

Chapitre 8 – Introduction à la vision par ordinateur*

Joël Lefebvre (UQÀM)

INF600F – Traitement d'images

Automne 2024

Survol du cours

- Introduction à la vision par ordinateur
- Réseau de neurones à convolution (CNN)
- Tâches de vision
- Examen final

Annonces

- **Laboratoire** : Révision pour l'examen
- **TP4** : remise avant le dimanche 15 décembre avant 23h59
- **Examen final**
 - Mercredi, 11 décembre entre 14h à 17h
 - 1 feuille de note manuscrite de format Lettre (8 ½ x 11)
 - **Plus de détails à la fin du cours**

Vision par ordinateur

« *In the 60s, Marvin Minsky assigned a couple of undergrads **to spend the summer** programming a computer to use a camera to identify objects in a scene. He figured they'd have the problem solved by the end of the summer. **Half a century later, we're still working on it.** » (XKCD)*



IN CS, IT CAN BE HARD TO EXPLAIN
THE DIFFERENCE BETWEEN THE EASY
AND THE VIRTUALLY IMPOSSIBLE.

Plusieurs niveaux de vision

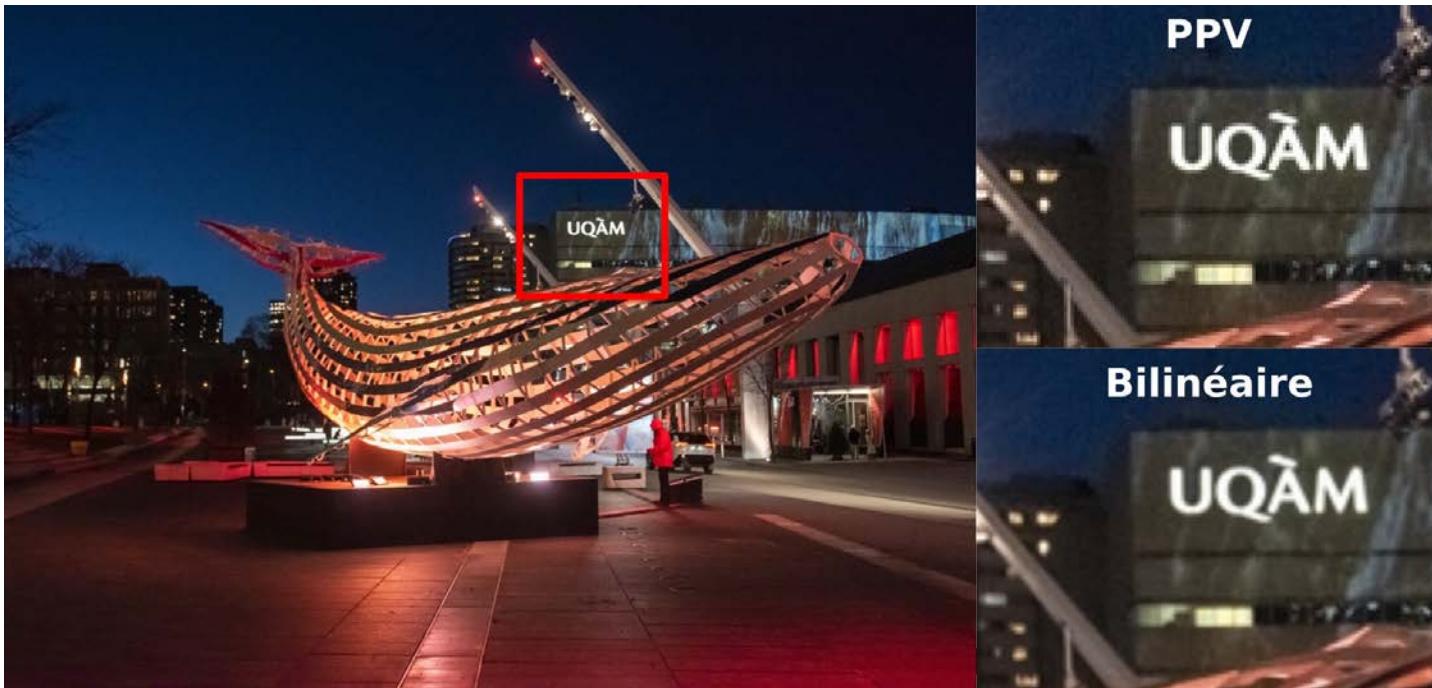
- Vision de bas niveau
 - **C'est ce qu'on a étudié cette session !**
- Vision de niveau intermédiaire
- Vision de haut niveau

Que voit un ordinateur ?

- Une **matrice** 2D (ton de gris) ou 3D (couleur) !



Bas niveau : redimensionnement



Bas niveau : ajustements d'image

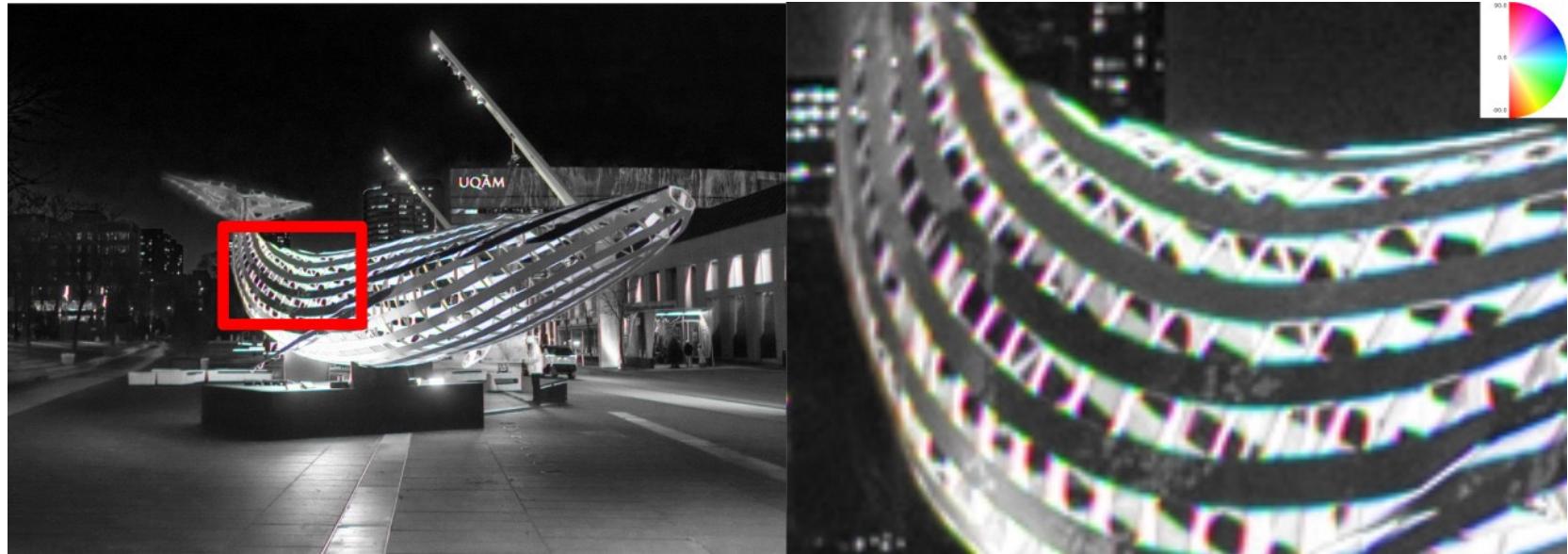


Exemple d'opérations ponctuelles, de filtres, etc.

Bas niveau : Contours



Bas niveau : Gradients orientés



Bas niveau : Segmentation (Couleur)



Postérisation d'une image couleur en 3 niveaux par canal

Application : Sports

- HowStuffWorks : « How the First-Down Line Works » ([Lien](#))



Application : Segmentation et matage (matting)



(a)



(b)



(c)



(d)



(e)

Matage d'une image naturelle. Source: (Szeliski, 2021, Fig. 10.40)

Application : Segmentation et matage (video)

- Lin, S., Ryabtsev, A., Sengupta, S., Curless, B., Seitz, S. and Kemelmacher-Shlizerman, I. (2020). **Real-Time High-Resolution Background Matting.** <https://arxiv.org/abs/2012.07810>



Application : Biométrie

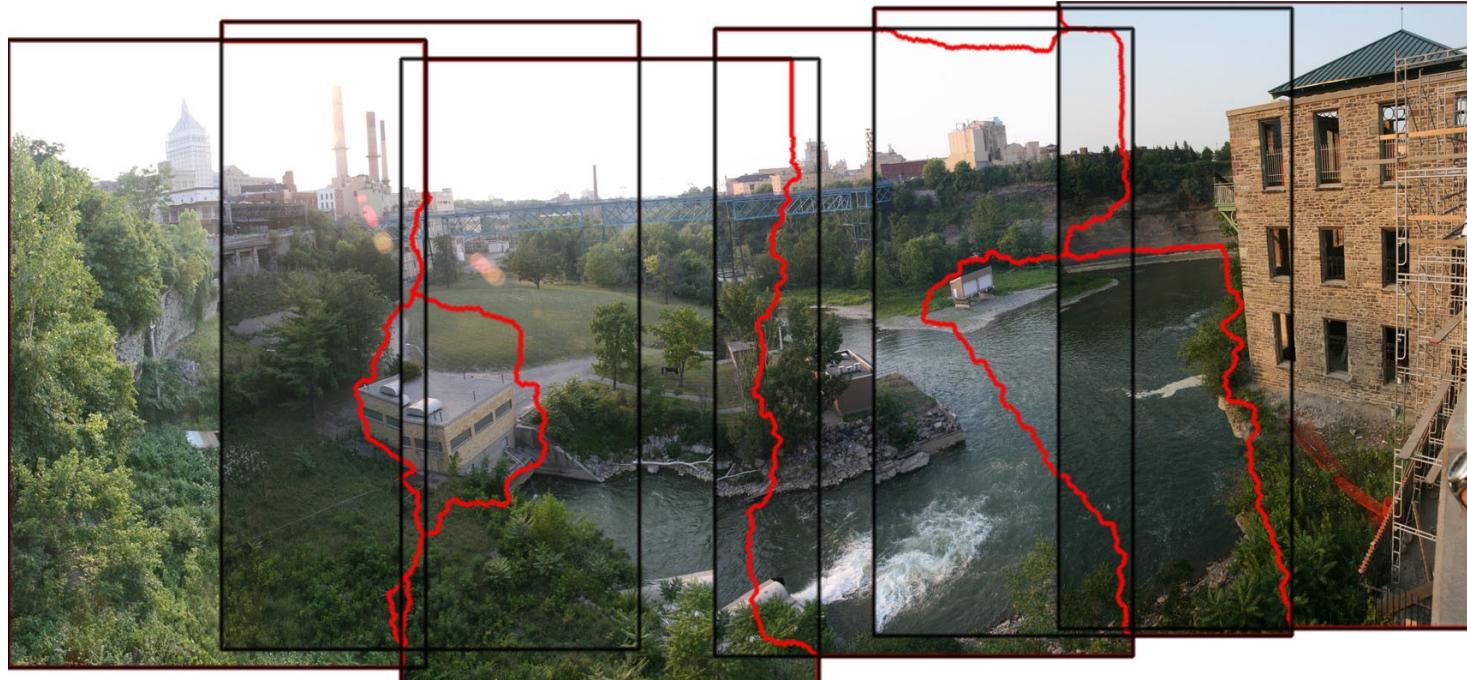
- « How the Afghan Girl was Identified by Her Iris Patterns » ([Lien](#))
- National Geographic : Finding the Afgan Girl ([YouTube](#))



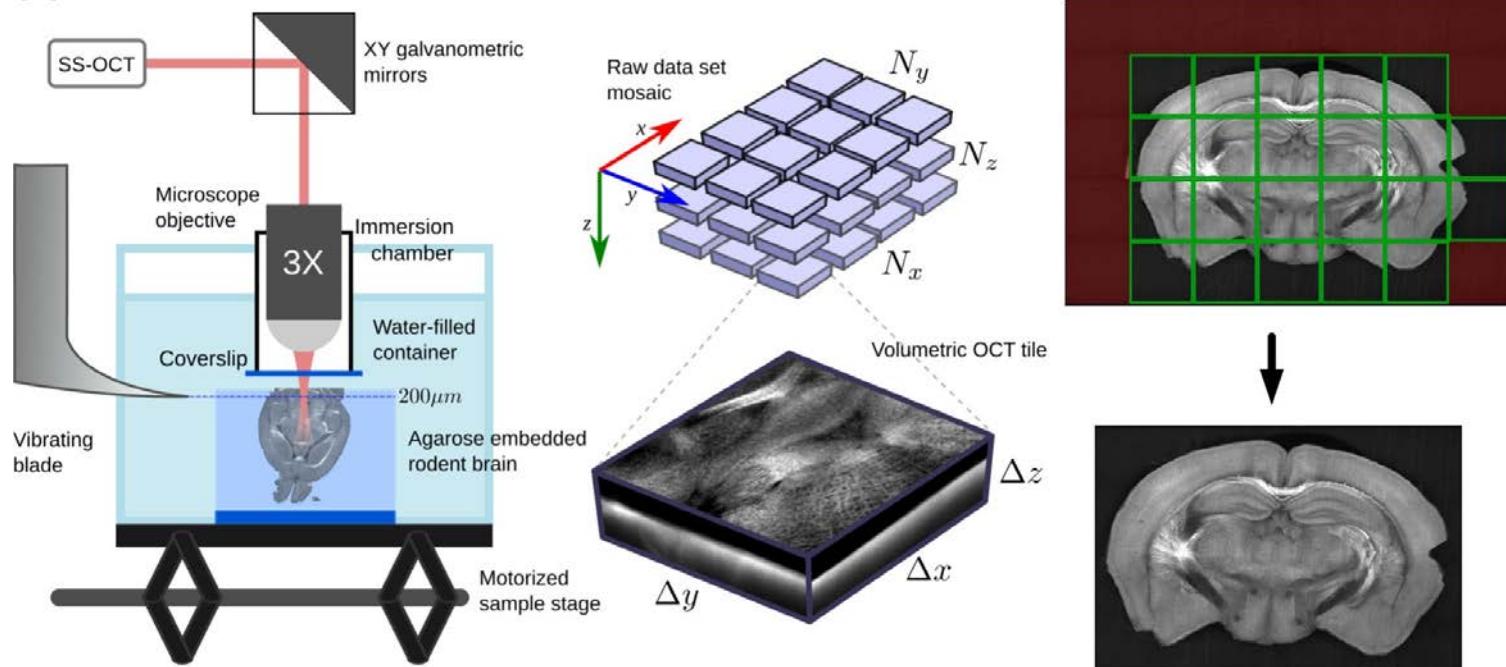
Vision de bas niveau : Résumé

- Image → Image
- Manipulation de photos
- Extraction de caractéristiques
- Étape initiale des pipelines de traitement et de vision

Vision de niveau intermédiaire : assemblage panoramique



Niveau intermédiaire : histologie sérielle 3D



Reconstruction de mosaïques 3D pour l'histologie sérielle ([Lefebvre2017](#))

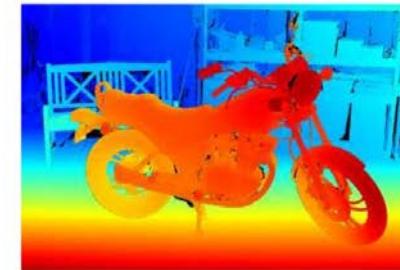
3D, profondeur et stéréo



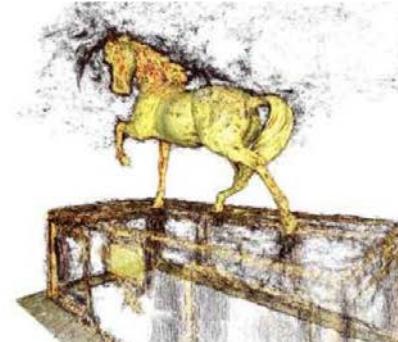
(a)



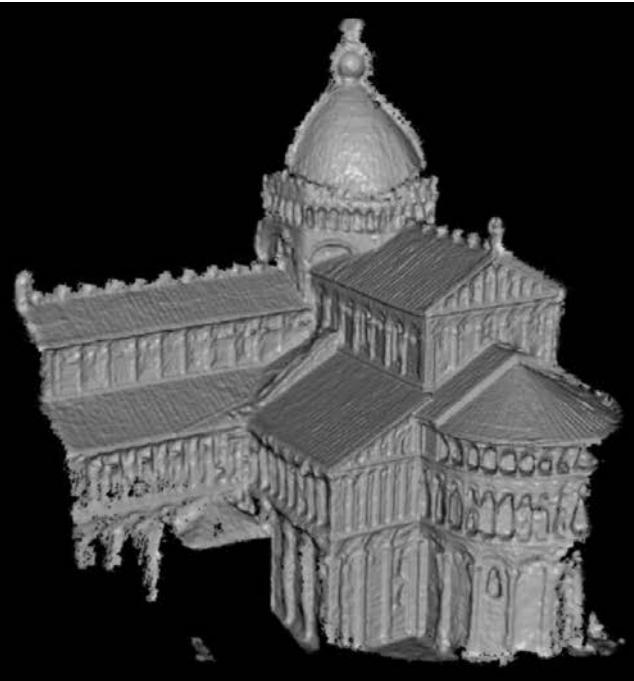
(b)



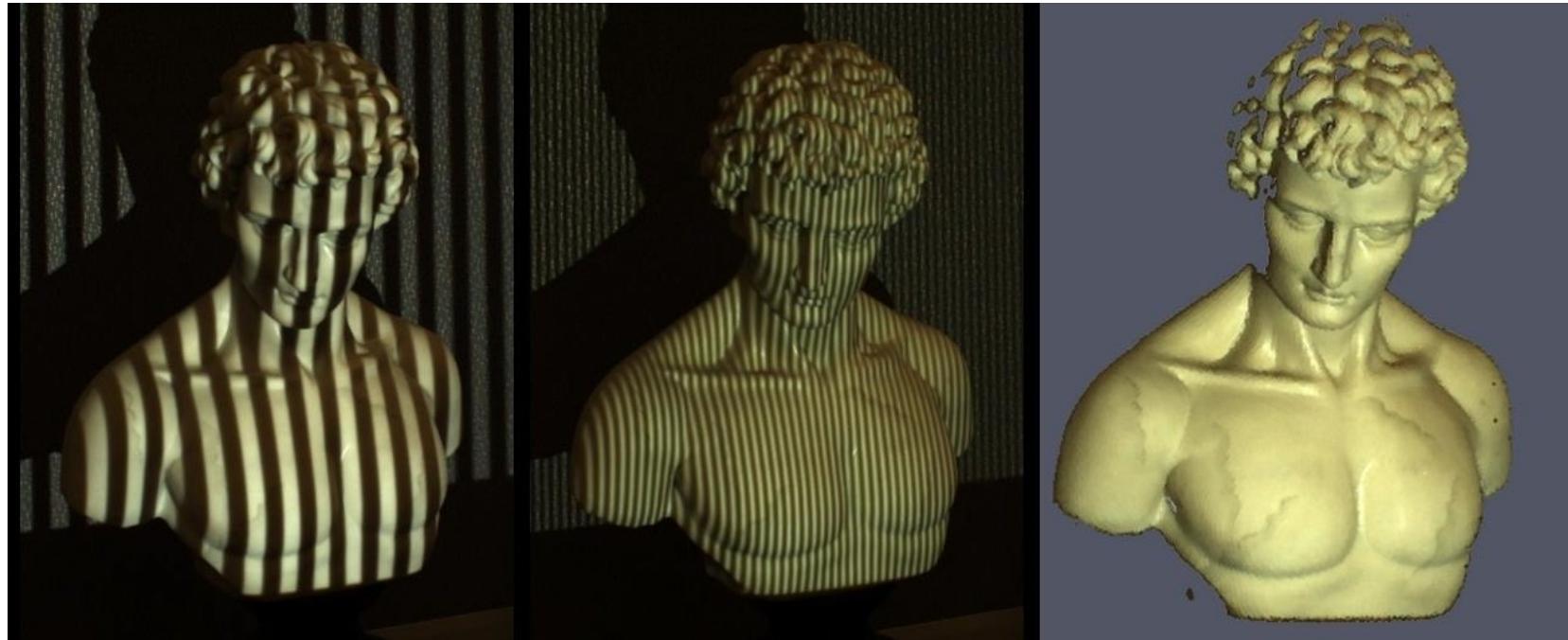
(c)



