

Réseaux de neurones convolutifs en vision par ordinateur

A. Carlier

2023

Plan du cours

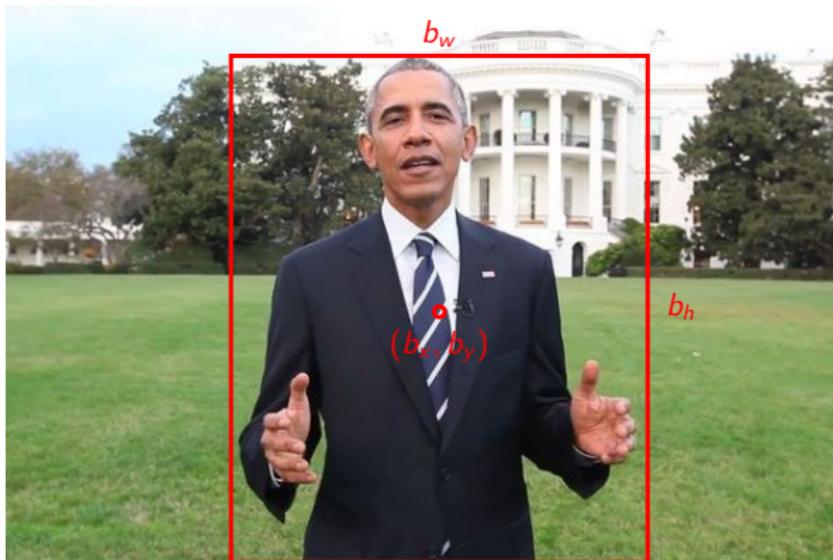
1 Localisation

2 Détection d'objet

3 Segmentation d'image

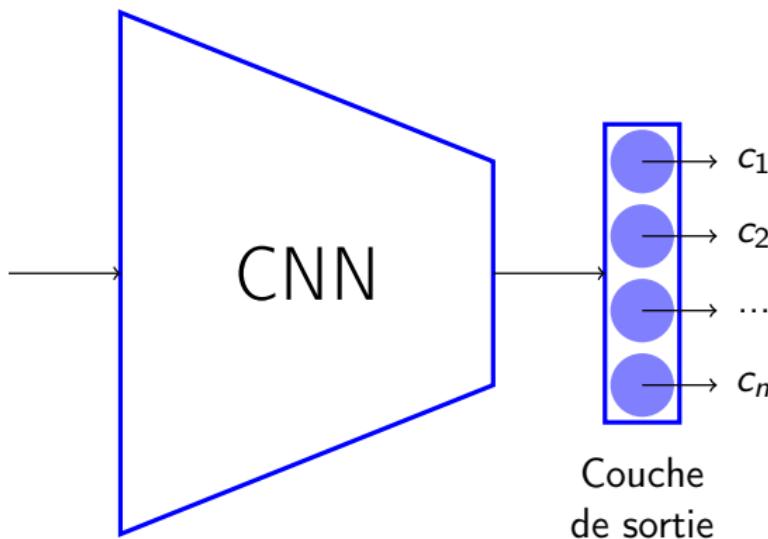
4 Estimation de pose

Classification et localisation conjointes



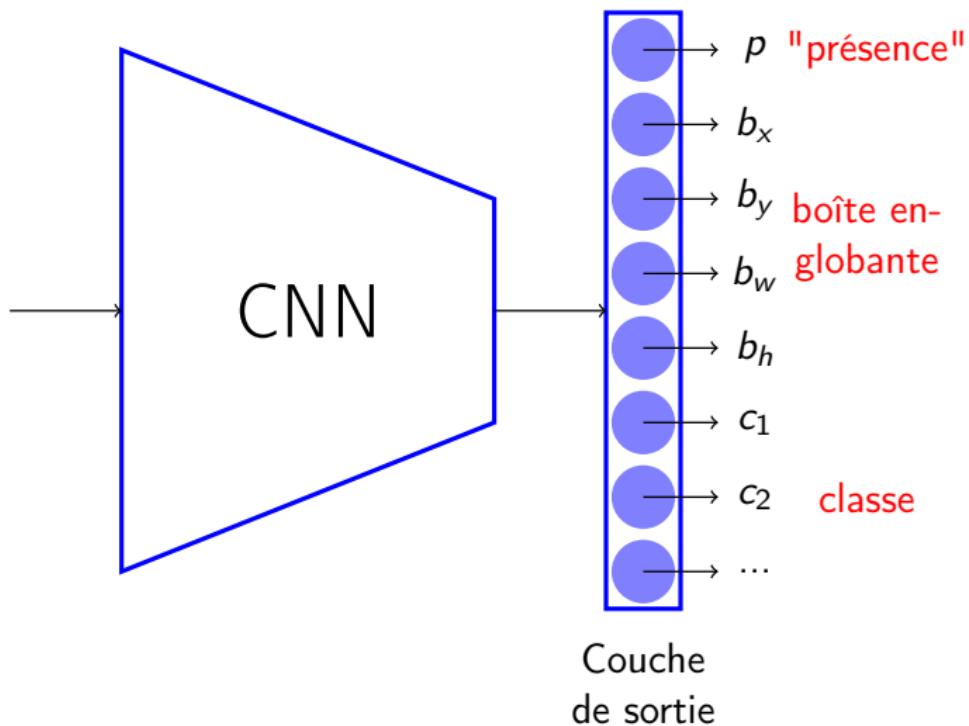
On cherche dans ce problème à classifier et localiser **un objet** par image.

Classification et localisation conjointes



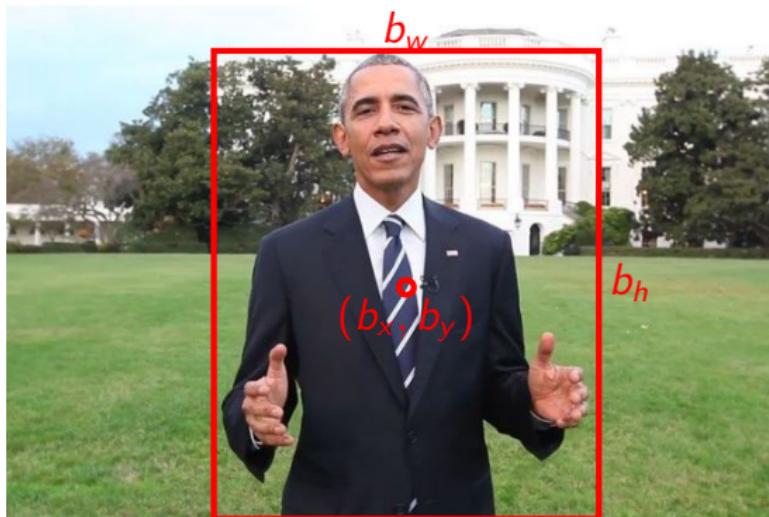
Nous devons ajouter une information spatiale sur l'objet classifié.

Classification et localisation conjointes



Nous pouvons rajouter les coordonnées de la boîte englobante autour de l'objet.

