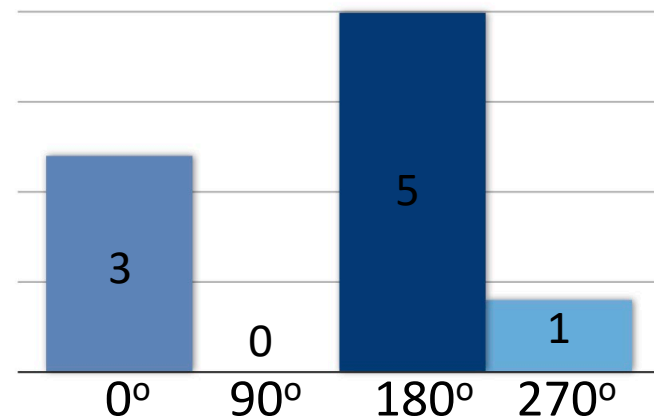
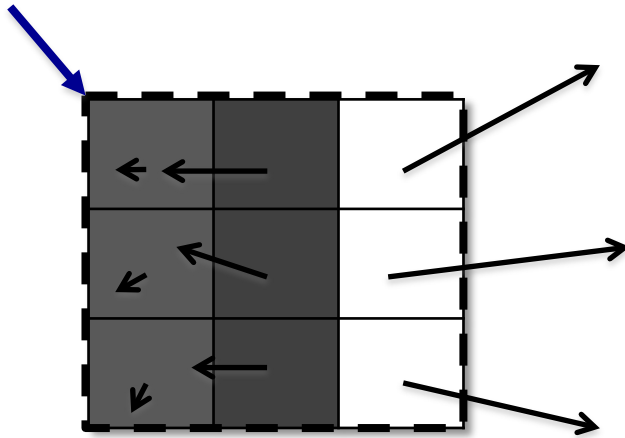
Histogramme de gradients

- Un type de caractéristiques populaire est l'**histogramme de gradients** (*histogram of gradients* ou **HoG**)
 - ◆ calculer le **champ de gradients** de l'image
 - ◆ partitionner (diviser) l'image en plusieurs **segments** (*cells*)
 - ◆ dans chaque segment, faire un **histogramme** des orientations des gradients contenus dans ce segment
 - ◆ le vecteur de caractéristiques pour l'image est la concaténation de tous ces histogrammes

Histogramme de gradients

- Pour calculer un histogramme d'orientations
 - ◆ on partitionne les orientations possibles en quelques cases (ex. 4 cases à 0° , 90° , 180° et 270° , ou 8 cases à 0° , 45° , ..., 315°)
 - ◆ la valeur de chaque case est le compte du nombre de gradients qui tombent dans chaque case
 - » chaque gradient « vote » pour l'orientation la plus proche

segment (cell)



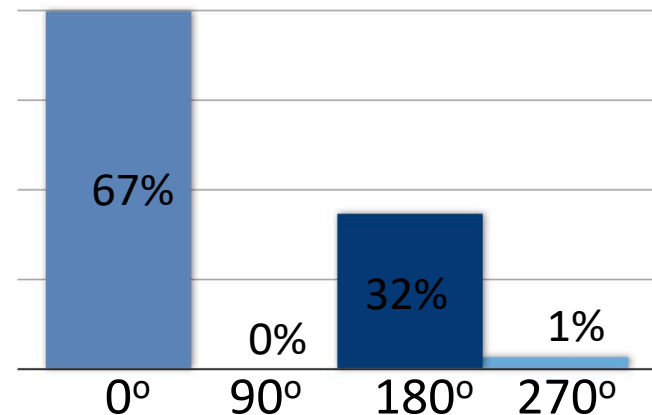
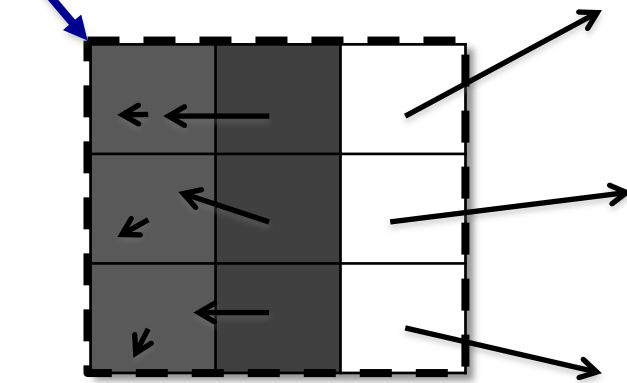
Histogramme de gradients

- Façon de tenir compte de la grandeur (norme) du gradient
 - ◆ on peut ajouter une case additionnelle pour les gradients dont la norme est sous un seuil donné
 - ◆ on utilise un poids c du vote d'un gradient $G[i,j,:]$ comme suit

$$c[i,j] = \text{sqrt}(\text{sum}(G[i,j,:]^2)) / N$$

segment (cell)

où N est la somme des normes dans le segment où se trouve $G[i,j,:]$



Histogramme de gradients

- Exemple sur une image de personne



Image



Orientation
histograms

illustre la
perpendiculaire
du gradient, pour
faciliter
la visualisation

Histogramme de gradients

- On pourrait traiter le problème de reconnaissance d'objets comme un problème de classification standard en apprentissage automatique
 - ◆ entrée x_t : représentation HoG d'une image
 - ◆ cible y_t : présence ($y_t=1$) ou absence ($y_t=0$) d'un objet à reconnaître
- On peut ainsi collecter un ensemble d'entraînement à donner à un algorithme d'apprentissage pour la classification
(Perceptron, régression linéaire, réseau de neurones, etc.)

Histogramme de gradients

- Visualisation des orientations importantes, apprises par un classifieur linéaire



Image



Orientation
histograms

Visualisation des poids du classifieur



poids
positifs



poids
négatifs