Niveau intermédiaire : reconstruction 3D à partir de plusieurs images

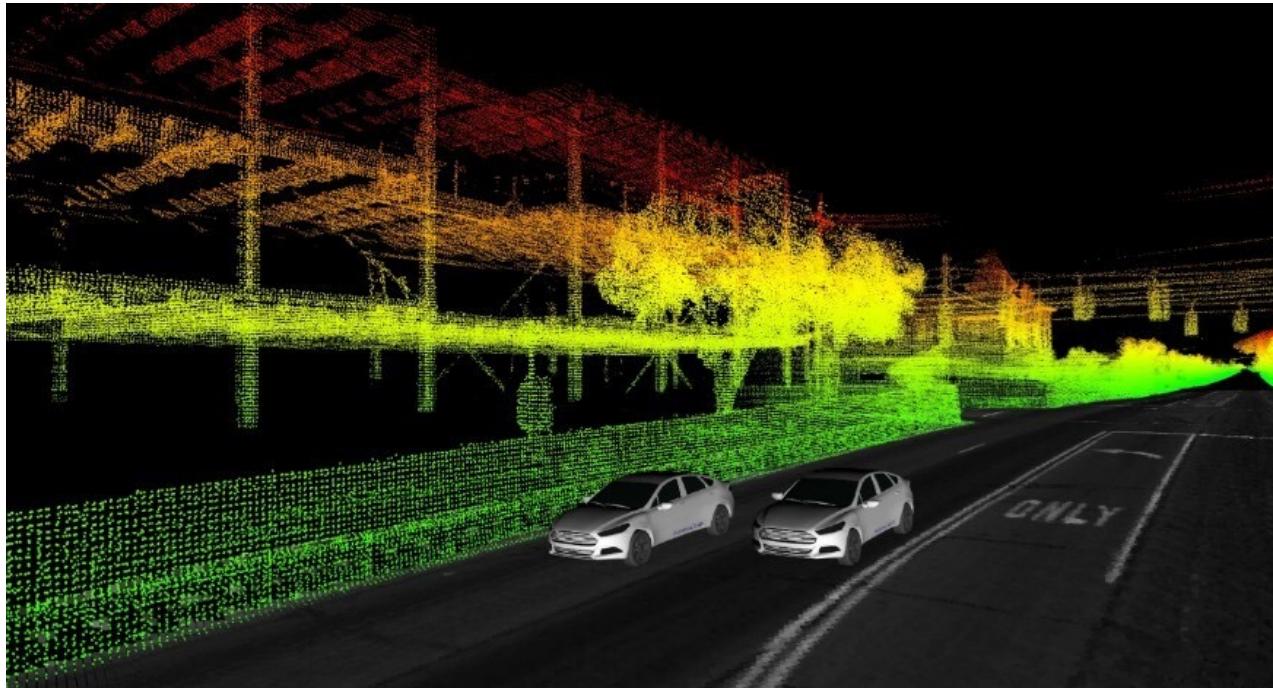


Niveau intermédiaire : lumière structurée



Reconstruction d'une forme 3D en utilisant une illumination structurée (Source : [Instructables](#))

Niveau intermédiaire : télémétrie



Niveau intermédiaire : flux optique

- « Le mouvement apparent des objets, surfaces et contours d'une scène visuelle, causé par le mouvement relatif entre un observateur (l'oeil ou une caméra) et la scène. »
[\(Wikipedia\)](#)
- <https://joe-from-mtl.github.io/teaching/demos/opticalFlowDense.html>



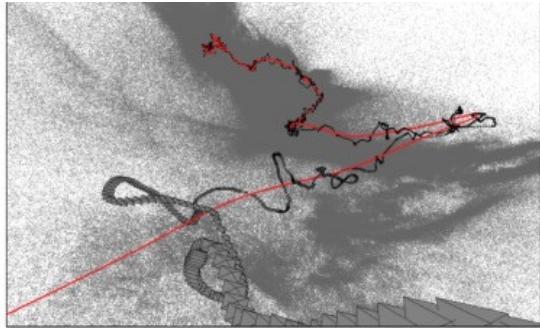
Niveau intermédiaire : time lapse



Timelapse à Seattle générée à partir d'une caméra 360. ([YouTube](#))

Niveau intermédiaire : Hyperlapse

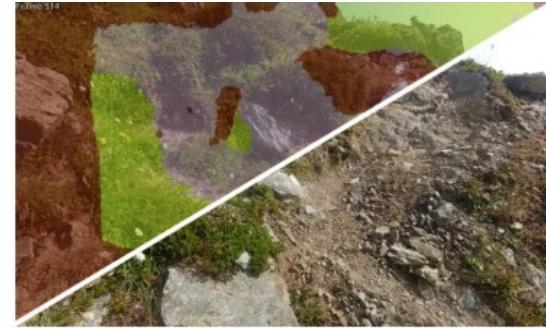
- Conversion d'un vidéo de point de vue saccadé (*first person*) en *time lapse* fluide ([YouTube](#))



(a) Scene reconstruction



(b) Proxy geometry

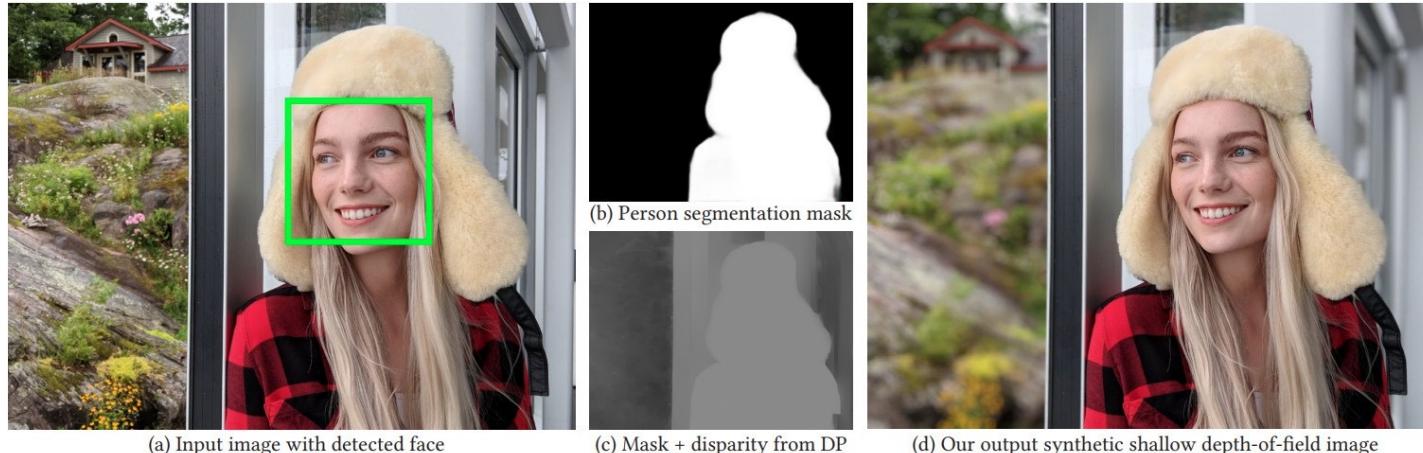


(c) Stitched & blended

Figure 1: Our system converts first-person videos into hyper-lapse summaries using a set of processing stages. (a) 3D camera and point cloud recovery, followed by smooth path planning; (b) 3D per-camera proxy estimation; (c) source frame selection, seam selection using a MRF, and Poisson blending.

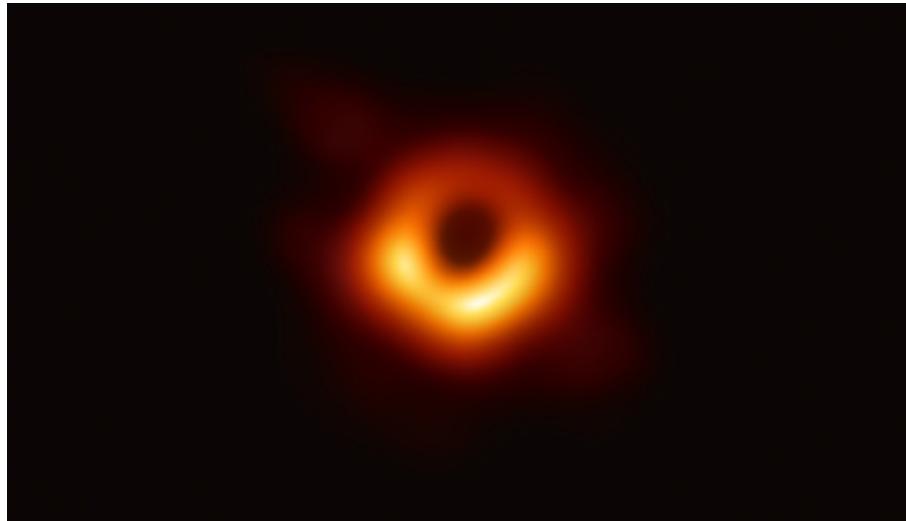
Application : Photographie computationnelle

- Wadhwa, N., et al (2018). Synthetic Depth-of-Field with a Single-Camera Mobile Phone. *arXiv*. (Google Research).
<https://doi.org/10.1145/3197517.3201329>



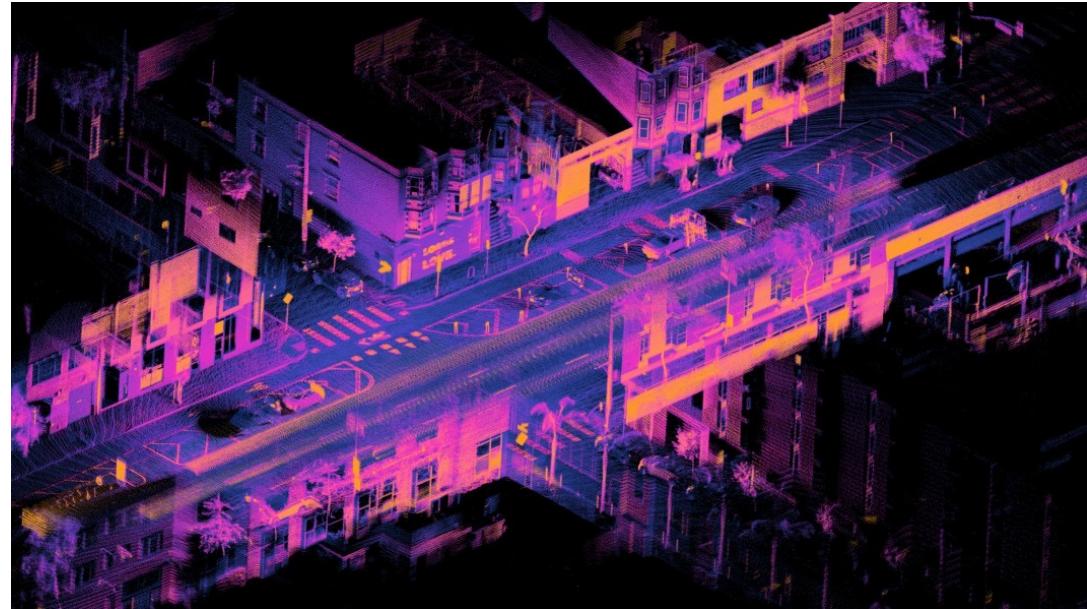
Application : Ouverture encore plus large . . .

- NASA : Comment les scientifiques ont capturé la première image d'un trou noir ([Lien](#))



Vision de niveau intermédiaire : Domaines

- Image ↔ Image
- Image ↔ Monde
- Image ↔ Temps



Vision de haut niveau : Classification

- Qu'y a-t-il dans l'image ?



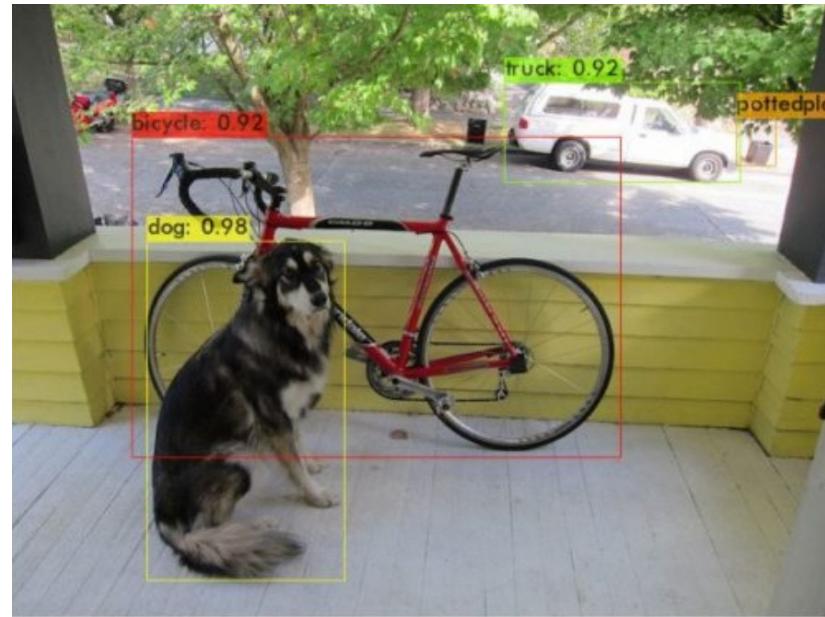
Vision de haut niveau : Étiquetage (tagging)

- Quelles sont **toutes** les choses dans l'image ?



Vision de haut niveau : Détection

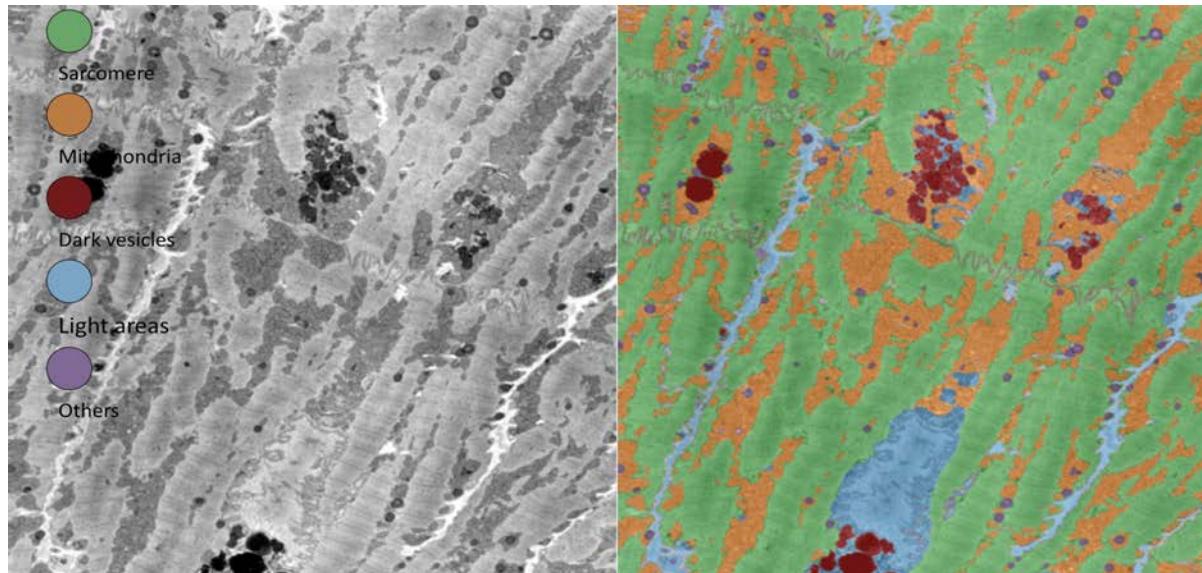
- Quelles sont **toutes** les choses dans l'image ?
- Où sont-elles ?



Exemple de détection d'objets avec la méthode YOLO (Source : datacorner.fr)

V. haut niveau : Segmentation sémantique

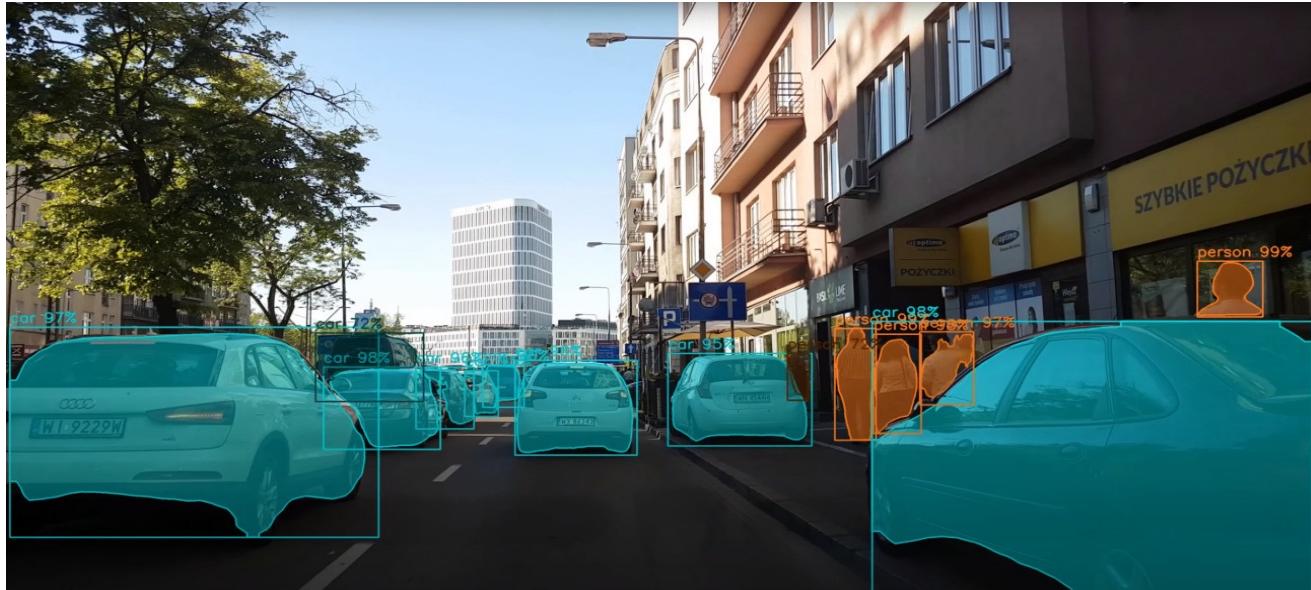
- Quelle est la classe / catégorie de chaque pixel dans l'image ?