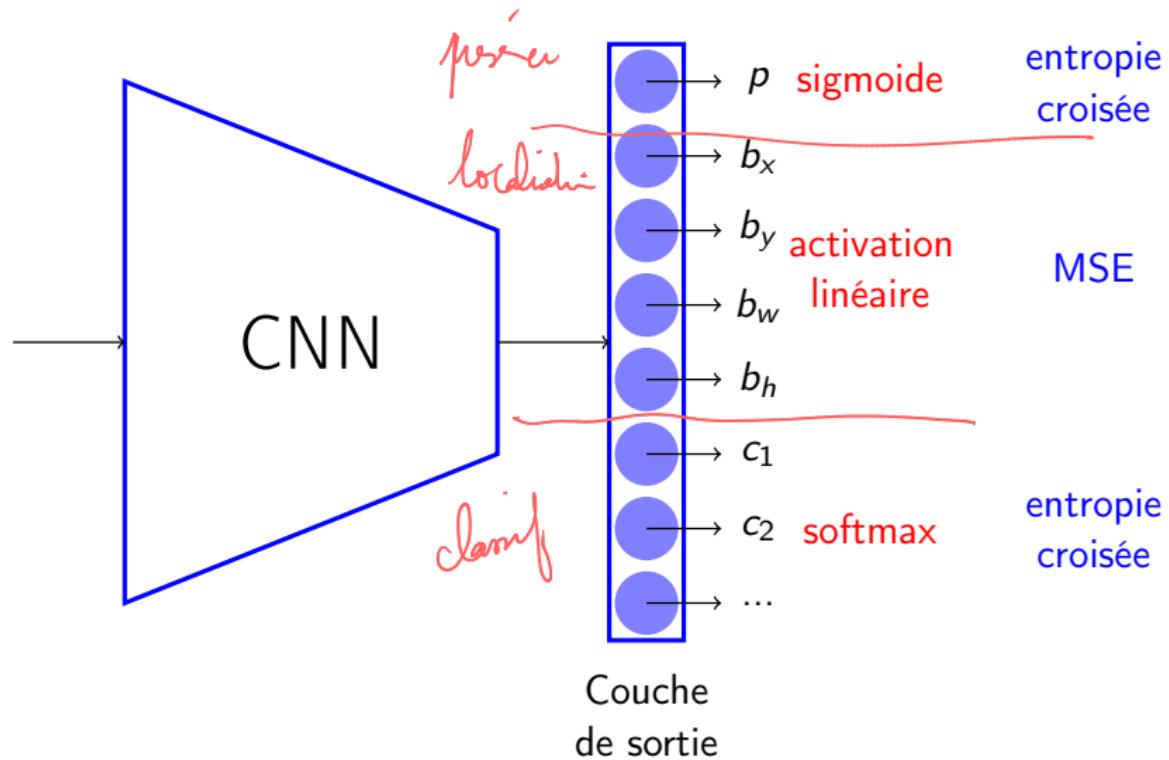
Classification et localisation conjointes



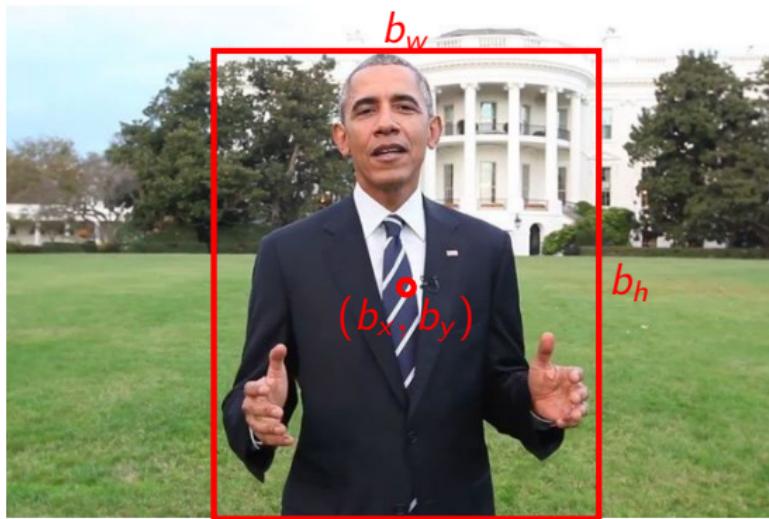
$$y = \begin{pmatrix} 1 \\ 0.52 \\ 0.46 \\ 0.5 \\ 0.91 \\ 0 \\ \dots \\ 1 \\ \dots \\ 0 \end{pmatrix}$$

Classification et localisation conjointes

Quelle fonction de coût ?



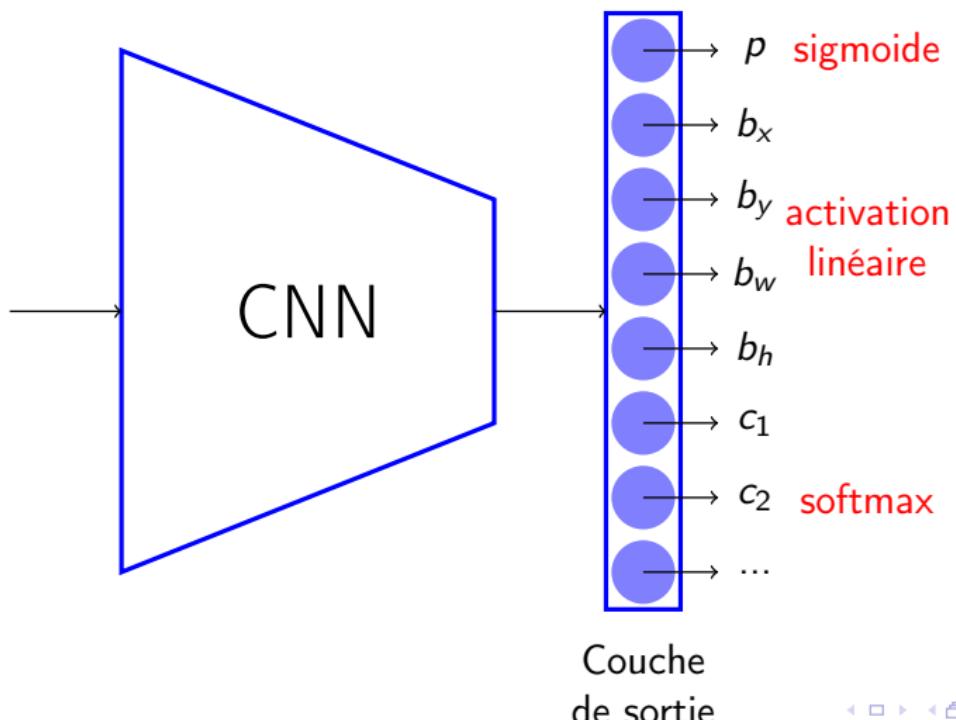
Classification et localisation conjointes



$$\hat{y} = \begin{pmatrix} 1 \\ 0.52 \\ 0.46 \\ 0.5 \\ 0.91 \\ 0 \\ \dots \\ 1 \\ \dots \\ 0 \end{pmatrix}$$

Classification et localisation conjointes

Quelle fonction de coût ?



$$\begin{aligned}\mathcal{L} = & \text{entropie croisée} \quad \lambda_1 \mathcal{L}_1 \\ & + \\ & \text{MSE} \quad \lambda_2 \mathcal{L}_2 \\ & + \\ & \text{entropie croisée} \quad \lambda_3 \mathcal{L}_3\end{aligned}$$

Plan du cours

1 Localisation

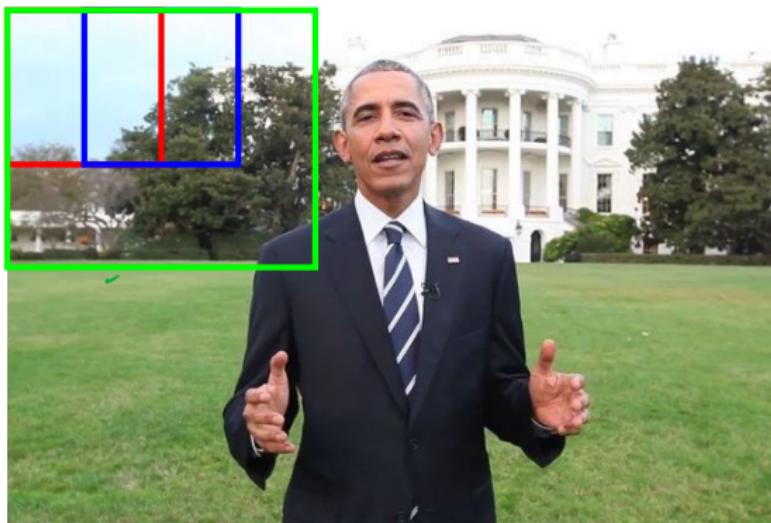
2 Détection d'objet

3 Segmentation d'image

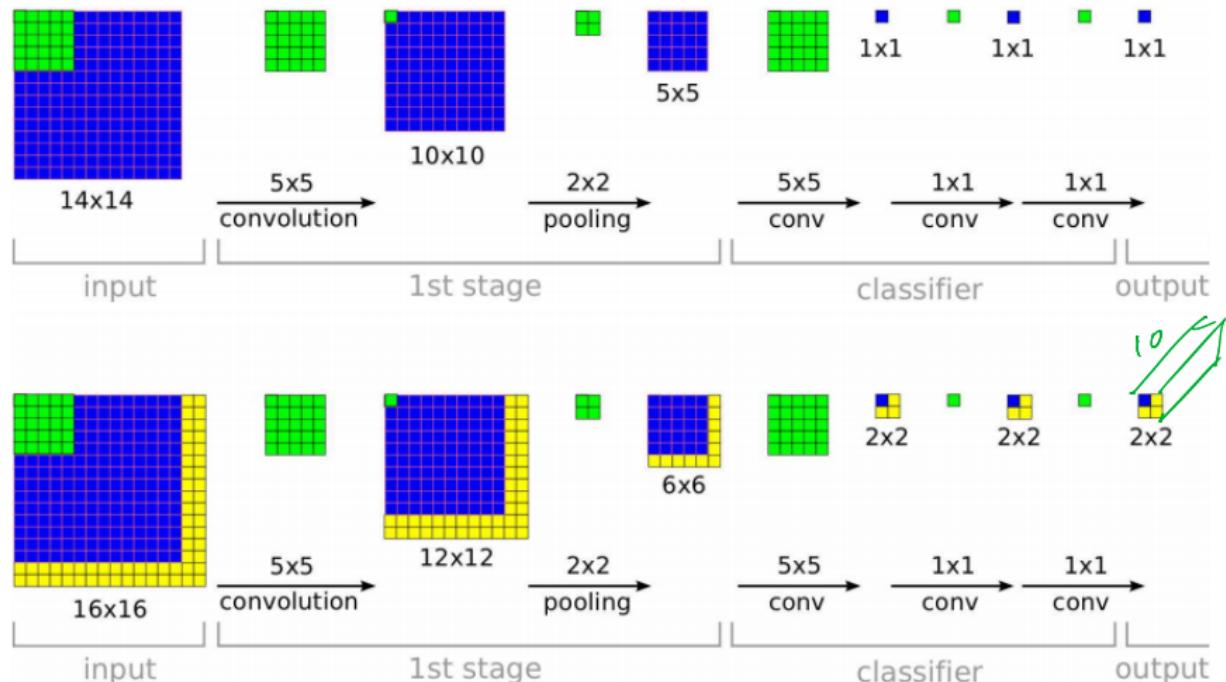
4 Estimation de pose

De la localisation à la détection d'objet

Une première idée : utiliser une fenêtre glissante multi-échelle pour localiser sur chaque sous-partie de l'image les objets présents.



Fenêtre glissante convolutionnelle



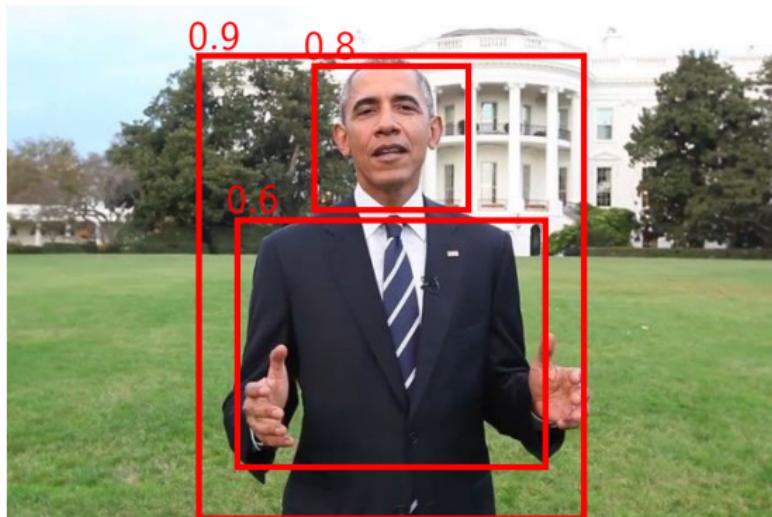
[Sermanet et al.] OverFeat : Integrated Recognition, Localization and Detection using Convolutional Networks

Fenêtre glissante convolutionnelle



[Sermanet et al.] OverFeat : Integrated Recognition, Localization and Detection using Convolutional Networks

Suppression des doublons (*Non-Max Suppression*)



- ① Suppression des boîtes avec un indice de confiance < 0.6 .
- ② Sélection parmi les boîtes restantes de la boîte b_{max} avec le plus haut indice de confiance.
- ③ Suppression de toutes les boîtes ayant un coefficient de Jaccard > 0.5 avec b_{max} .

Famille des R-CNN

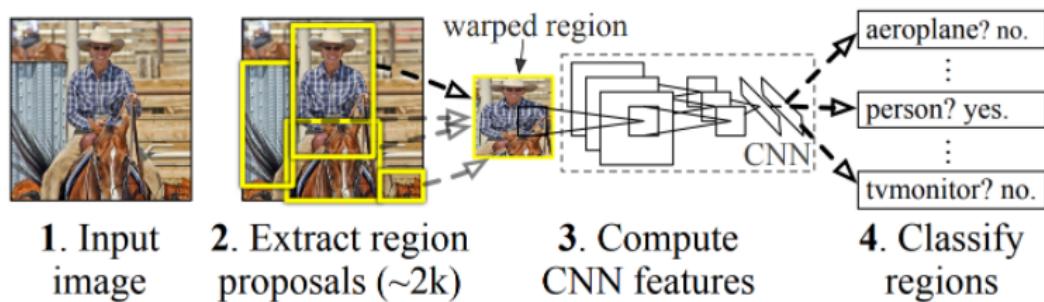
Pour éviter les fenêtres glissantes : génération de régions candidates.



[Uijlings et al.] Selective Search for Object Recognition

R-CNN

Génération des régions candidates et classification de ces régions.