Segmentation sémantique d'un volume acquis par microscopie électronique

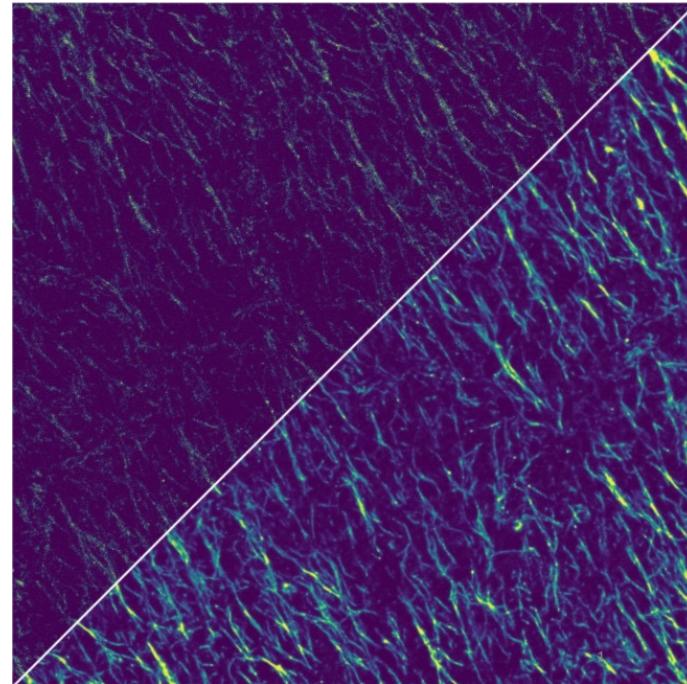
V. haut niveau : Segmentation des instances

- Où sont **toutes** les choses séparées dans l'image ?



Vision de haut niveau : Plusieurs autres tâches

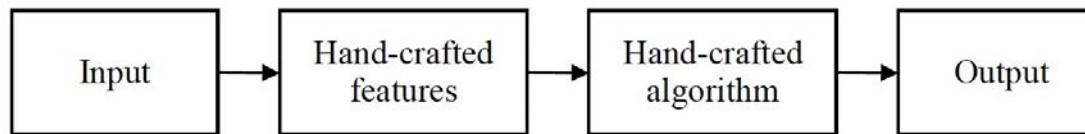
- Image → Sémantique
 - Classification d'images
 - Détection d'objets
 - Segmentation
- Applications
 - 3D à partir d'une seule image
 - Reconstruction
 - Débruitage / Super-résolution
 - Récupération d'images
 - Rotobotique
 - Et bien d'autres !



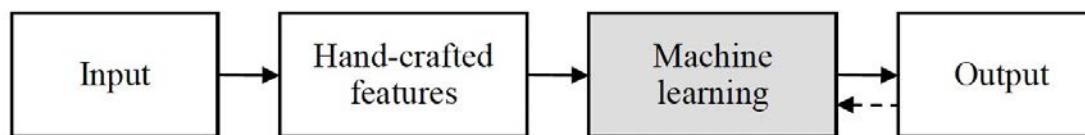
Réseau de neurones à convolution (CNN)

INF600F – Traitement d'images

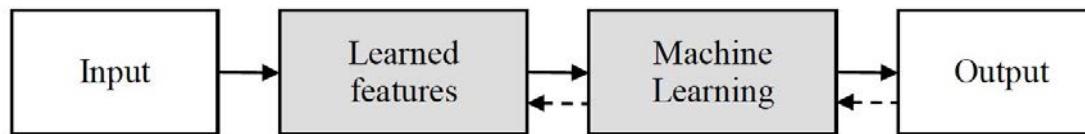
Types de pipeline de vision



(a) Traditional vision pipeline

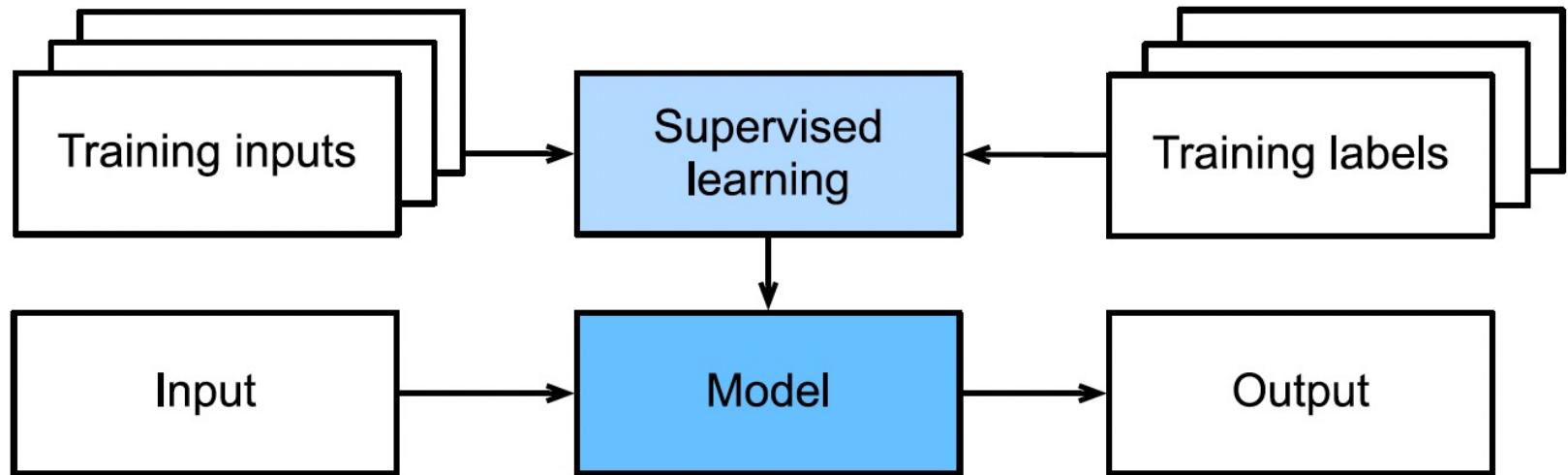


(b) Classic machine learning pipeline



(c) Deep learning pipeline

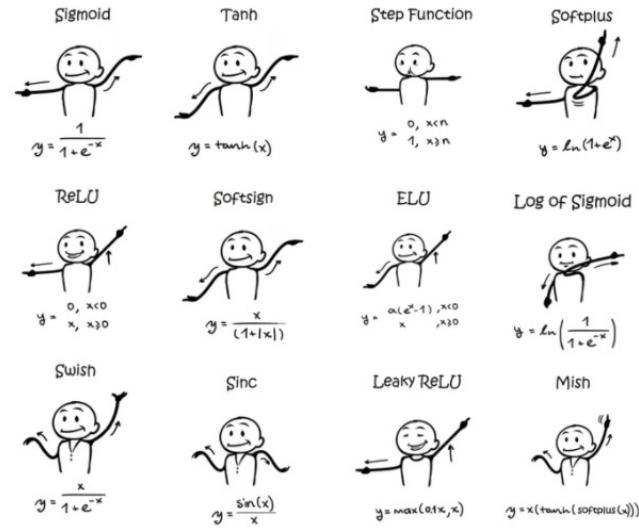
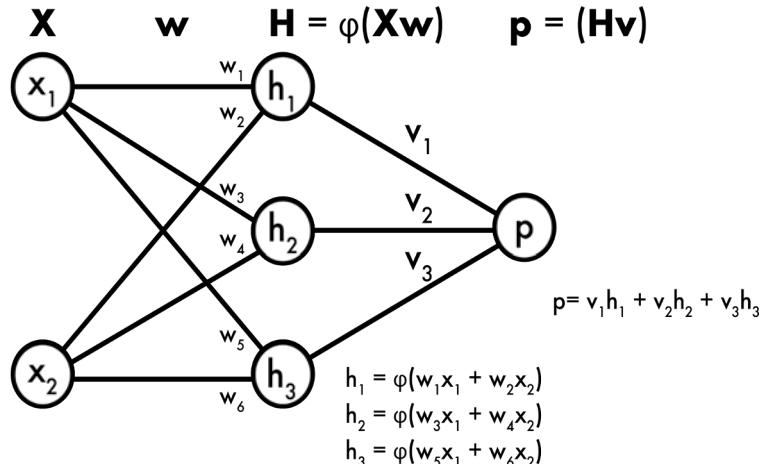
Apprentissage supervisé



Réseau de neurones simple

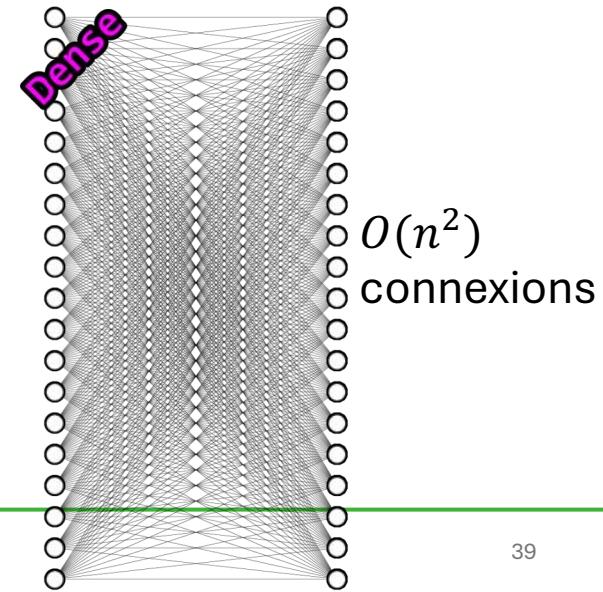
- Réseau à 1-couche cachée, activation sigmoïde pour la couche cachée, sortie linéaire

$$F(X) = \varphi(Xw)v$$



Trop de poids à apprendre !

- Les réseaux de neurones sont **densément connectés**.
- **Mais est-ce vraiment ce que nous voulons lors du traitement des images ?**



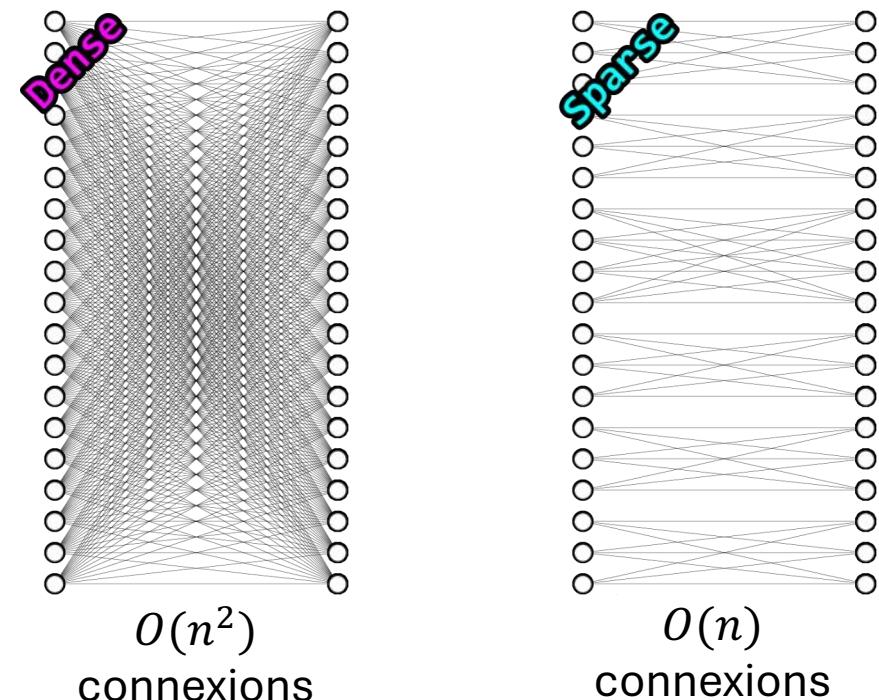
Réseaux de neurones et images

- Les réseaux de neurones sont **densément connectés**. Chaque neurone de la couche i est connecté à chaque neurone de la couche $i + 1$
- **Caractéristiques HOG** : 36 caractéristiques / patch de taille 8×8
- Disons qu'on souhaite traiter des **images en couleur** :
 - **Entrée** : image RGB de taille $256 \times 256 \times 3$
 - **Couche cachée** : carte de caractéristiques (*feature map*) $32 \times 32 \times 36$
 - **Sortie** : 1000 classes
- Entrée → Couche cachée représente **7.2 milliards de connexions !**
- Par comparaison, un **cerveau humain** possède en **moyenne 86 milliards de neurones**, et seulement une portion de celles-ci est utilisée pour la vision*.

*Herculano-Houzel, S. (2012). The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost. *Proceedings of the National Academy of Sciences*, 109(Supplement 1), 10661–10668.

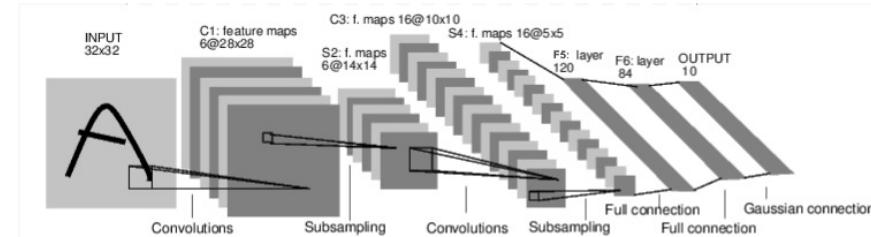
Trop de poids à apprendre ! Une solution ?

- Pour la vision, on préférerait avoir des **connexions clairsemées (sparse)**
 - Moins de poids
 - Régions rapprochées – reliées
 - Régions éloignées – non reliées
- **Comment pouvons-nous faire cela ?**
- Convolutions !



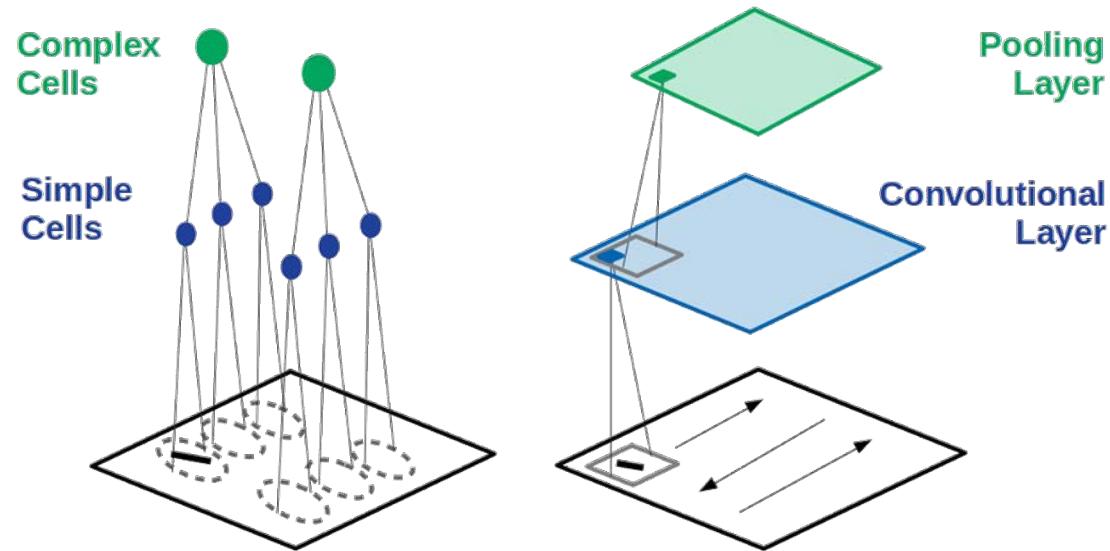
Réseau de neurones à convolution

- ***Convolutional Neural Network (CNN)***
- On utilise des convolutions plutôt que des connexions denses pour traiter les images
- Profite de la structure de nos données !
- **Hypothèse : les pixels proches sont reliés, les pixels éloignés le sont moins.**



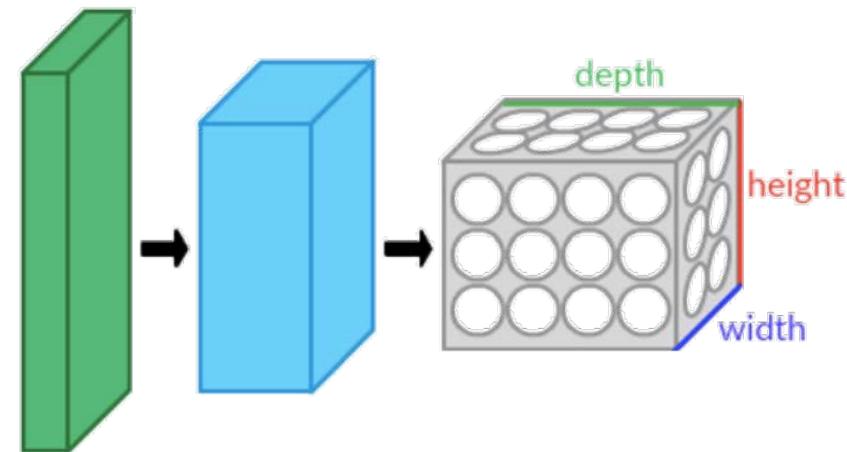
Analogie avec le système visuel humain

- « Hubel and Wiesel (1962) discovered that **simple cells** (left, blue) have preferred locations in the image (dashed ovals) wherein they respond most strongly to bars of particular orientation. **Complex cells** (green) receive input from many simple cells and thus have more spatially invariant responses. These operations are replicated in a CNN (right). »



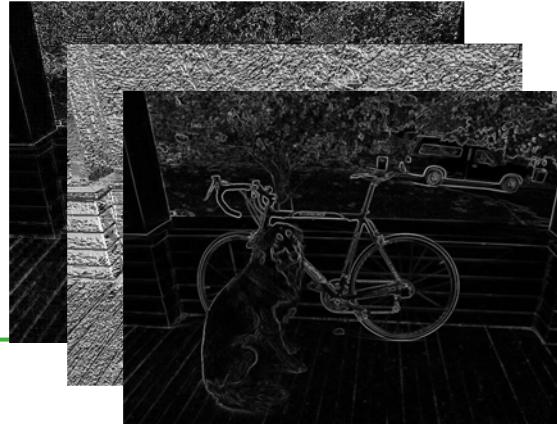
Couche de convolution

- **Entrée** : une image
- **Traitement** : convolution avec plusieurs filtres
- **Sortie** : une image,
 - **# canal = # filtres**
- La sortie est encore une somme pondérée des entrées, avec une activation



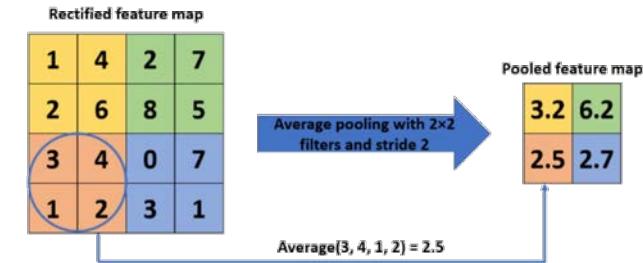
Les images sont énormes !