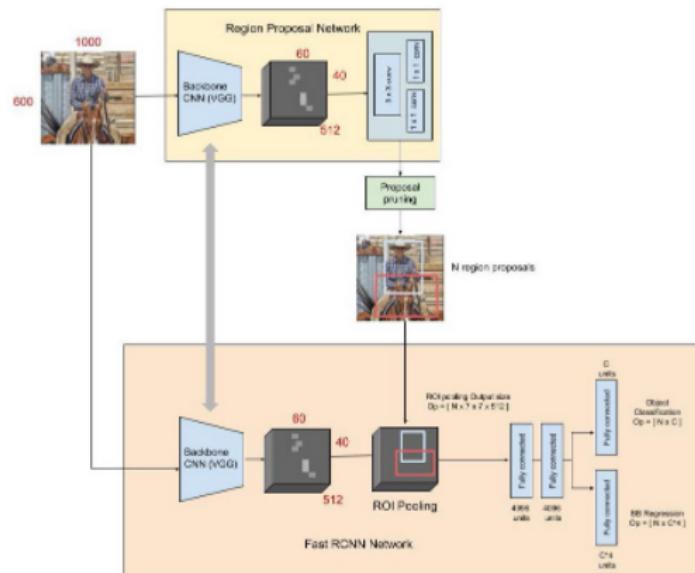


Inconvénient de ces méthodes : nécessité de faire de nombreuses prédictions par un CNN pour chaque image. **Très peu optimisé !**

[Girshick et al.] Rich feature hierarchies for accurate object detection and semantic segmentation

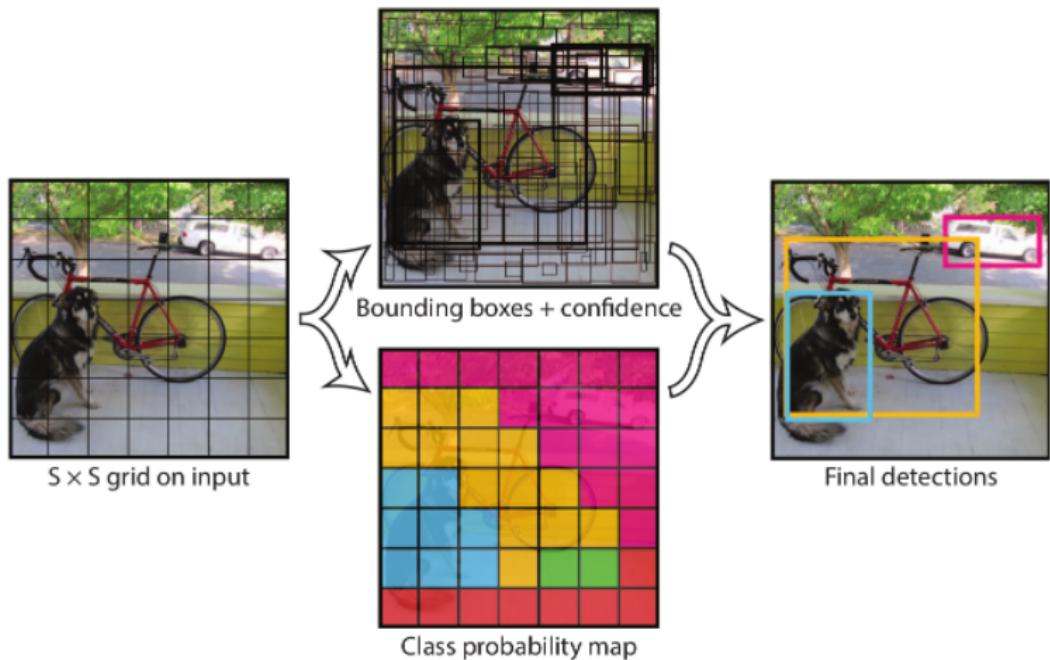
Faster R-CNN

Un même réseau (poids partagés) pour générer des régions candidates et évaluer si elles contiennent un objet (plus rapide mais pas temps réel !)



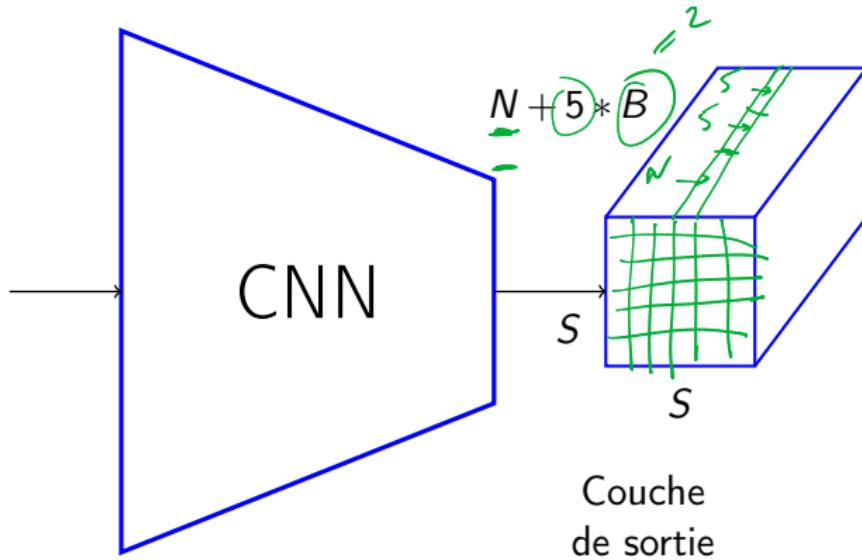
[Ren et al.] Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks

YOLO



[Redmon et al.] You Only Look Once : Unified, Real-Time Object Detection

YOLO



où :

- S est la taille de la grille (sur PASCAL VOC, $S = 7$)
- B est le nombre d'objets détectés par cellule (PASCAL : $B = 2$)
- N est le nombre de classes (PASCAL : $N = 20$)

YOLO : fonction de perte

~~hypothèses~~

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

~~MSE~~

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right]$$

~~Amplitude~~

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2$$

~~λ_{coord}~~

$$+ \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Une erreur quadratique moyenne sur tous les termes !

de YOLO à YOLOv??

Un grand nombre d'améliorations sont réalisées, par exemple :

- Utilisation de "modèles" de boîtes englobantes (*anchor boxes*, YOLO v2).
- Entraînement, et prédiction, multi-échelle.
- Évolution du CNN utilisé comme base.
- Modification de la manière d'encoder les boîtes englobantes.
- Nouvelles augmentations de données (e.g. mosaicing)
- Suppression des *anchor boxes* (YOLO "v8")

et nombreuses autres subtilités...

Pour certaines, ces modifications sont reprises des autres algorithmes de détection d'objet (Faster-RCNN, SSD, etc.)

Plan du cours

1 Localisation

2 Détection d'objet

3 Segmentation d'image

4 Estimation de pose

La base de données MS COCO



plus de 300K images, et plus de 2,5M d'instances d'objets segmentés (plus de 22 heures-humain pour 1000 segmentations)

[Lin et al.] Microsoft COCO : Common Objects in Context.

Les défis de la segmentation



Un grand nombre de prédictions : une pour chaque pixel de l'image.

Les défis de la segmentation



La prédiction pour un pixel donné nécessite une analyse d'information locale conjointe à une compréhension plus globale de la scène.

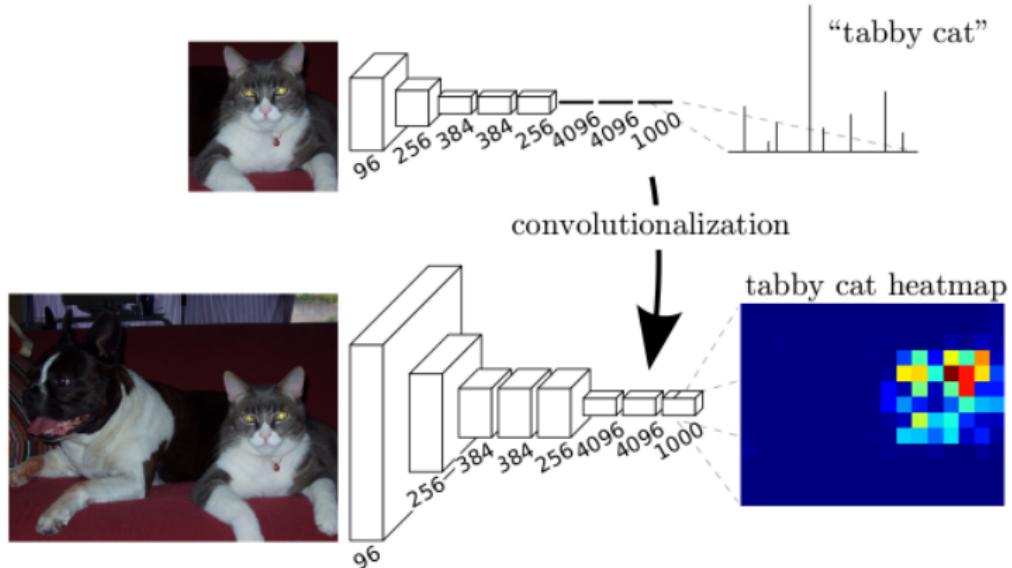
Segmentation par fenêtre glissante



Inconvénients :

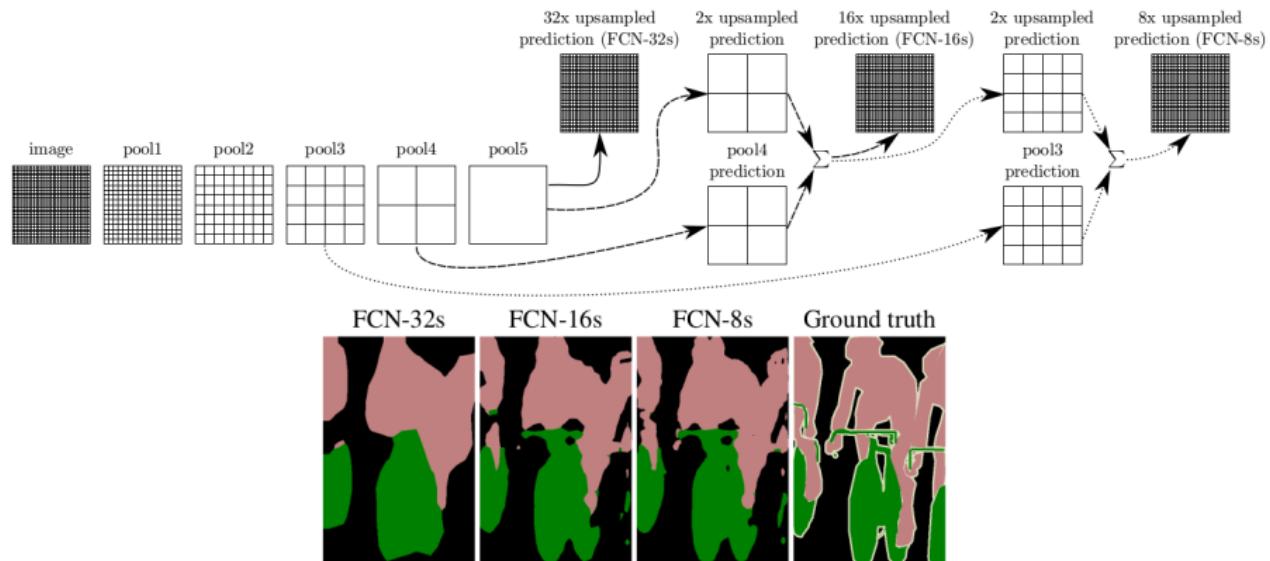
- Très coûteux.
- Pas de prise en compte de l'image globale.

Fully Convolutional Networks (2014)



- Premier algorithme à implanter une segmentation *end-to-end*.
- Repose sur une reformulation des réseaux entièrement convolutifs.
- Base VGG-16.

Fully Convolutional Networks (2014)



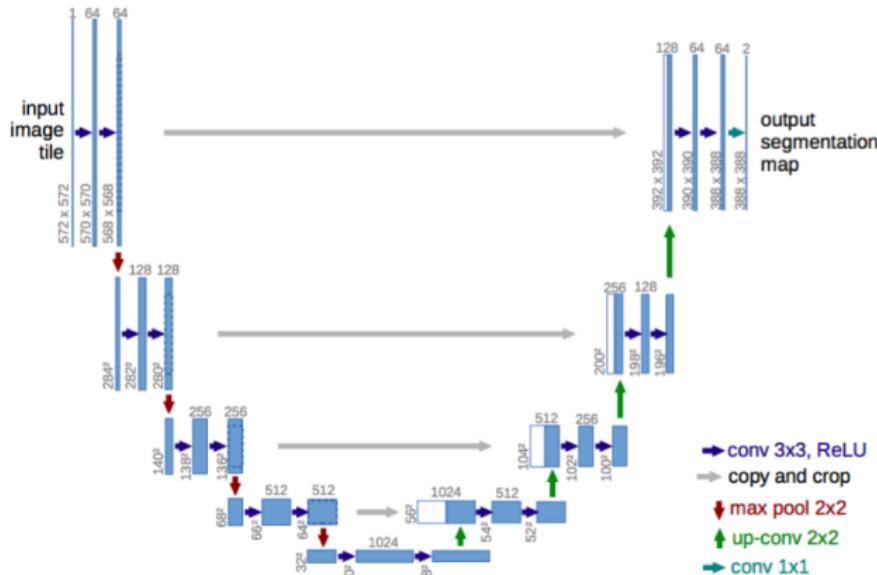
- Prédictions successivement affinées en prenant en compte des tenseurs intermédiaires.

[Long et al.] Fully Convolutional Networks for Semantic Segmentation.

Un mot sur les fonctions de coût

- Chaque pixel porte une fonction softmax afin de déterminer quelle classe est la plus probable.
- On utilise l'entropie croisée pour comparer la classe prédite d'un pixel et la classe réelle.
- La fonction de coût finale est la moyenne des entropies croisées sur l'ensemble des pixels de l'image.