- Même les images de taille 256 x 256 ont plusieurs centaines de milliers de pixels. Et ces images sont considérées petites !
- **Solution : agrégation de l'information.**
- Nous n'avons pas besoin de toute l'image, elles peuvent être **sous-échantillonnées** sans éliminer les informations utiles



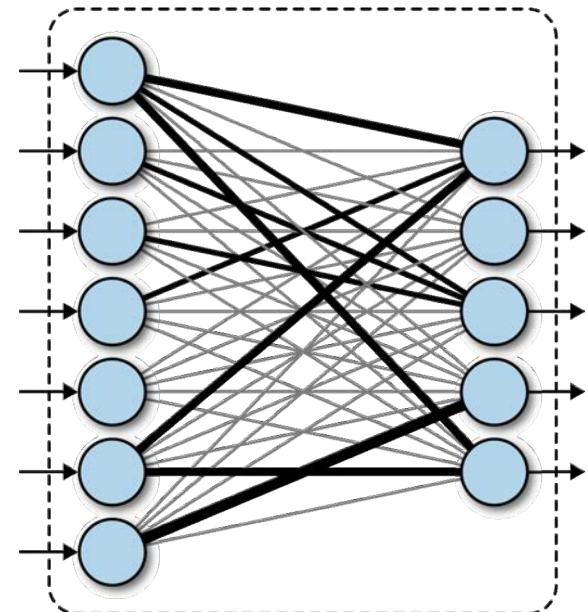
Couche de regroupement (*Pooling Layer*)

- **Entrée** : une image
- **Traitement** : regroupement (pool) des valeurs des pixels dans une région
- **Sortie** : une image dont la taille a été réduite d'un certain facteur (*stride*)
- **Hyper paramètres** :
 - **Quel type de regroupement ?** Moyenne, max, min ...
 - **Quelle taille de foulée ?** Contrôle le sous-échantillonnage
 - **Quelle taille de région ?** Généralement pas plus grand que la foulée.
- **Paramètres fréquents** : *max pooling* de 2x2 ou 3x3, foulée=2



Couche (entièrement) connectée

- **(Fully) Connected Layer**
- Une **couche de réseau de neurones standard**, où chaque neurone de l'entrée est connecté à chaque neurone de la sortie.
- Souvent utilisée pour convertir les cartes de **caractéristiques de l'image (image feature map)** vers la sortie finale, ou pour convertir la carte des caractéristiques de l'image en un seul vecteur.
- Élimine les informations spatiales



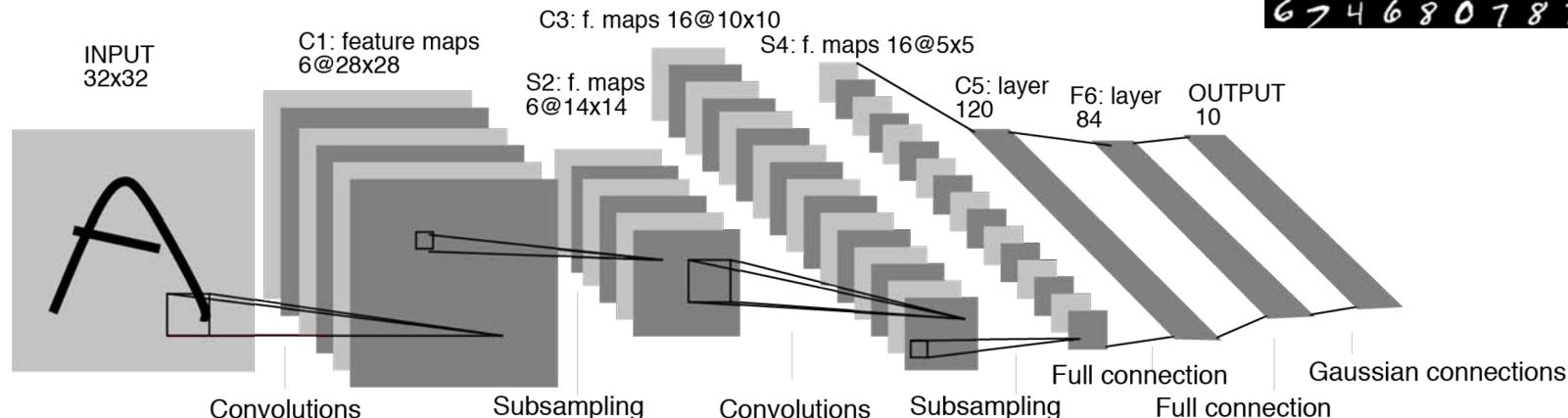
Composantes d'un Convnet

- **Couches de convolution** : extraction des caractéristiques
- **Couches de regroupement** : sous-échantillonner les cartes de caractéristiques, ce qui rend les traitements plus efficaces
- **Couches connectées** : Souvent utilisées pour la dernière couche, pour faire correspondre les cartes de caractéristiques de l'image avec les prédictions
- **D'autres types de couches spécialisées existent** (ex. : couches récurrentes, suréchantillonnage, normalisation, convolution dilatée, dropout, etc.)

LeNet : Premier Convnet pour les images

- Précision de 99% avec MNIST ([Yann LeCun, 1998](#))
- Possède tous les éléments d'un convnet moderne
- Mise à jour des poids par rétropropagation

5	0	4	1	9	2	1	3	1	4
3	5	3	6	1	7	2	8	6	9
4	0	9	1	1	2	4	3	2	7
3	8	6	9	0	5	6	0	7	6
1	8	1	9	3	9	8	5	9	3
3	0	7	4	9	8	0	9	4	1
4	4	6	0	4	5	6	1	0	0
1	7	1	6	3	0	2	1	1	7
8	0	2	6	7	8	3	9	0	4
6	7	4	6	8	0	7	8	3	1



ImageNet : Jeu de données d'images énorme

- Ensemble de données d'images et d'étiquettes 14 millions d'images, 22k catégories

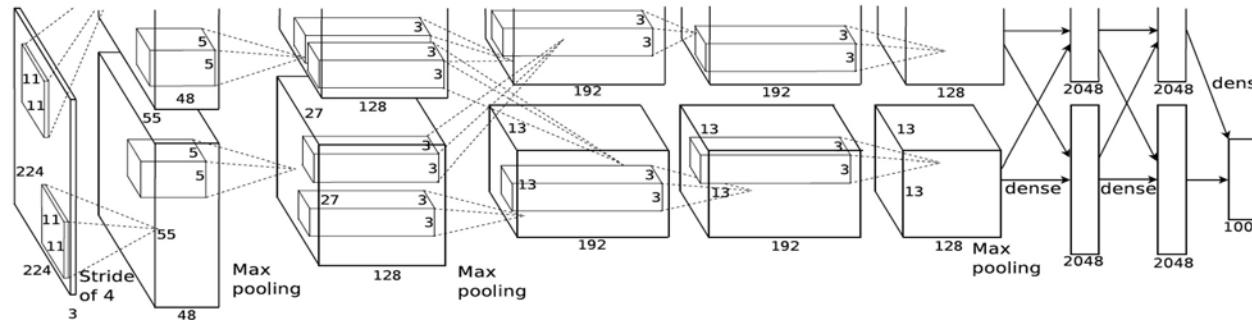


ImageNet: Crédit à la création du jeu de données



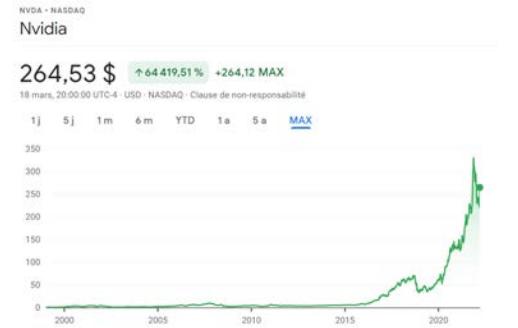
AlexNet : Premier bon réseau

- Participation au défi ImageNet
- Précision beaucoup plus élevée que les méthodes « traditionnelles » (ex. SVM sur SIFT)
- **Pourquoi y a-t-il autant de temps entre LeNet et AlexNet ?**



GPU : ce qui rend possibles les CNNs modernes

- Alex Krizhevsky a passé beaucoup de temps à implémenter les composants d'un CNN sur un GPU, son logiciel Cudaconvnet a été le **premier framework** de réseau neuronal « moderne »
- Les GPUs permettent une **accélération >100x** en comparaison avec un CPU, l'entraînement a tout de même nécessité plusieurs semaines pour ImageNet
- Cette idée, CNNs + GPUs, a initié une **révolution en vision** par ordinateur depuis 2016.



Qu'est-ce que ces réseaux apprennent ?

- Des caractéristiques ! Voici la première couche de AlexNet.



Idée : Trouver des images intéressantes

- Nous pouvons alimenter notre réseau en images et voir lesquelles activent certains neurones



(a) Unit sensitive to white flowers.



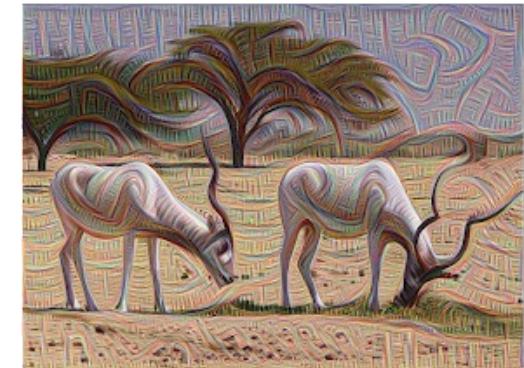
(c) Unit sensitive to round, spiky flowers.



(d) Unit sensitive to round green or yellow objects.

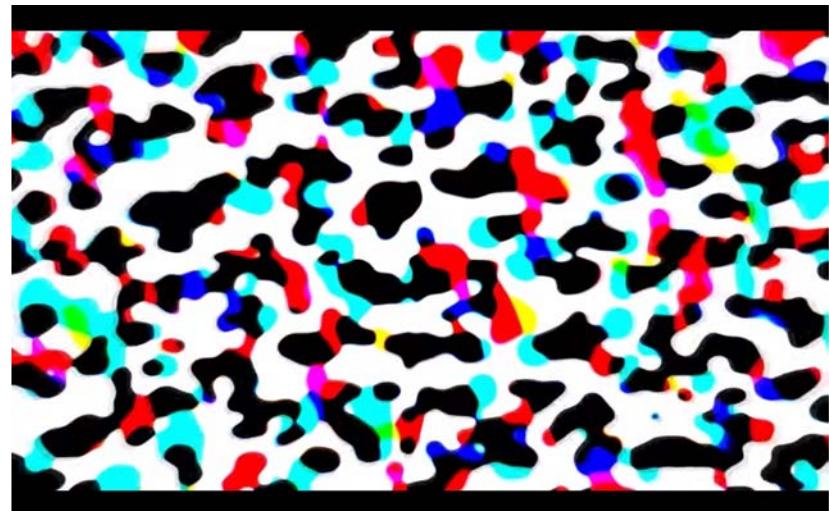
Idée : Créer des images intéressantes

- Au lieu d'optimiser le réseau, nous pouvons également effectuer une **descente de gradient sur l'image**
- Ceci **optimise l'image pour activer certains neurones** ou certaines couches, alors nous pouvons apprendre ce que font ces couches ou ces neurones !
- Vous avez peut-être vu cela sous le nom de **rêve profond (*DeepDream*)**



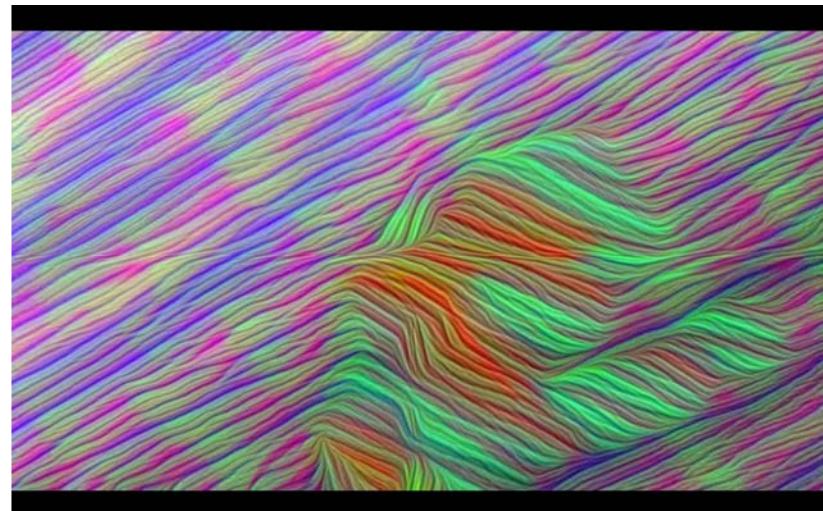
Premières couches : blobs

- Les neurones réagissent fortement à un contraste élevé



Couches intermédiaires : bords, courbes et yeux

- Les neurones répondent à des caractéristiques et des formes simples



Couches intermédiaires : bords, courbes et yeux

- Les neurones répondent à des caractéristiques et des formes simples



Couches intermédiaires : bords, courbes et yeux

- Les neurones répondent à des caractéristiques et des formes simples



Couches profondes : yeux, objets, chiens

- Les neurones répondent à des arrangements de caractéristiques



Couches profondes : yeux, objets, chiens

- Les neurones répondent à des arrangements de caractéristiques



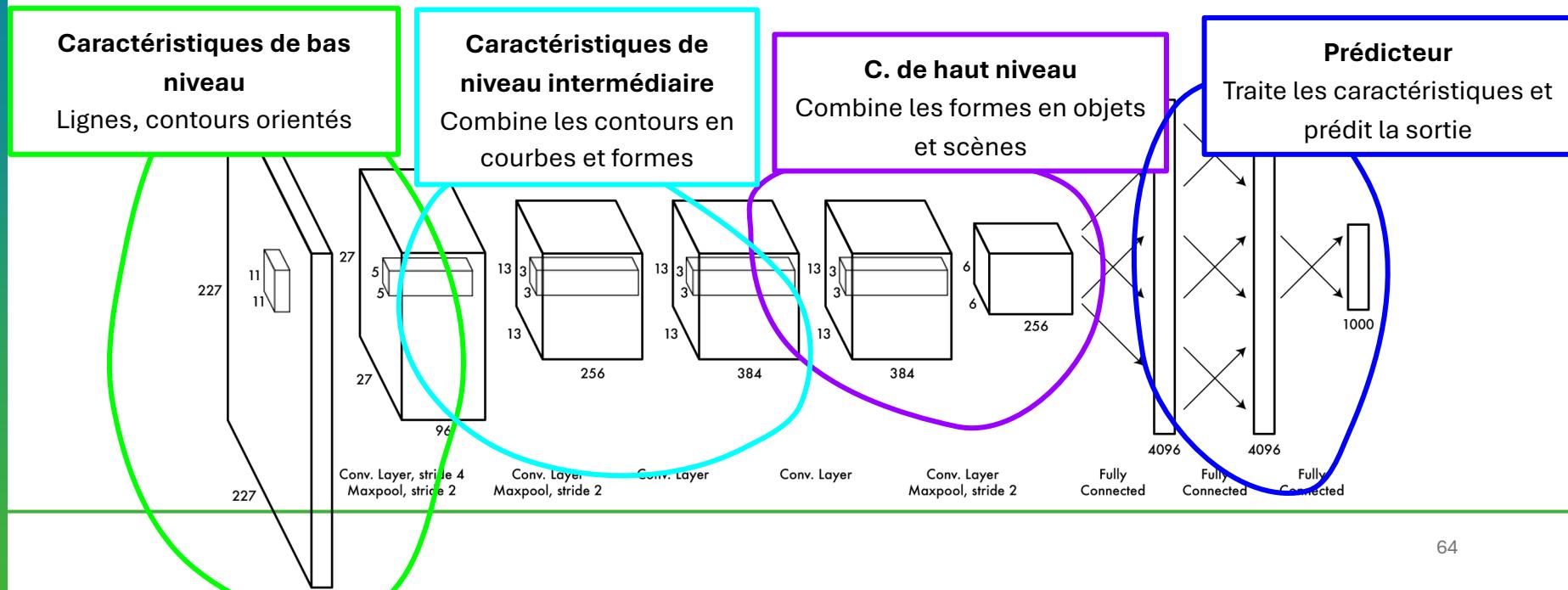
Couches profondes : yeux, objets, chiens

- Les neurones répondent à des arrangements de caractéristiques



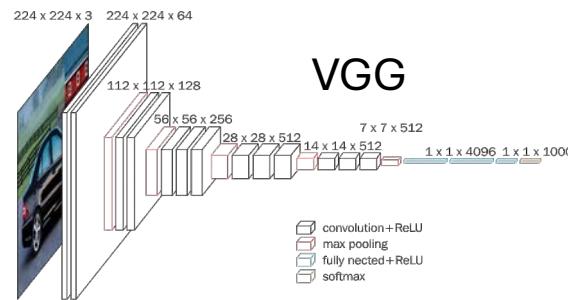
Les réseaux de neurones fonctionnent !

- Au moins, ils semblent faire ce que nous voulons qu'ils fassent

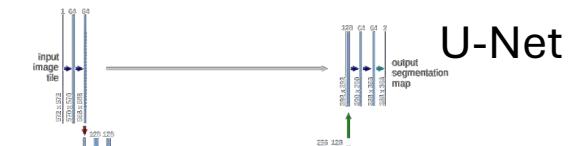
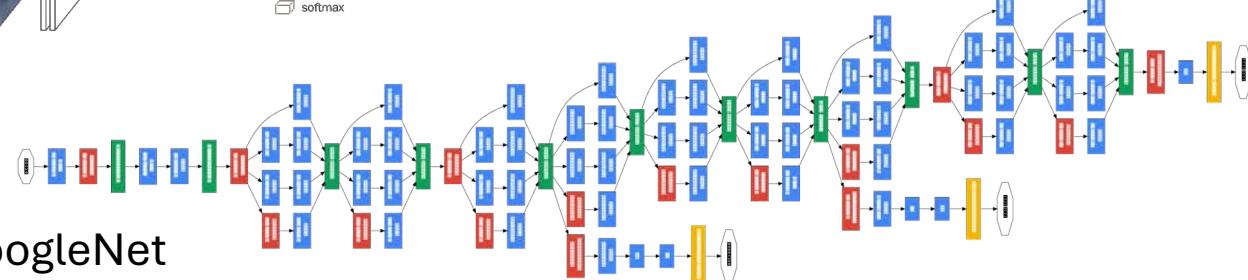


Plusieurs architectures de réseaux

- Des améliorations progressives pour améliorer les performances des réseaux et faciliter leurs entraînements
- VGG
- GoogleNet
- ResNet
- U-Net
- Transformeur

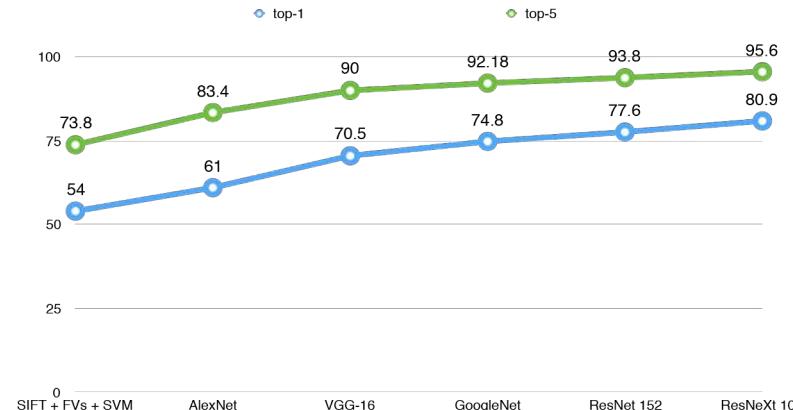


GoogleNet



Quel sera le prochain réseau performant ?