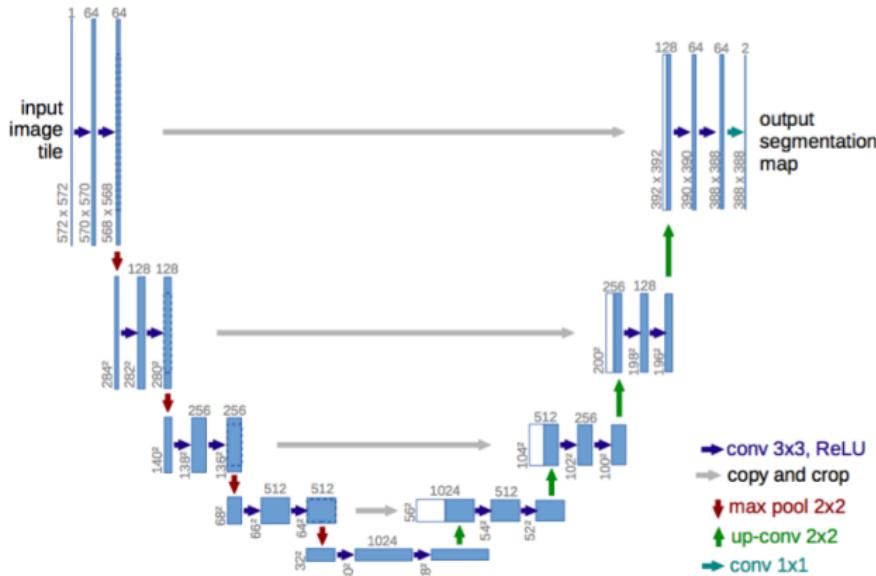
UNet (2015)



- Une évolution de FCN.
- Probablement un des réseaux les plus utilisés pour la segmentation !

[Ronneberger et al.] U-Net : Convolutional Networks for Biomedical Image Segmentation.

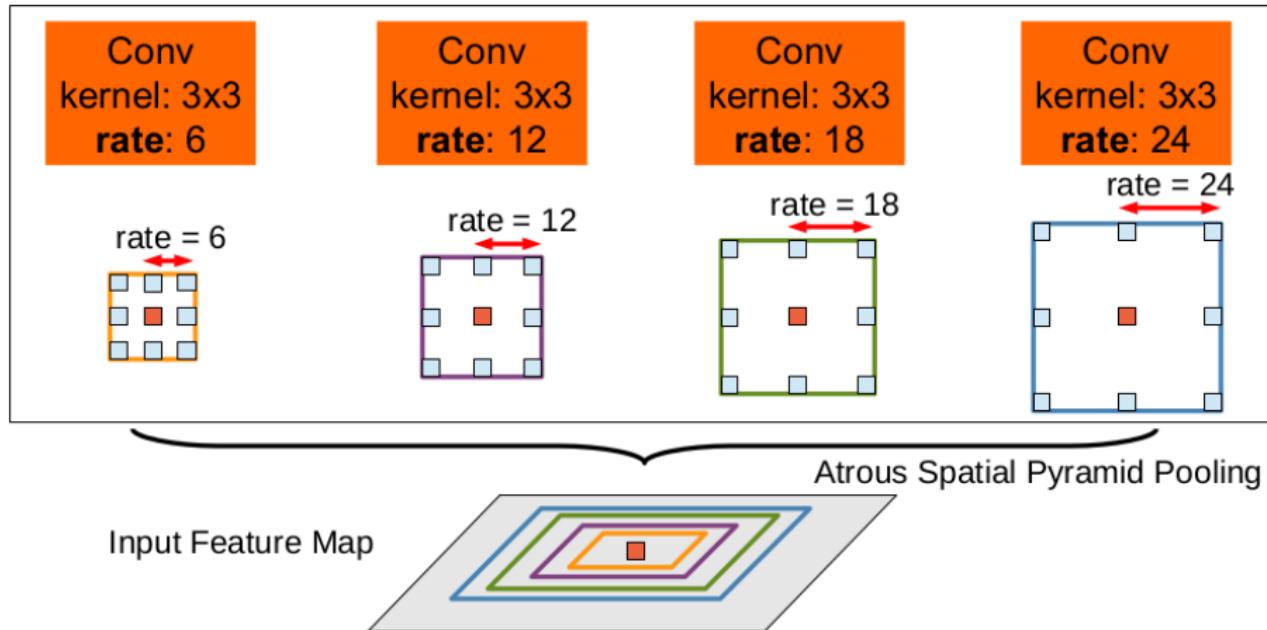
UNet (2015)



Architecture adaptable avec n'importe quelle base convolutive !

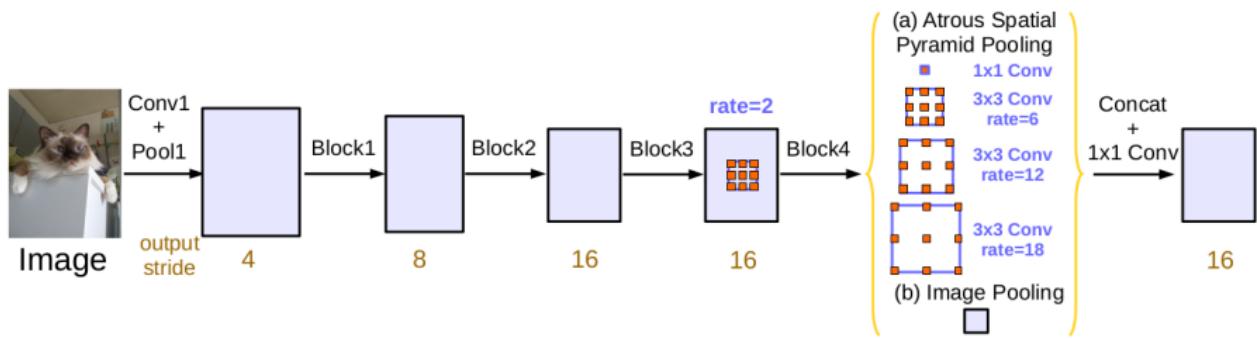
[Ronneberger et al.] U-Net : Convolutional Networks for Biomedical Image Segmentation.

Pyramide spatiale (DeepLab v3)



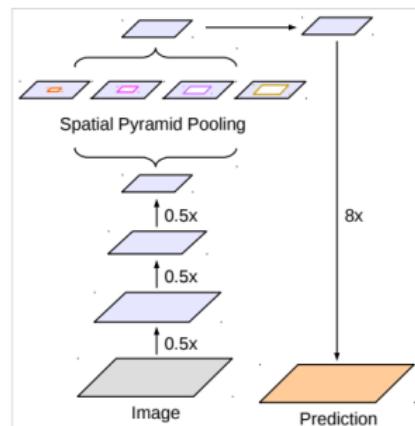
[Chen et al.] Rethinking Atrous Convolution for Semantic Image Segmentation

DeepLab v3

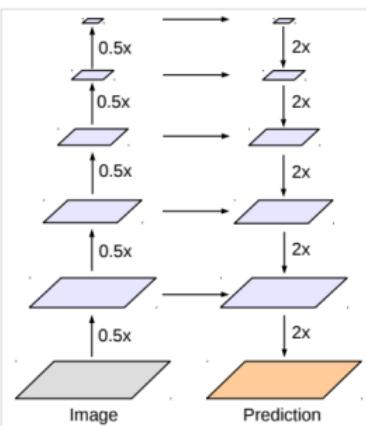


[Chen et al.] Rethinking Atrous Convolution for Semantic Image Segmentation

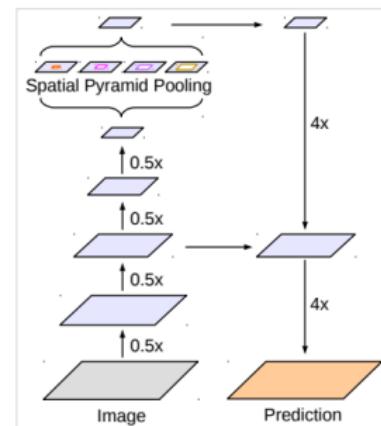
DeepLab v3+



(a) Spatial Pyramid Pooling



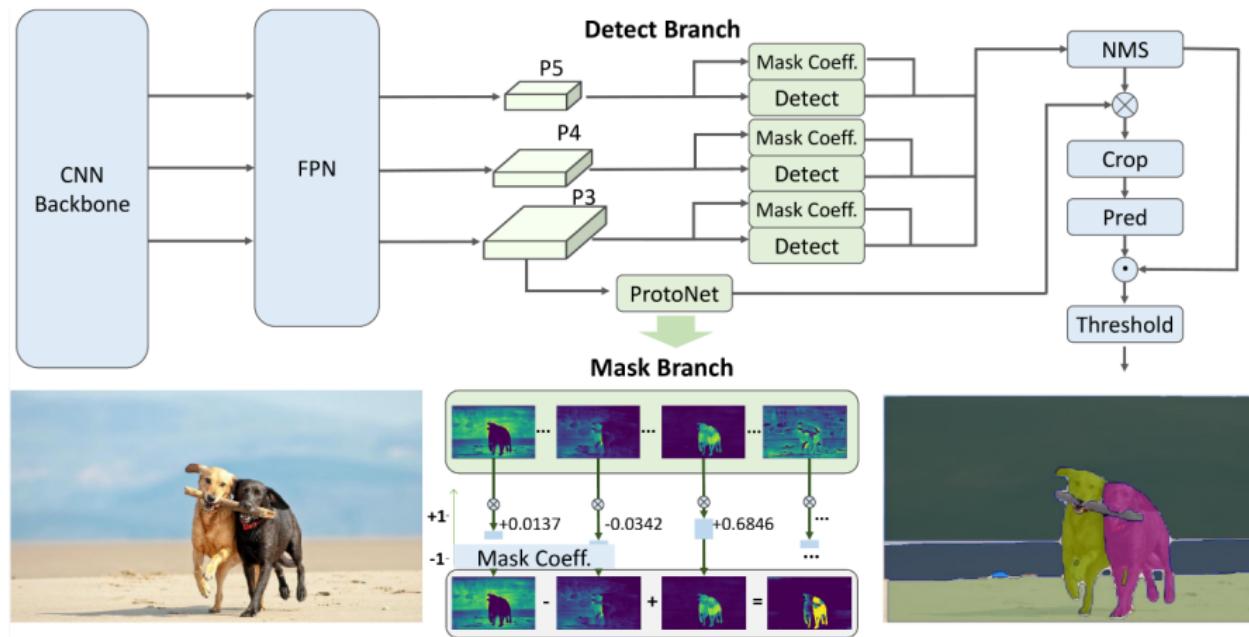
(b) Encoder-Decoder



(c) Encoder-Decoder with Atrous Conv

[Chen et al.] Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation

Fast Segment Anything (2023)



Entraînement sur 11 millions d'images, annotées avec plus d'un milliard de masques (!!)

[Zhao et al.] Fast Segment Anything

Bilan sur la segmentation

Deux enjeux principaux :

- **Densité de la couche de sortie** : augmentation de la dimension par deconvolution ou convolution "à trous".
- **Traitement multi-échelle** : forme en U (*auto-encoder*), pyramide spatiale.

Challenges :



Plan du cours

1 Localisation

2 Détection d'objet

3 Segmentation d'image

4 Estimation de pose

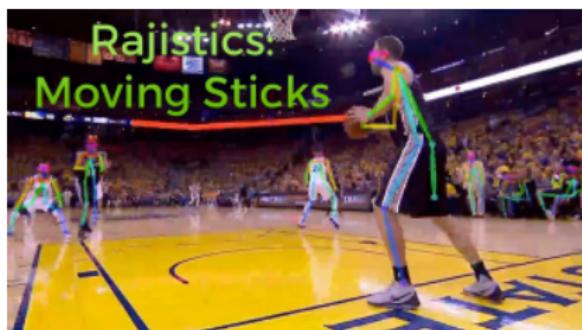
Keypoints Challenge MSCOCO 2019



200000 images, 250000 personnes, 1,7M de joints.

Estimation de pose

Problème de localisation des "joints", ou articulations, d'une personne sur une image.

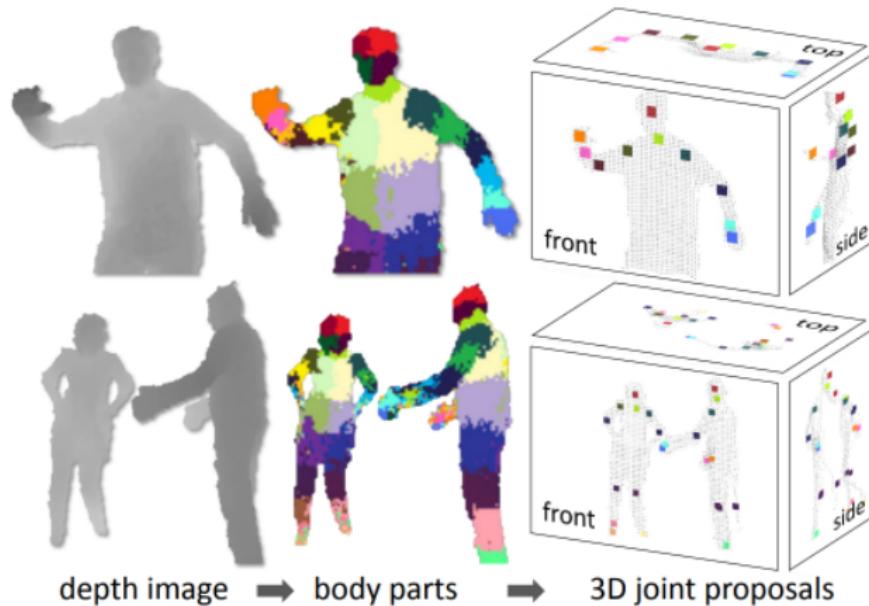


Nombreuses applications : reconnaissance d'actions, animation, jeux (Just Dance), *sports analytics*, etc.

Difficulté : prédictions locales mais nécessité d'une compréhension globale de la scène (comme pour la segmentation)

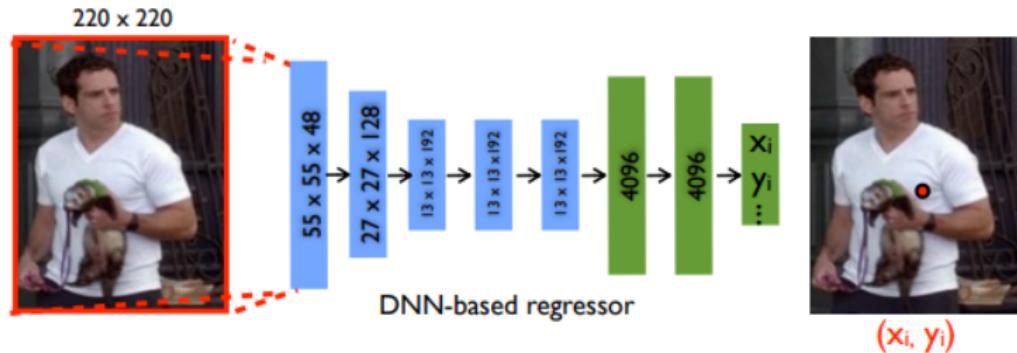
Estimation de pose

Approche "classique" implantée dans la Kinect (2011) :



Segmentation par parties du corps (forêts aléatoires) puis estimation de mode (*mean-shift*) pour l'extraction des joints.