- Les réseaux commencent à **saturer ImageNet**, fournissant des améliorations de 1-2 %
- Maintenant que la vision fonctionne réellement bien, on peut aborder de nouvelles tâches
- Exemple d'applications
 - Segmentation
 - Détection d'objets
 - Sous-titrage



Tâches de vision par ordinateur

INF600F – Traitement d'images

Tâches de vision par ordinateur

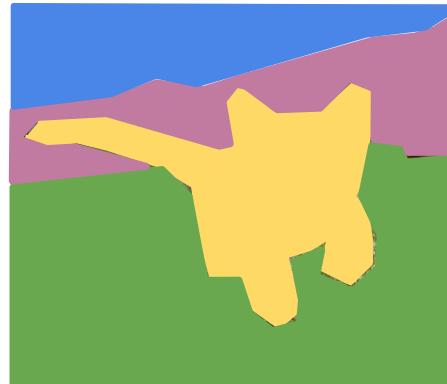
Classification



CHAT

Pas d'information
spatiale

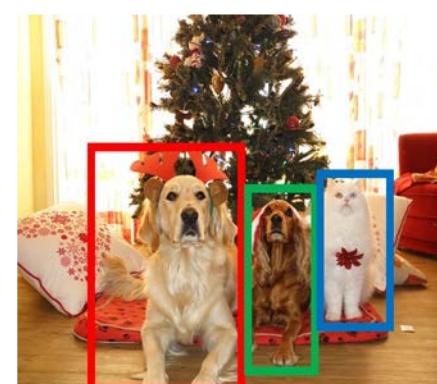
Segmentation sémantique



**HERBE, CHAT,
ARBRE, CIEL**

Pas d'objets,
seulement des pixels

Détection d'objet



**CHIEN, CHIEN,
CHAT**

Plusieurs objets

Segmentation d'instance

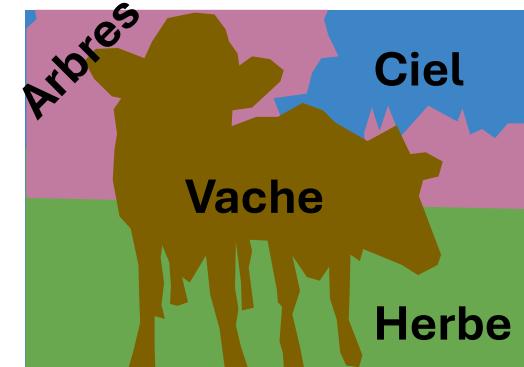
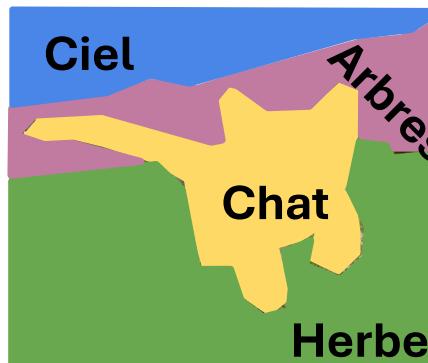


**CHIEN, CHIEN,
CHAT**

[CC0 domaine public](#)

Segmentation sémantique

- **Tâche** : Étiqueter chaque pixel dans l'image avec une catégorie
- Ne différencie pas entre les instances, ne s'intéresse qu'aux pixels
- **Exemple** : 1 seule région de pixels pour les 2 vaches

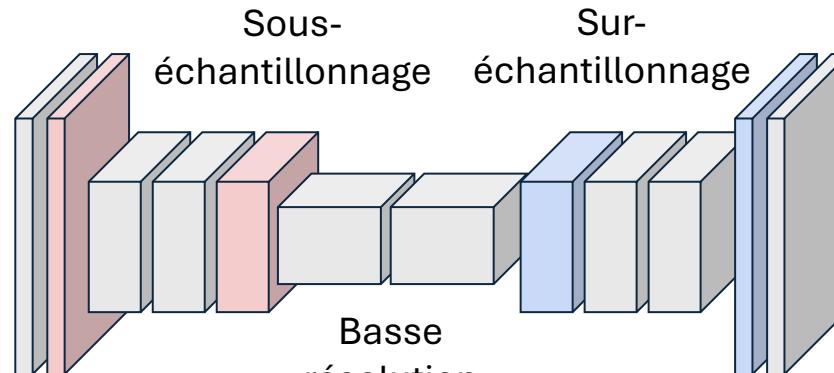


FCN & Goulot d'étranglement (*Bottleneck*)

- Concevoir un réseau en tant qu'une série de plusieurs couches de convolutions, avec du **souséchantillonage (downsampling)** et du **suréchantillonage (upsampling)** dans le réseau !



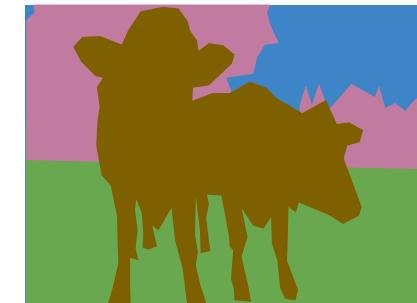
Entrée : $3 \times H \times W$



Haute résolution
 $D_1 \times H/2 \times W/2$

Sous-
échantillonage
Basse
résolution
 $D_3 \times H/4 \times W/4$

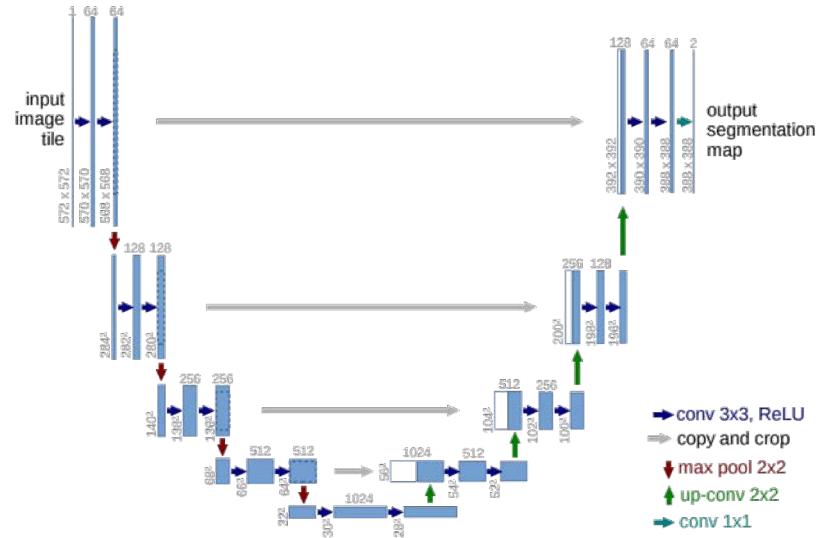
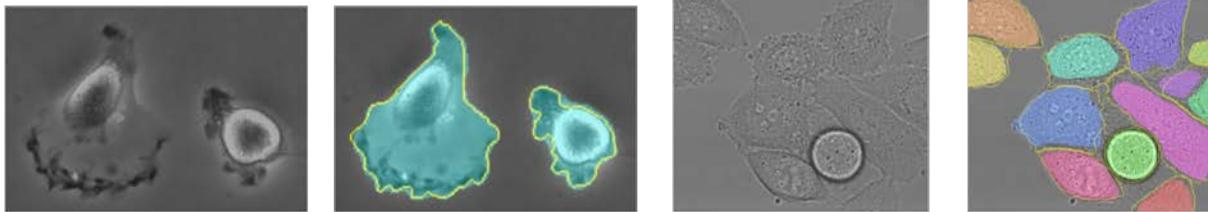
Sur-
échantillonage
Haute résolution
 $D_1 \times H/2 \times W/2$



Prédictions :
 $H \times W$

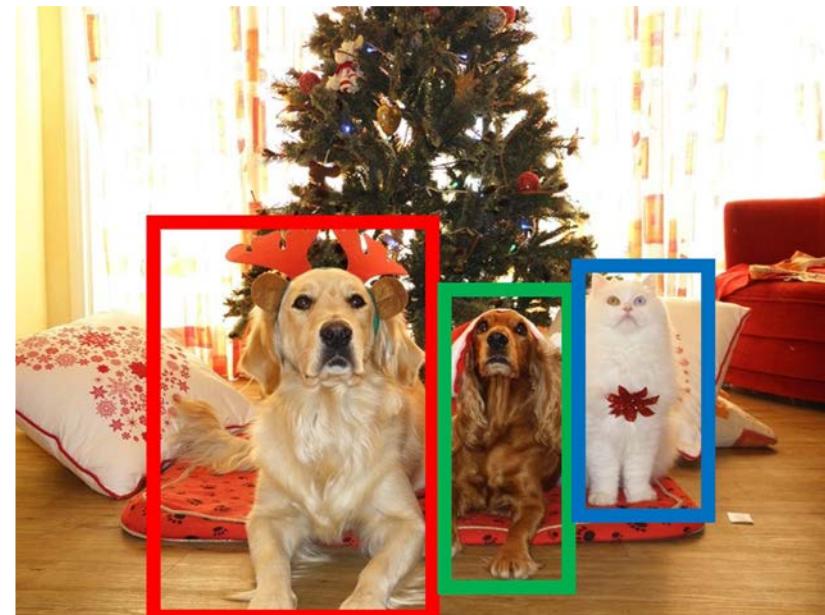
U-Net

- Connexions entre les niveaux d'encodage et de décodage (*skip connections*)
- Très répandu pour la segmentation, le débruitage, inpainting, flux optique, stéréo, ... a

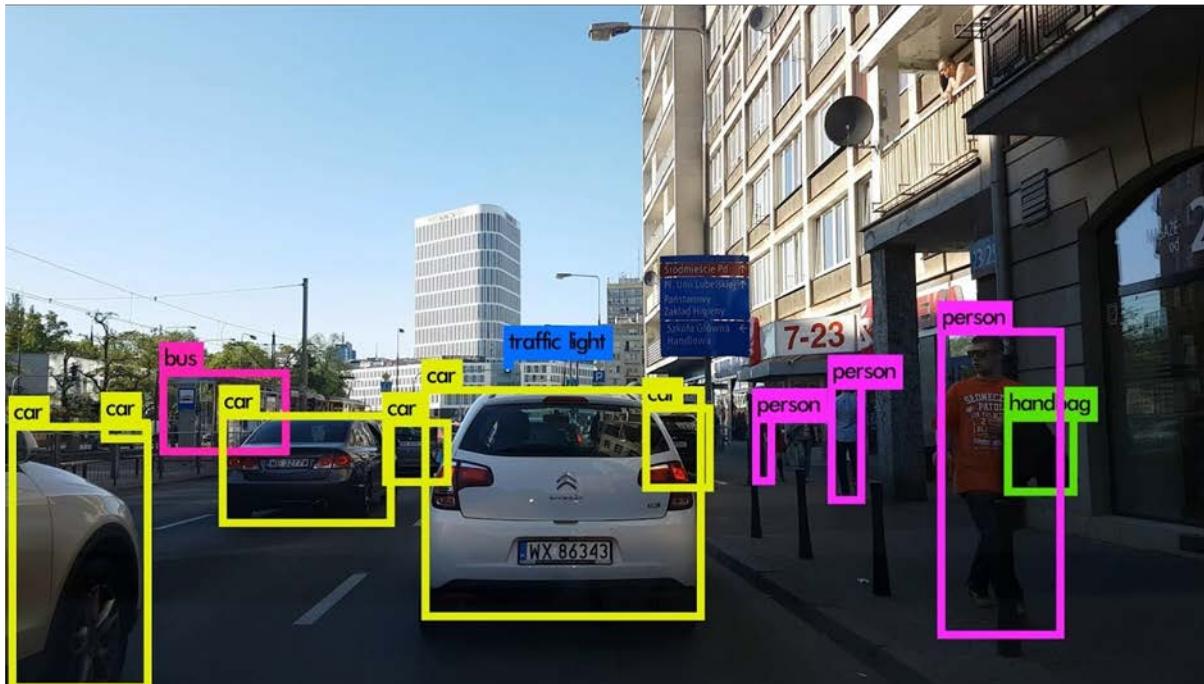


Détection d'objet : Définition de la tâche

- **Entrée** : une seule image RGB
- **Sortie** : un ensemble d'objets détectés. Pour chaque objet, prédire :
 1. L'étiquette de la **catégorie** (à partir d'un ensemble fixe de catégories connues)
 2. Boîte englobante (**Bounding box**, 4 nombres : x, y, largeur, hauteur)

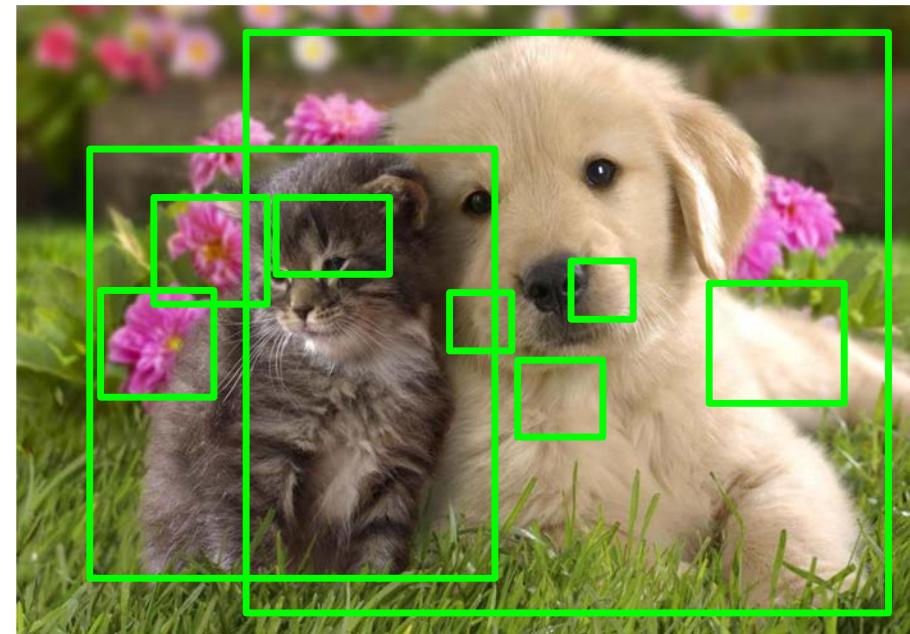


Détection d'objet

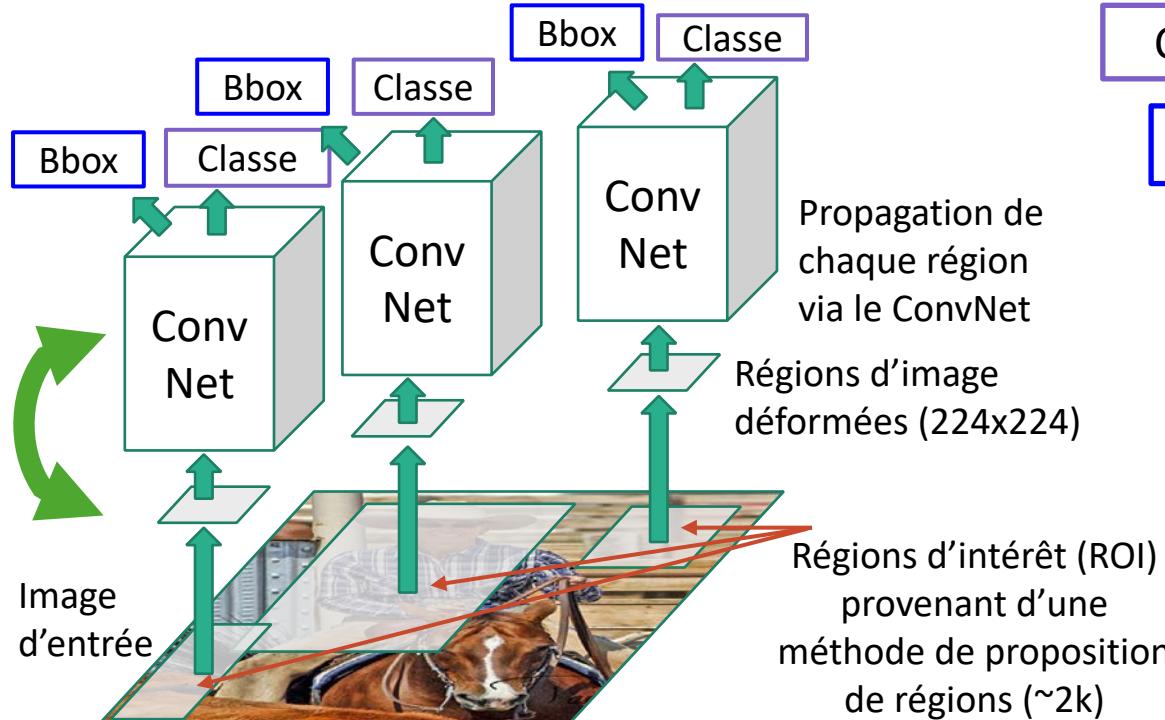


Propositions de région (*Region Proposals*)

- Trouver un **petit ensemble de boîtes** susceptibles de couvrir tous les objets
- Souvent basé sur des **heuristiques** : ex. rechercher des régions d'image **ressemblant à des «taches» (blobs)**
- Relativement **rapide** à exécuter. Par exemple, une recherche sélective donne 2000 propositions de régions en quelques secondes sur CPU



R-CNN: Region-Based CNN



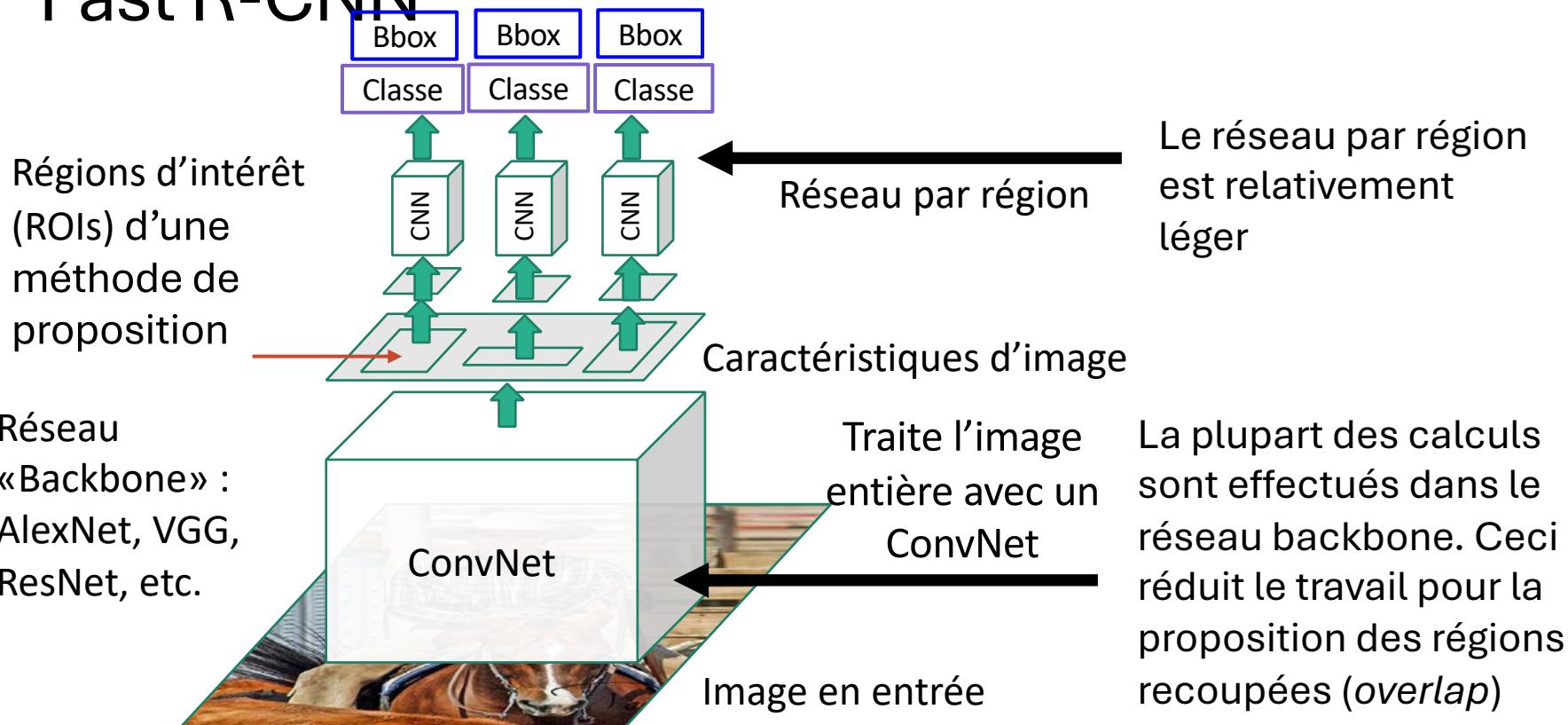
Classification de chaque région

Régression des *bounding box*

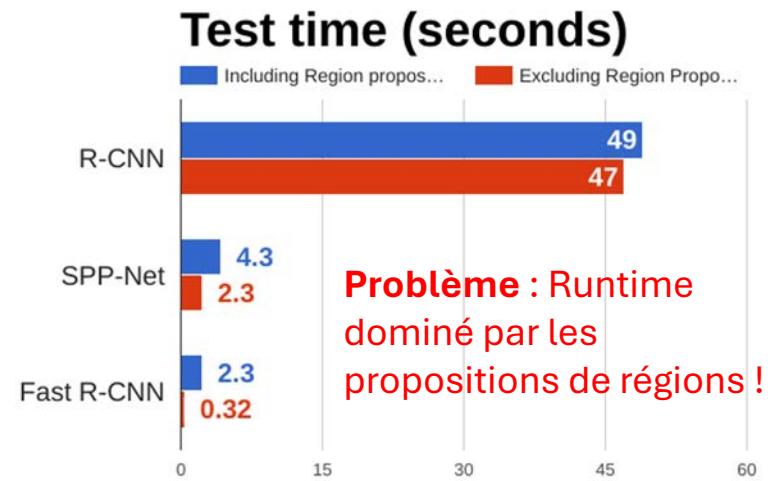
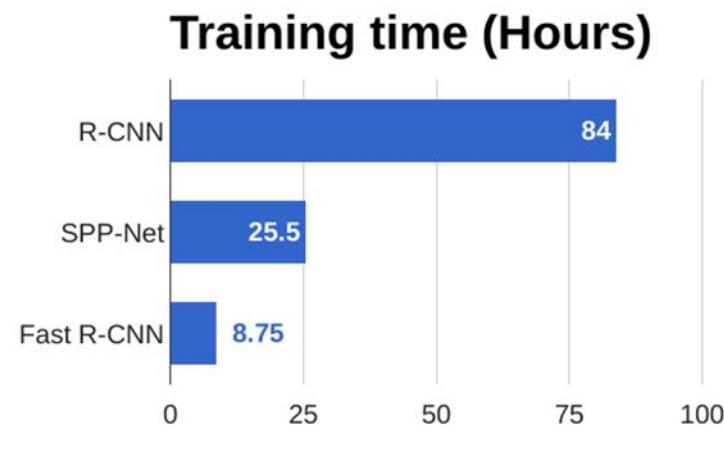
Problème : Très lent !
On doit effectuer $\approx 2k$ propagations avant pour chaque image !

Solution : Exécuter le CNN *avant* la déformation !

Fast R-CNN



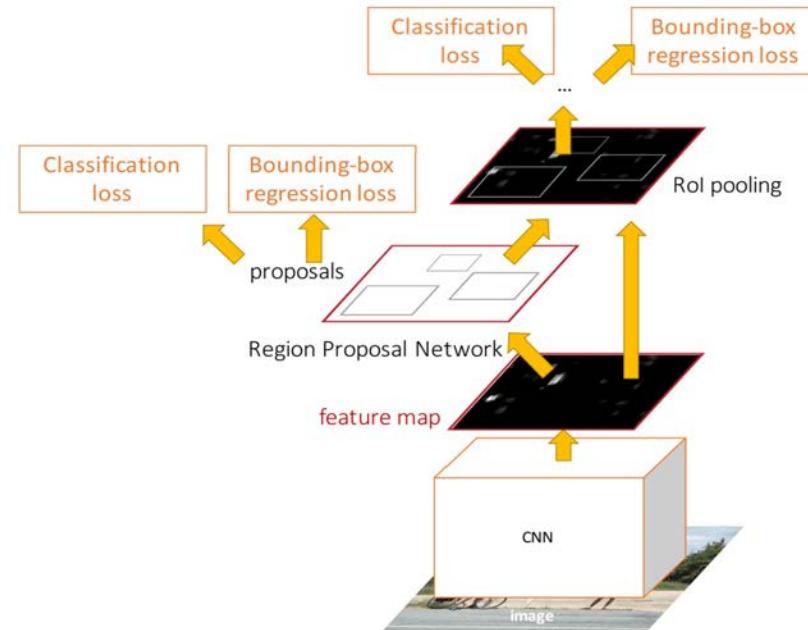
Fast R-CNN v.s. R-CNN «lent»



Propositions de région calculées par l'algorithme heuristique de "recherche sélective" sur le CPU -- apprenons-les plutôt avec un CNN !

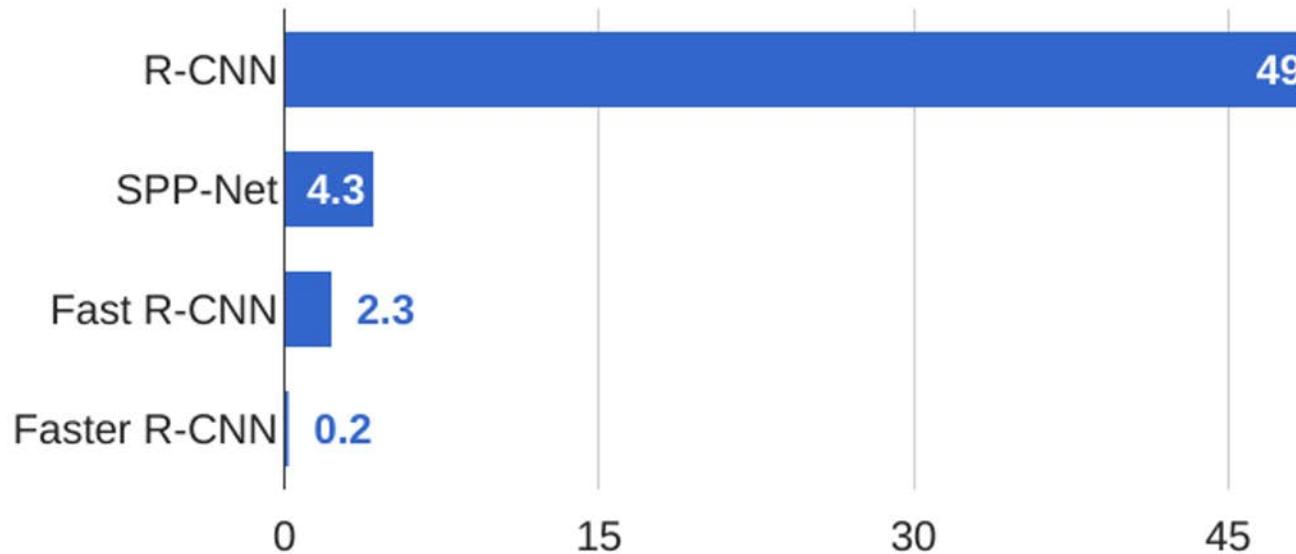
Faster R-CNN : Apprendre à proposer des régions

- Ajout d'un **réseau de proposition de région (Region Proposal Network, RPN)**
- Le reste est la même chose que Fast R-CNN : Rogner les caractéristiques pour chaque proposition, classifier chacune d'elle

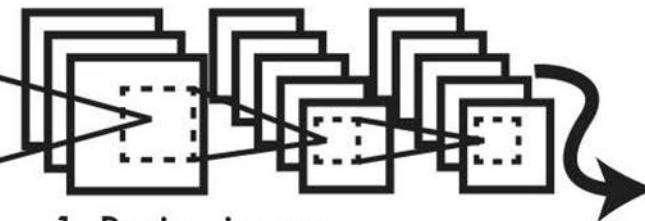


Faster R-CNN : Apprendre à proposer des régions

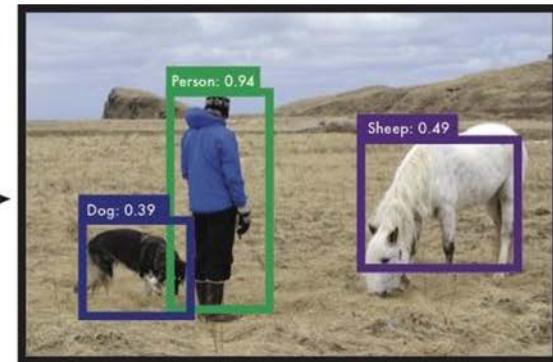
R-CNN Test-Time Speed



YOLO : You Look Only Once



1. Resize image.
2. Run convolutional network.
3. Threshold detections.

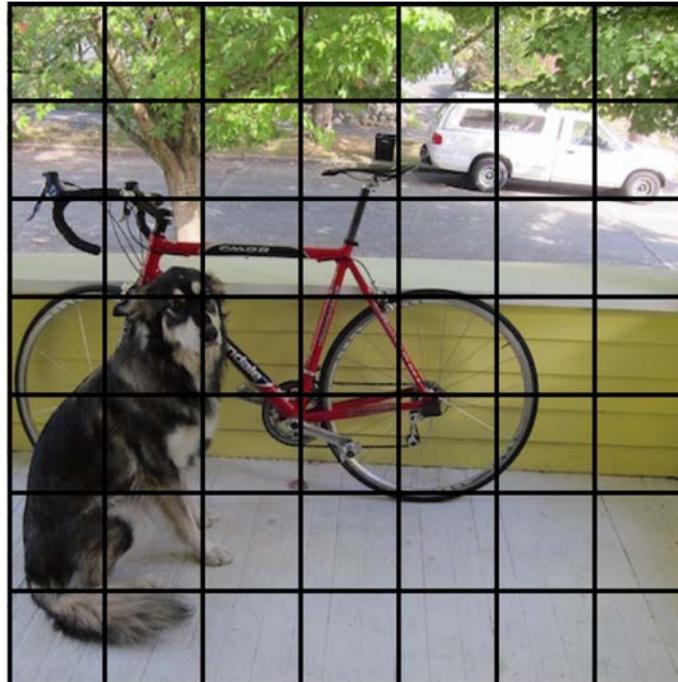


Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	155
YOLO	2007+2012	63.4	45
<hr/>			
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16 [28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

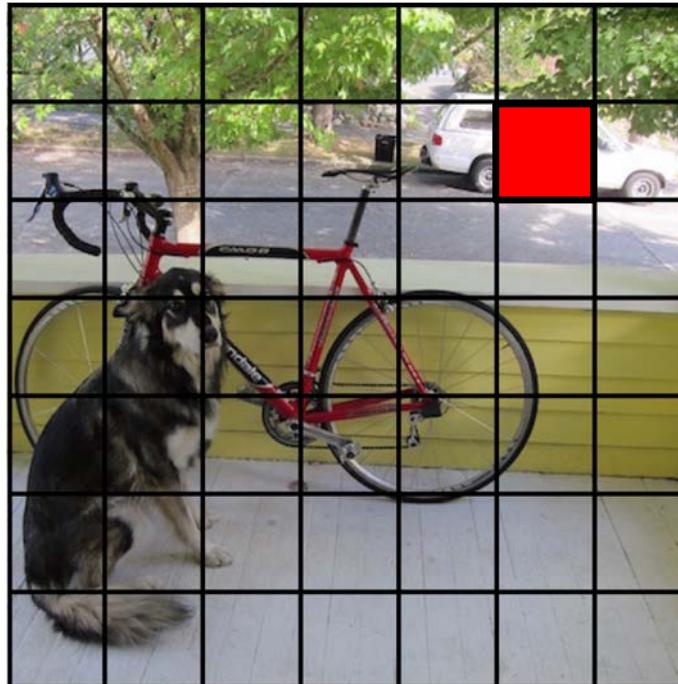
YOLO : Image initiale



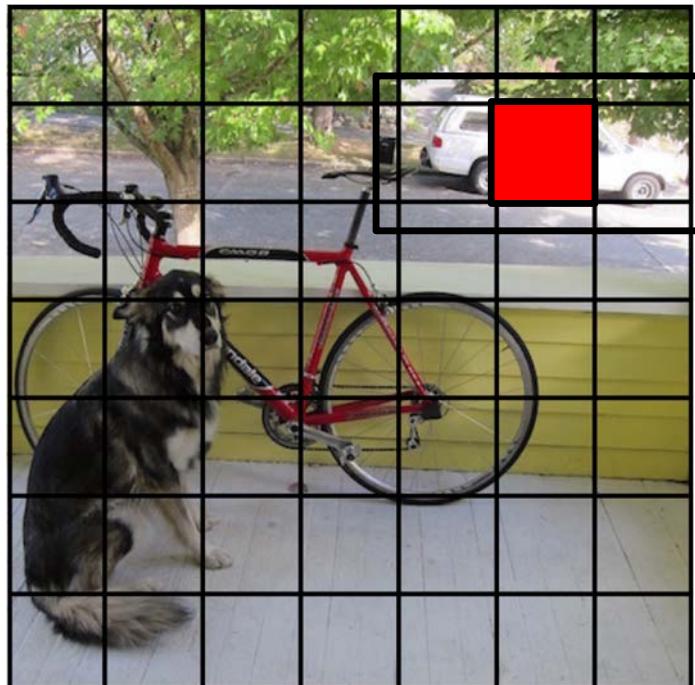
YOLO : Séparation en grille



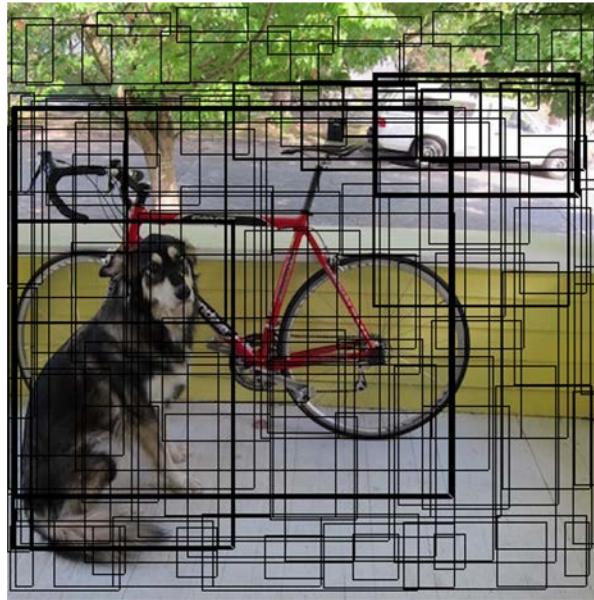
YOLO : Pour chaque cellule, prédire $P(\text{obj})$



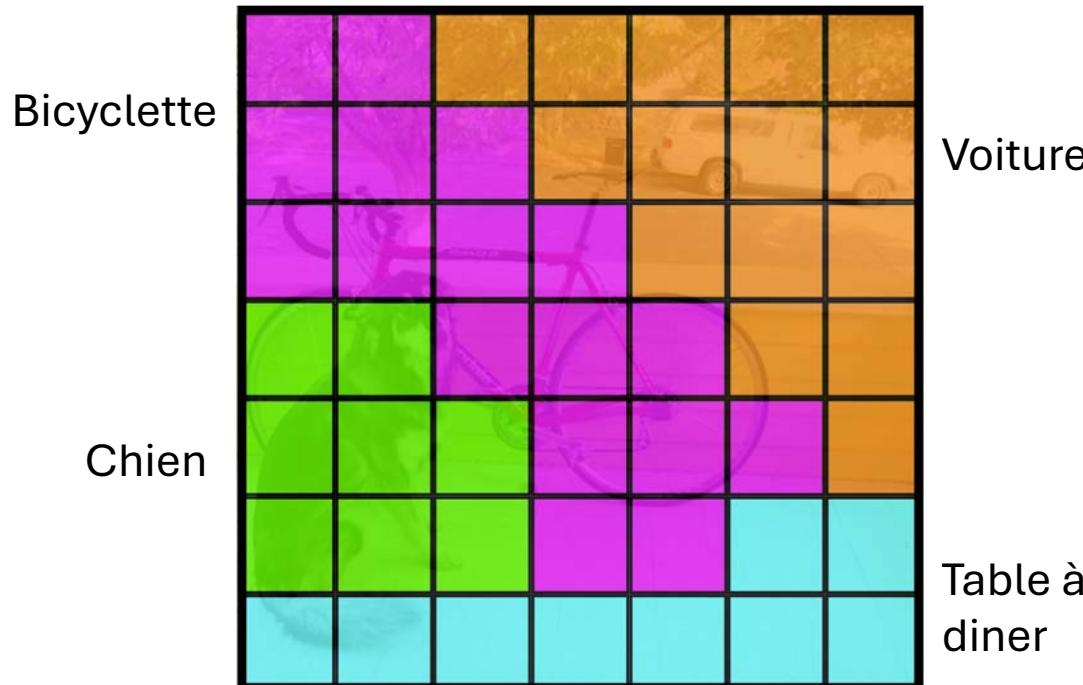
YOLO : Prédire aussi la boîte englobante



YOLO : Prédire aussi la boîte englobante



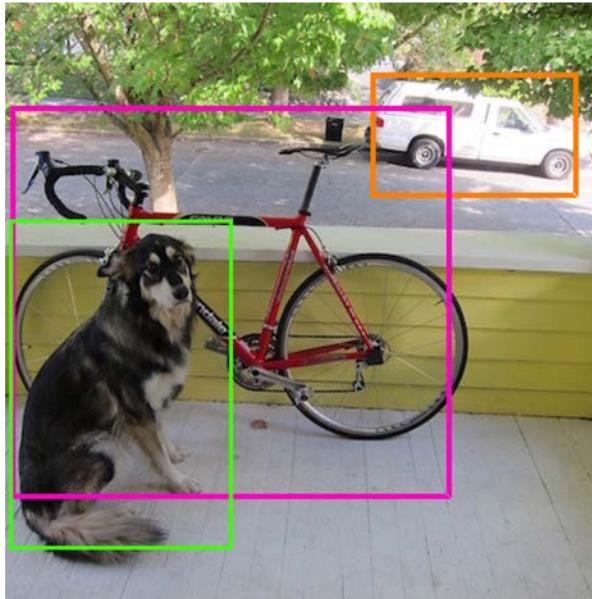
YOLO : Prédire aussi la probabilité de chaque classe



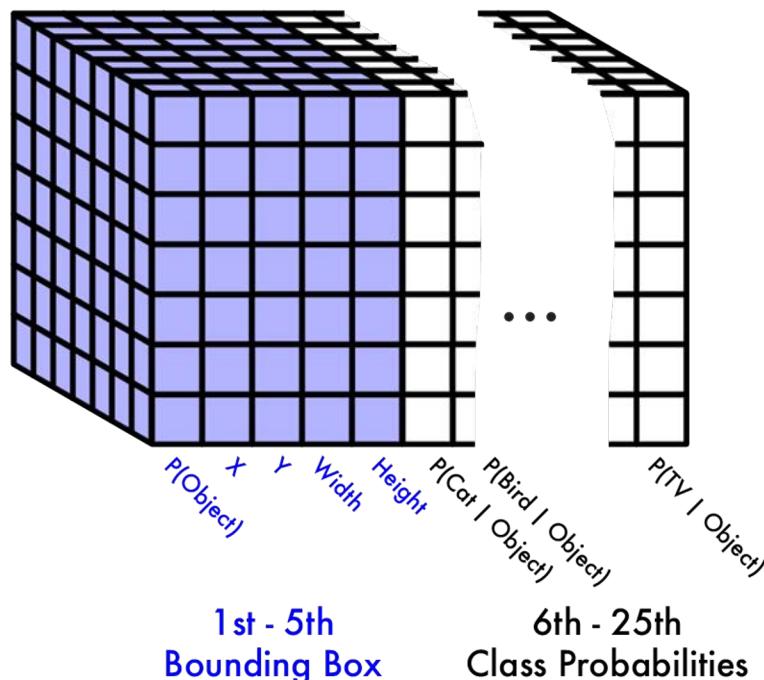
YOLO : Prédire la probabilité de chaque classe



YOLO : Seuillage et suppression non-max



YOLO : Détection d'encodage de tenseur

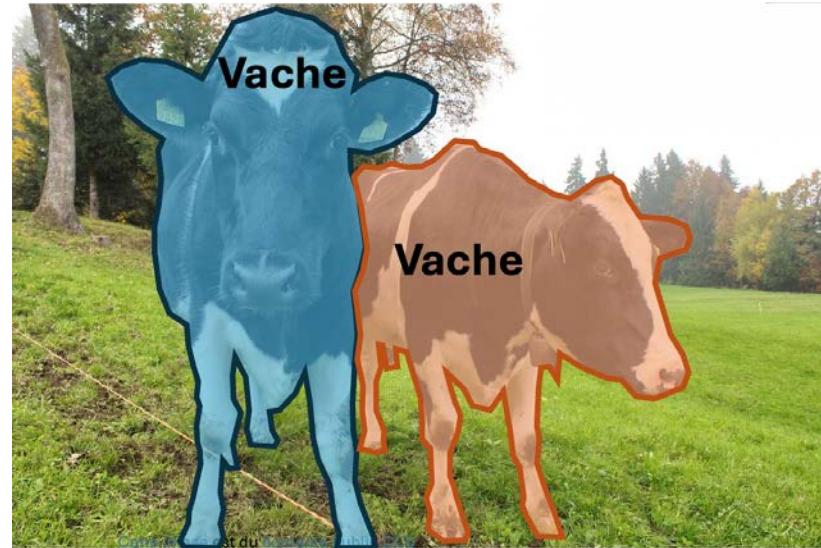


Détection d'objet : Code Open-Source

- La détection d'objet est un sujet difficile !
- N'implémentez pas cette tâche de zéro
- **Detectron2 (PyTorch):**
 - <https://github.com/facebookresearch/detectron2>
 - Fast / Faster / Mask R-CNN, RetinaNet
- **TensorFlow Detection API:**
 - https://github.com/tensorflow/models/tree/master/research/object_detection
 - Faster R-CNN, SSD, RFCN, Mask R-CNN
- **SAM : Segment Anything de Meta**
 - <https://segment-anything.com/>

Tâches de vision : Segmentation d'instance

- Déetecte tous les **objets** dans l'image, et identifie les **pixels associés** à chaque objet (seulement les choses)
- **Approche** : Effectuer une détection d'objet, puis prédire un masque de segmentation pour chaque objet

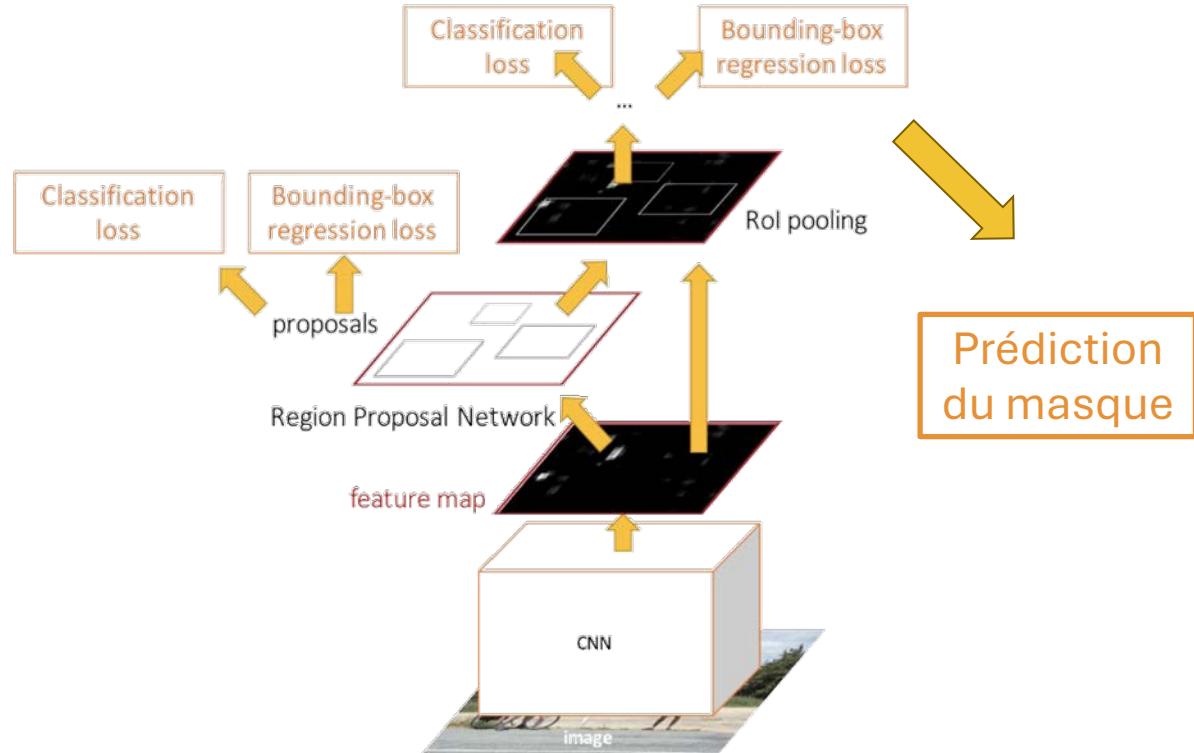


Segmentation d'instance : Mask R-CNN

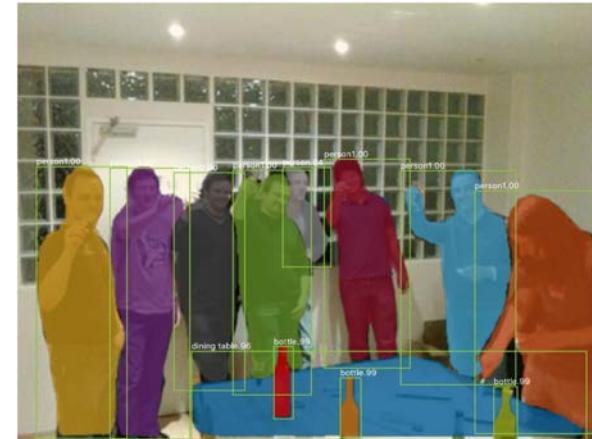
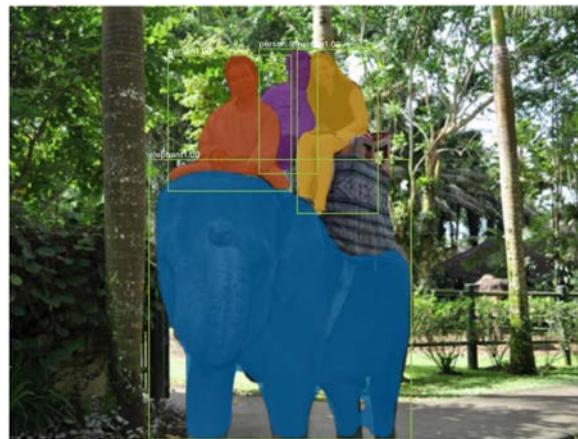
Segmentation d'instance



CHIEN, CHIEN,
CHAT



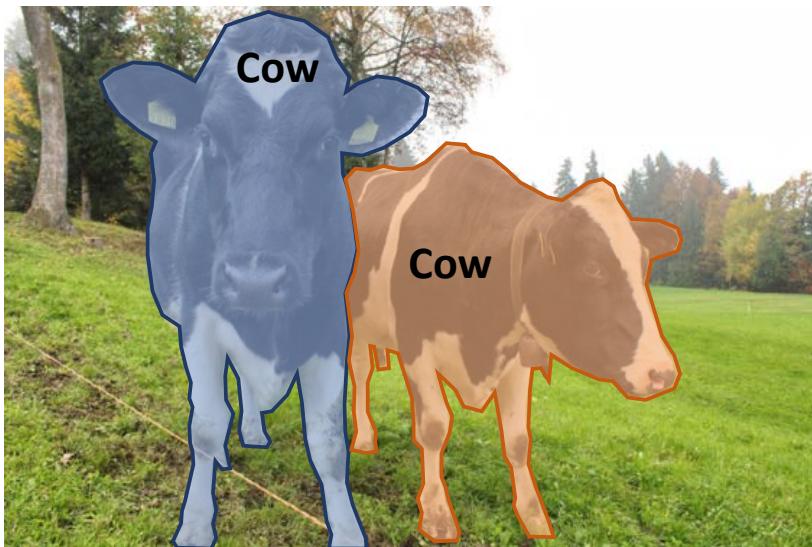
Mask R-CNN : de très bons résultats !



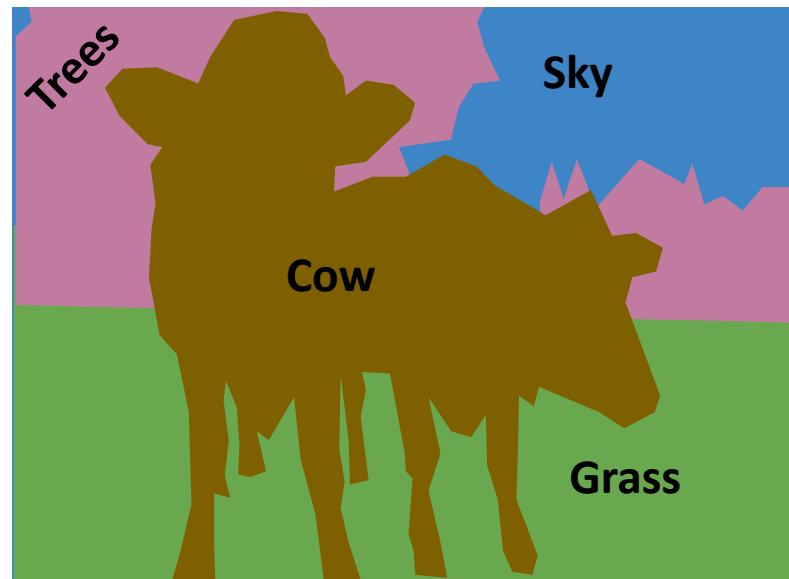
Démo de Mask R-CNN (Vidéo, 15min, <https://youtu.be/UWtac4cFERM>)

Au-delà de la segmentation d'instance

Segmentation d'instance : Instances séparées d'objet, mais seulement les *chooses*

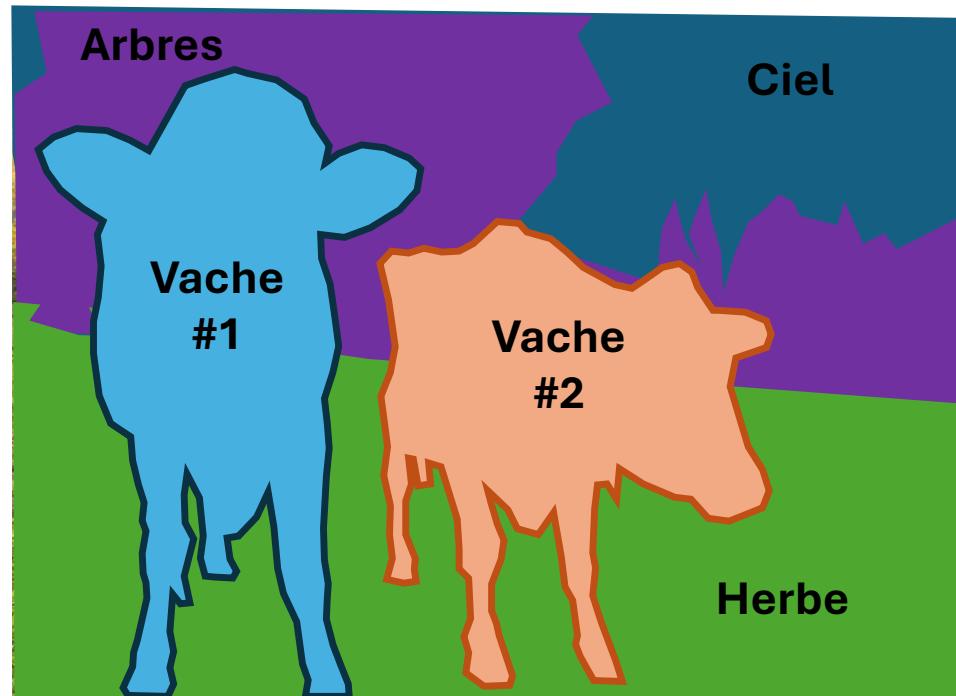


Segmentation sémantique : Identifie autant les *chooses* que les *trucs*, mais ne les sépare pas en instances

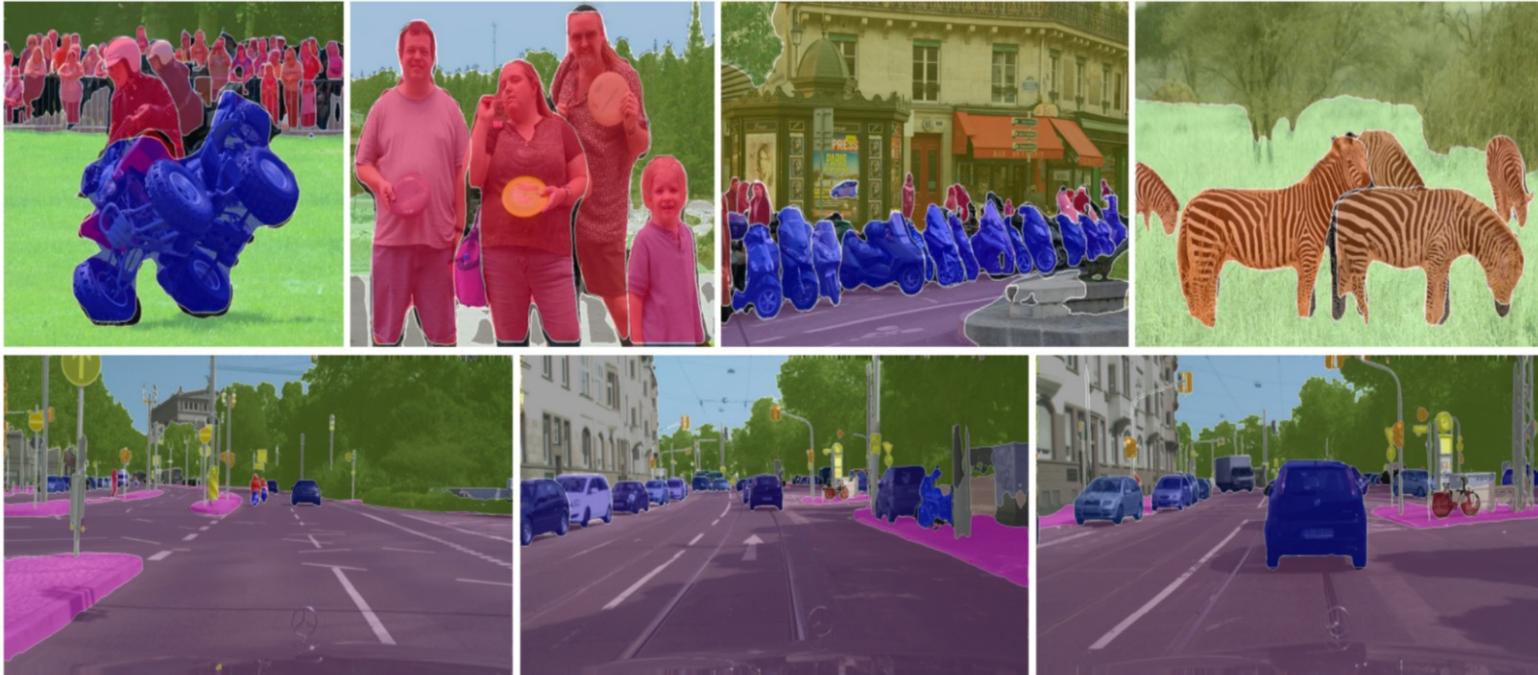


Au-delà de la segmentation d'instance : Segmentation panoptique

- Étiqueter tous les pixels dans l'image (autant les **choses** que les **trucs**)
- Pour les catégories de « choses », les **séparer en instances**



Segmentation panoptique



Examen final

INF600F – Traitement d'images

Examen final (mercredi 11 décembre)

- **2 salles** (séparé selon la première lettre de votre code permanent / nom)
 - **SH-2140 : A à K** (Salle habituelle du cours)
 - **SH-3220 : L à Z**
- **Pondération** : 35%
- **Heure** : 14h à 17h (durée de 3 heures)
- **Documentation** : 1 feuille de note (8 ½ x 11) recto verso (manuscrite)
- Aucune calculatrice / ordinateur / téléphone / etc.
- **Apportez votre carte étudiante / carte d'identité**
- **Les surveillants ne répondront à aucune question**, mais je visiterai les salles 2/3 fois au cours de l'examen
- En cas de doute, indiquez-le dans votre réponse.

Rappel du règlement 18

- Tout acte de **plagiat, fraude, copiage, tricherie ou falsification** de document commis par un.e étudiant.e, de même que toute participation à ces actes ou tentative de les commettre, à l'occasion d'un examen ou d'un travail faisant l'objet d'une évaluation ou dans toute autre circonstance, constituent une **infraction** au sens du règlement 18 (<http://r18.uqam.ca/>).
- Les règlements concernant le plagiat seront strictement appliqués.
- Les infractions seront déclarées **au comité d'infractions académiques** de la faculté qui a la responsabilité d'évaluer les actes et de décider des **sanctions**
- **Pour plus d'informations** : <http://www.infosphere.uqam.ca/rediger-untravail/eviter-plagiat>

Matière couverte

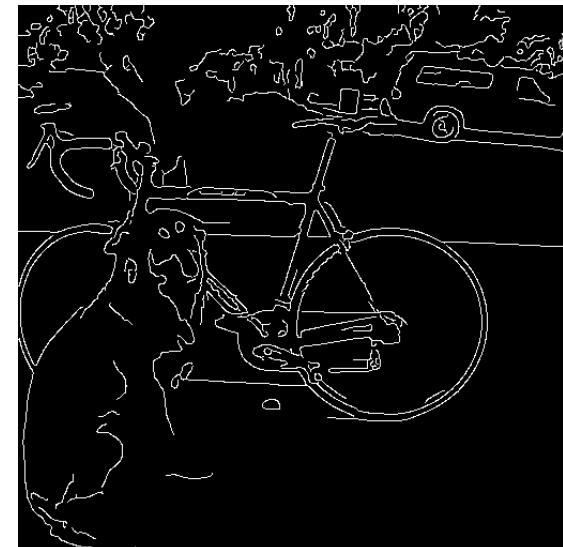
- Principalement les chapitres 5 à 7.
 - **Chapitre 5** : Segmentation
 - **Chapitre 6** : Morphologie mathématique
 - **Chapitre 7** : Extraction d'information
- **Une bonne compréhension de la matière des chapitres 1 à 4 est aussi nécessaire.**
- **Note** : révisez aussi les devoirs et ateliers pratiques, il pourrait y avoir des questions à ce sujet.
- Une **révision** de plusieurs concepts importants sera faite ce **vendredi à l'atelier pratique** avec Étienne.

Styles de questions

- Choix de réponses
- Vrai / Faux avec explication
- Explications des liens entre concepts en quelques phrases
- Association entre code et résultat, avec explication
- Calcul simple
- Question à développement (mise en situation)
- **Pas de programmation sur papier.** Si un algorithme ou du code est demandé, le pseudocode est accepté
- **Vaux mieux une réponse partielle qu'aucune réponse**

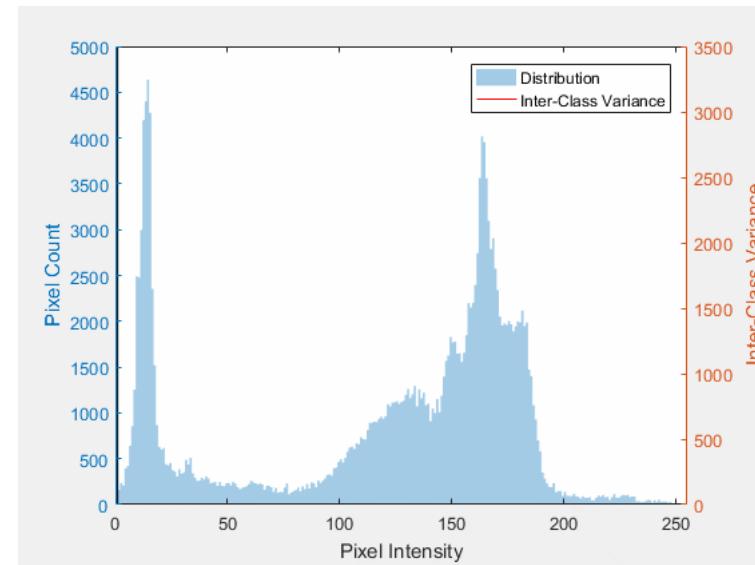
(Survol) Chapitre 5a : Segmentation

- Définition de la segmentation
- Régions vs Contours
- Détection de contours
 - Méthode du gradient
 - Filtres différentiels
 - Opérateur de Sobel
 - Laplacien
- Méthode de Canny
- Transformée de Hough



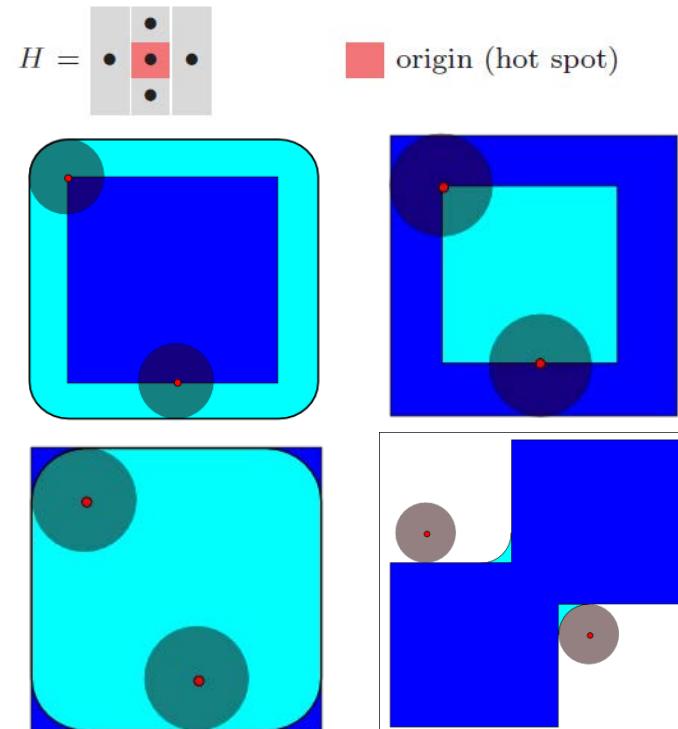
(Survol) Chapitre 5b : Segmentation

- Segmentation de régions
- Approches basées sur l'histogramme
 - Méthode du triangle
 - Statistiques de l'histogramme
 - Algorithme Isodata
 - Algorithme Otsu
- Seuillage global vs adaptatif
- Méthodes avancées
 - Croissance de région
 - Split and Merge
 - Clustering (ex. K-means)
 - Watershed



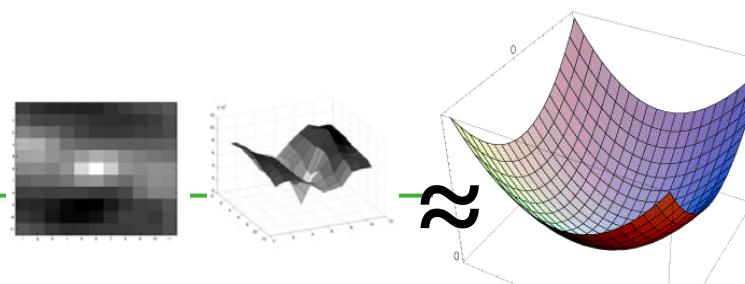
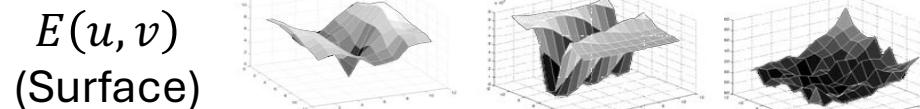
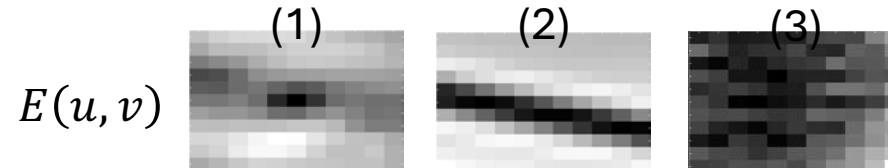
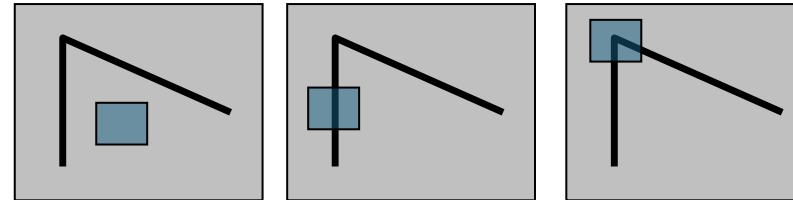
(Survol) Chapitre 6 : Morphologie

- Opérations sur les images binaires
- Filtre morphologique
- Élément structurant
- Érosion / Dilatation
- Algorithmes basés sur la morphologie mathématique
 - Ouverture / Fermeture
 - Hit or Miss
 - Remplissage de trous
 - Etc.
- Morphologie en niveaux de gris
 - Adaptation de la dilatation / érosion
 - Filtres Top-Hat / Bottom-Hat



(Survol) Chapitre 7a : Extraction d'information

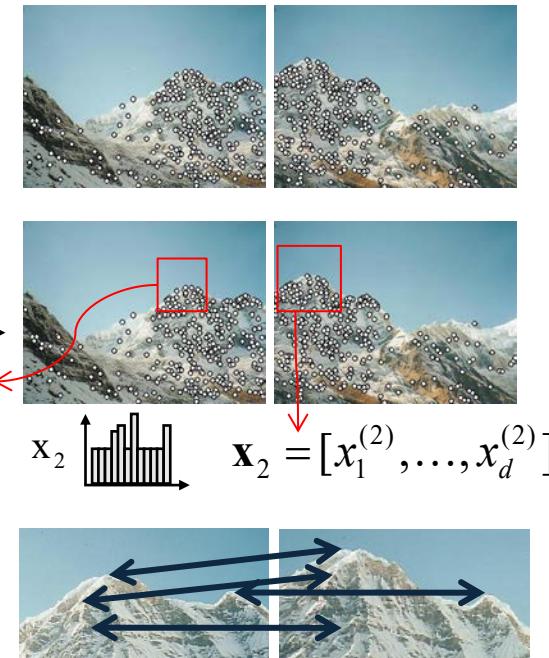
- Applications
- Points caractéristiques
- Mesure de l'unicité
- Détection de coins
- Tenseur de structure
- Méthode de Harris
- Matrice Hessienne



(Survol) Chapitre 7b : Extraction d'information

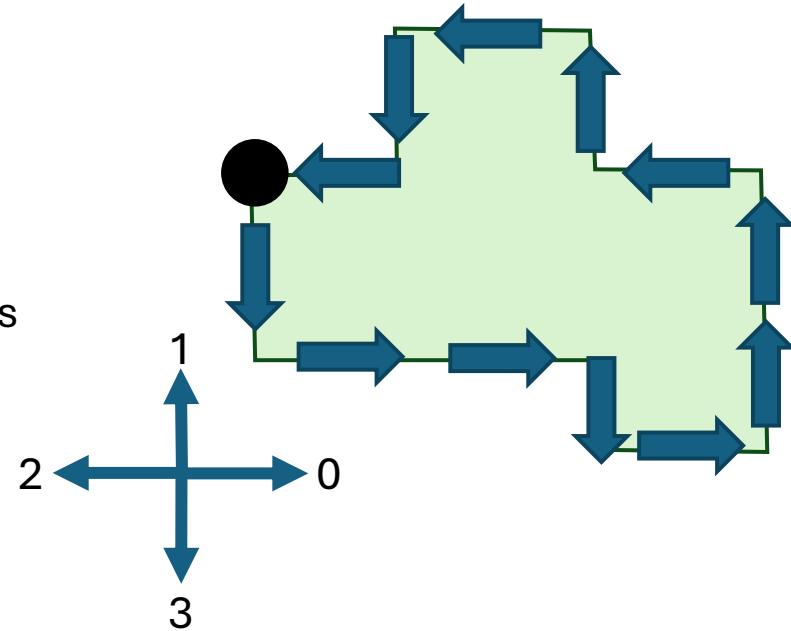
- Invariance et Covariance
- Détection automatique de l'échelle
- Détection de taches (blobs)
- Laplacien de la gaussienne (LoG)
- Différence de gaussiennes (DoG) $\mathbf{x}_1 = [x_1^{(1)}, \dots, x_d^{(1)}]$
- Descripteur de point caractésitiques
- Comparaison de descripteurs (ex. SSD, NCC)
- Descripteur MOPS
- SIFT

$$d(x_1, x_2) < T$$



(Survol) Chapitre 7c : Extraction d'information

- Détection des régions et contours
 - Étiquetage (3 méthodes)
 - Contours des régions
 - Suivi de frontières
 - Détection combinée des régions et contours
- Représentation des régions et contours
 - Codage par plage (*Run length encoding*)
 - Code à enchaînement
 - *Nombre de forme*
 - *Descripteur de Fourier*
- *Caractérisation des régions*
- *Caractérisation de la texture*



Code absolu : 3003011211232

Code différentiel : 10311013113

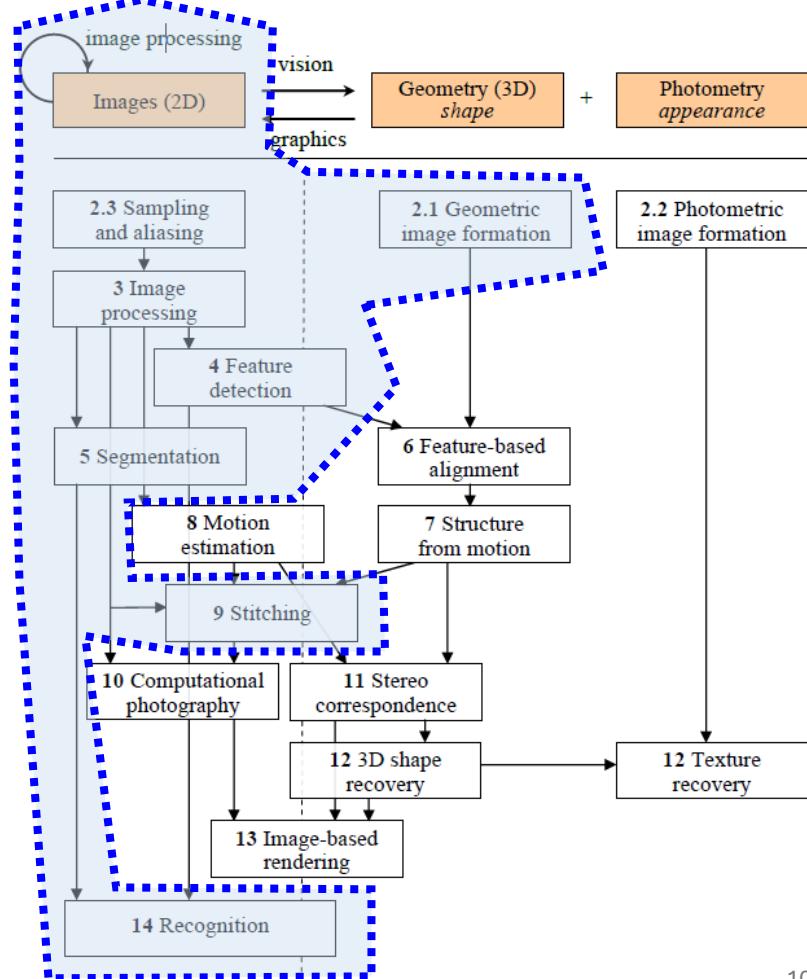
Nombre de forme : 013113130311

Suggestions pour l'étude

- Révisez les diapositives & références
- Révisez les devoirs
- Faites des liens entre les concepts vus en cours magistral et les devoirs
- Identifiez les concepts importants pour chaque chapitre, expliquez oralement en 1 ou 2 phrases
- Assurez-vous de maîtriser les bases de l'algèbre linéaire (ex. : vecteur, matrice, produit scalaire, produit vectoriel, multiplication matricielle) et du calcul différentiel et intégral
- **Bonne étude !**

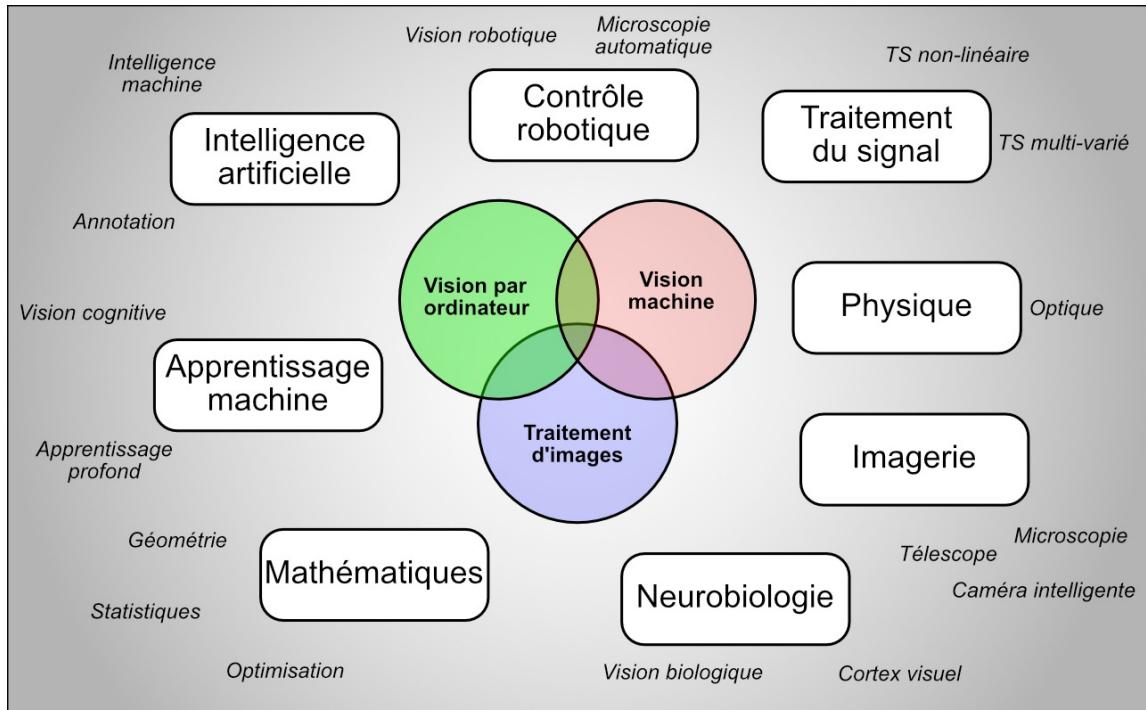
Retour sur le cours

- Formation d'images
- Manipulation d'images
- Filtrage spatial
- Filtrage fréquentiel
- Restauration d'images
- Segmentation
 - Régions, contours
- Morphologique mathématique
- Extraction de caractéristiques



Pour aller plus loin

- <https://www.learnopencv.com/>
- <https://www.pyimagesearch.com/>
- <https://www.fast.ai/>



Pour aller plus loin : Cours / Recherche

- Cours en imagerie numérique / IA
 - **INF5071 : Infographie** (Hiver 2025)
 - **INF889G : Vision par ordinateur** (Hiver 2025 (?), puis automne 2026)
 - **INF600F : Traitement d'images** (A2025, mercredi ↔ vendredi)
 - **DIC938X : Neuroinformatique** (H2026, nouveau cours)
- Contactez-moi si vous êtes intéressés à faire un stage d'été, une maîtrise ou un doctorat en lien avec le **traitement d'images**, la **vision par ordinateur**, **l'apprentissage profond** et la **microscopie / l'imagerie numérique**, ou un autre sujet.

Merci pour cette
belle session !

INF600F – Traitement
d'images