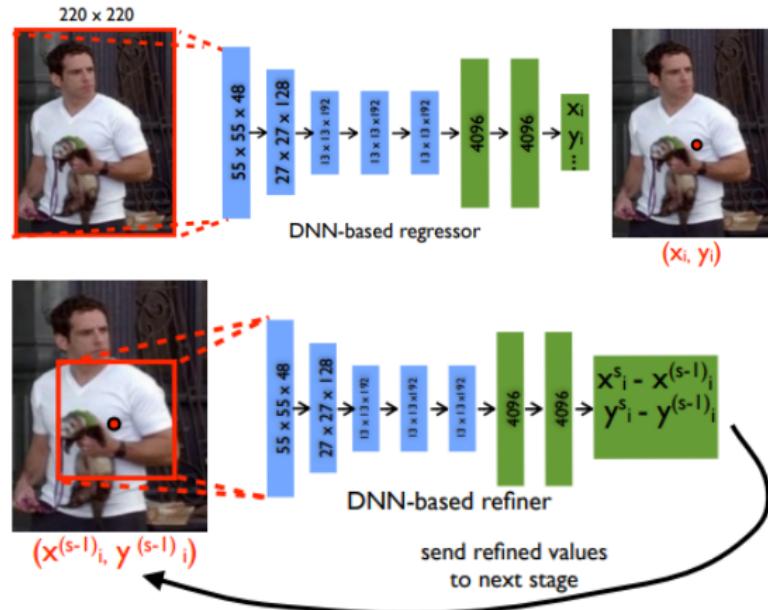
DeepPose (2014)



Utilisation d'AlexNet comme base convolutive, prédiction directe des coordonnées des joints. Augmentation **massive** des données ($\times 40$) !

[Toshev et al.] DeepPose : Human Pose Estimation via Deep Neural Networks

DeepPose (2014)



Affinage de la prédiction en repassant une sous-région de l'image dans le réseau de neurones.

[Toshev et al.] DeepPose : Human Pose Estimation via Deep Neural Networks

Prédiction de cartes de chaleur

Une idée introduite en 2015 consiste à prédire des cartes de chaleur pour chaque joint plutôt qu'une position.



Le problème se rapproche alors d'un problème de segmentation binaire.

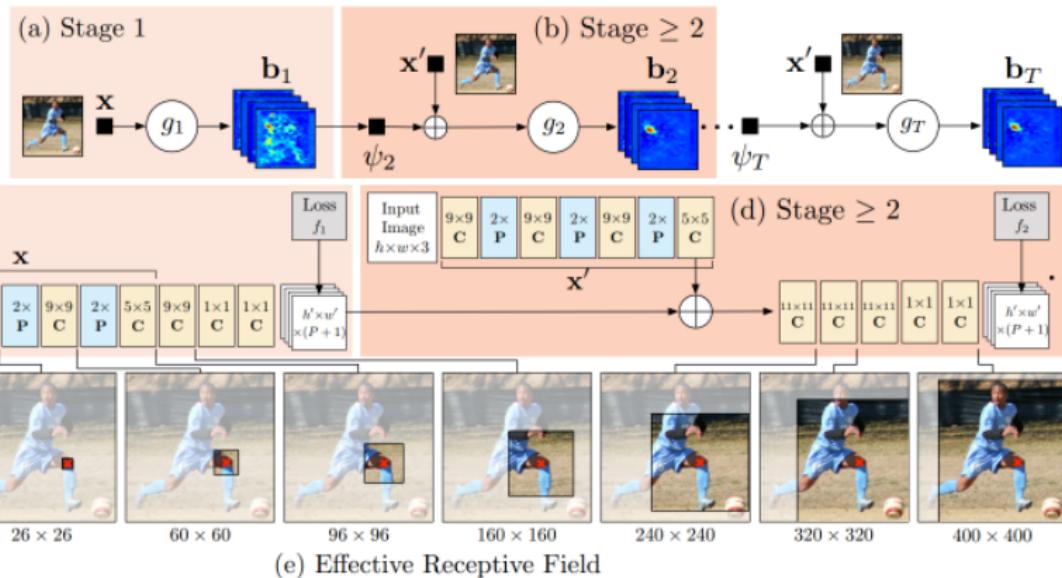
[Tompson et al.] Efficient Object Localization Using Convolutional Networks

OpenPose (2016-2018)

Convolutional
Pose Machines
(T -stage)

P Pooling

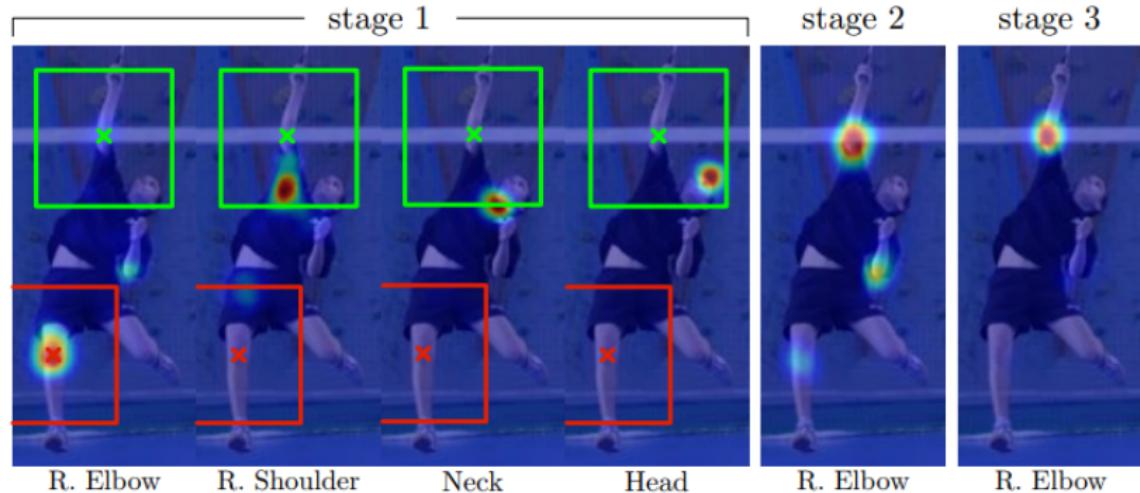
C Convolution



Affinage des prédictions en prenant en compte des premières versions des prédictions.

[Wei et al.] Convolutional Pose Machines

OpenPose (2016-2018)



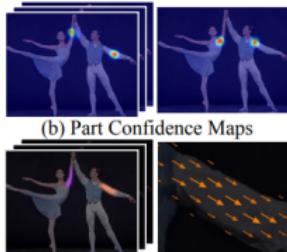
Les joints estimés avec certitude servent d'*a priori* aux joints pour lesquels il existe une ambiguïté.

[Wei et al.] Convolutional Pose Machines

OpenPose (2016-2018)



(a) Input Image



(b) Part Confidence Maps

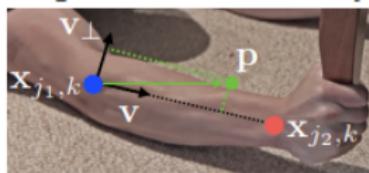
(c) Part Affinity Fields



(d) Bipartite Matching



(e) Parsing Results



Estimation conjointe des probabilités de présence des joints, et d'un champ de vecteur permettant dans une phase de post-traitement de connecter les joints.

[Wei et al.] OpenPose : Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields