UNIVERSITÉ LAVAL

GLO-4030/7030 APPRENTISSAGE PAR RÉSEAUX DE NEURONES PROFONDS

Détection

Segmentation

Hiver 2020

Philippe Giguère

Vision : tâches principales

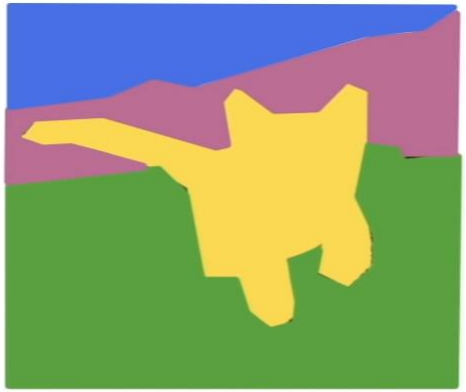
Classification

CAT

Détection

DOG, DOG, CAT

Segmentation



GRASS, CAT, TREE, SKY

Segmentation d'instances

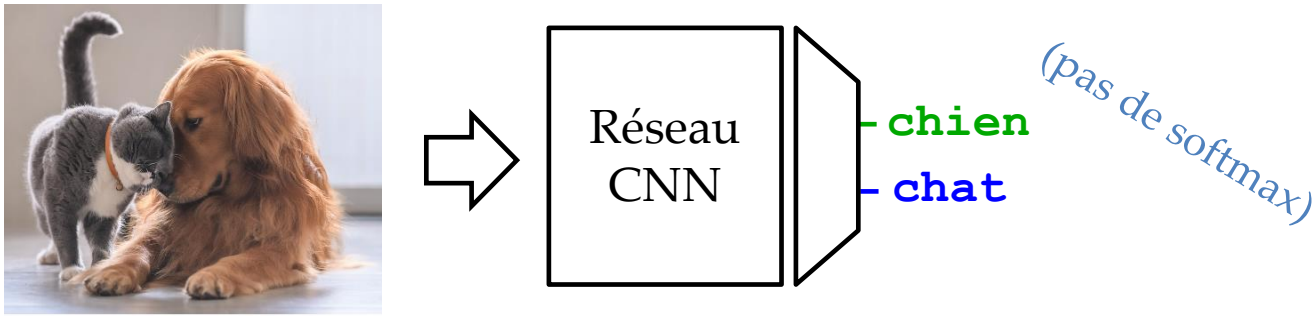
DOG, DOG, CAT

Adapté de cs231n

Détection

Détection

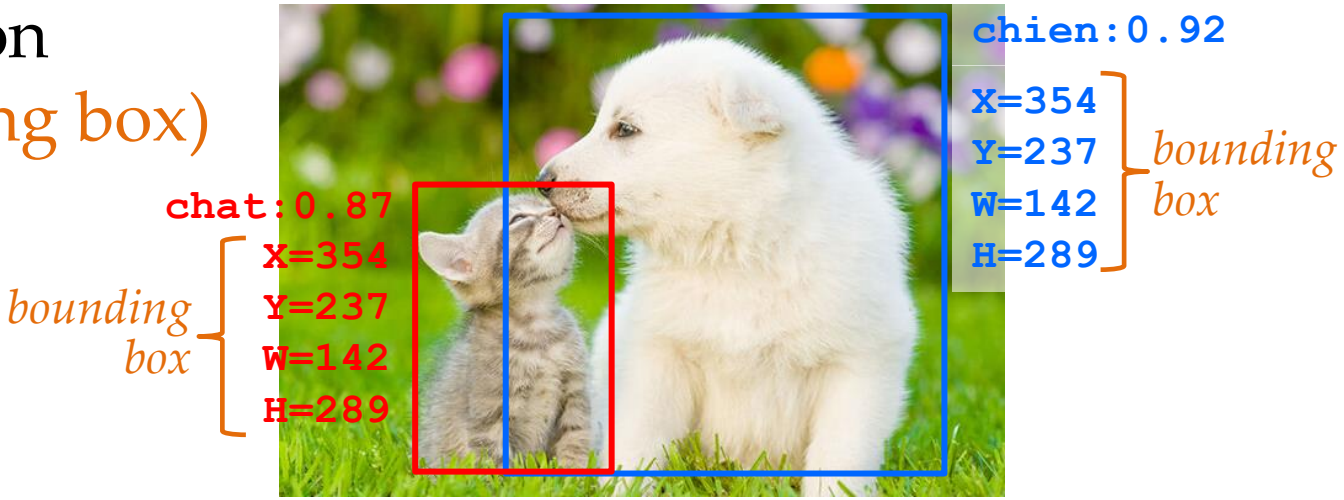
- Une des difficultés est que l'on ne connait pas le **nombre exact d'instances** dans l'image
  - si on savait d'avance que les images ne contiennent au maximum qu'un chat et un chien :



- Sujet de recherche très fertile en soit

Détection : définition

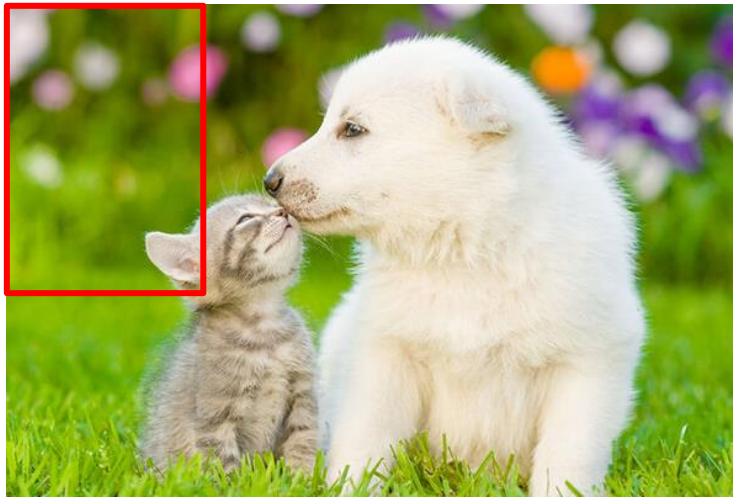
- Pour :
  - **une image d'entrée**
  - **une liste prédéterminée de classes**
- Trouver, pour tous les objets présents :
  - la position (bounding box)
  - la classe



- Nombre de prédictions **va varier d'une image à l'autre**
  - Architecture doit en tenir compte

Détection via classification

- Possible de faire de la détection par un réseau de classification
- Approche par **fenêtre coulissante** (crops)
- Passe chaque crop dans un réseau classificateur
- Conserve *n* prédictions les plus confiantes



- Choix de la géométrie de la fenêtre :
  - taille, aspect ratio
- **Fastidieux**, car des milliers de passes dans le réseau classificateur : dizaines de secondes

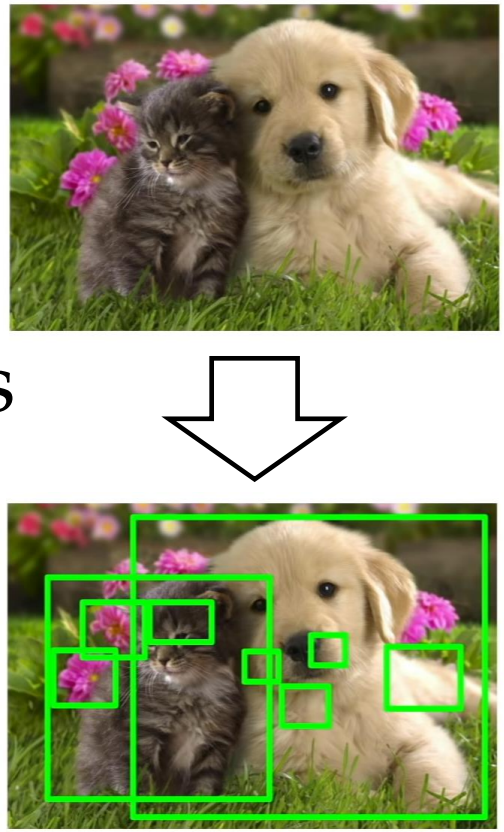
Catégories d'algorithmes

- **Basé sur des régions proposées** *(region proposal)*
  - R-CNN
  - Fast-RCNN
  - Faster RCNN

- **Grille fixe** *(grid-based)*
  - YOLO (v1, v2, v3)
  - SSD (single-shot detection)

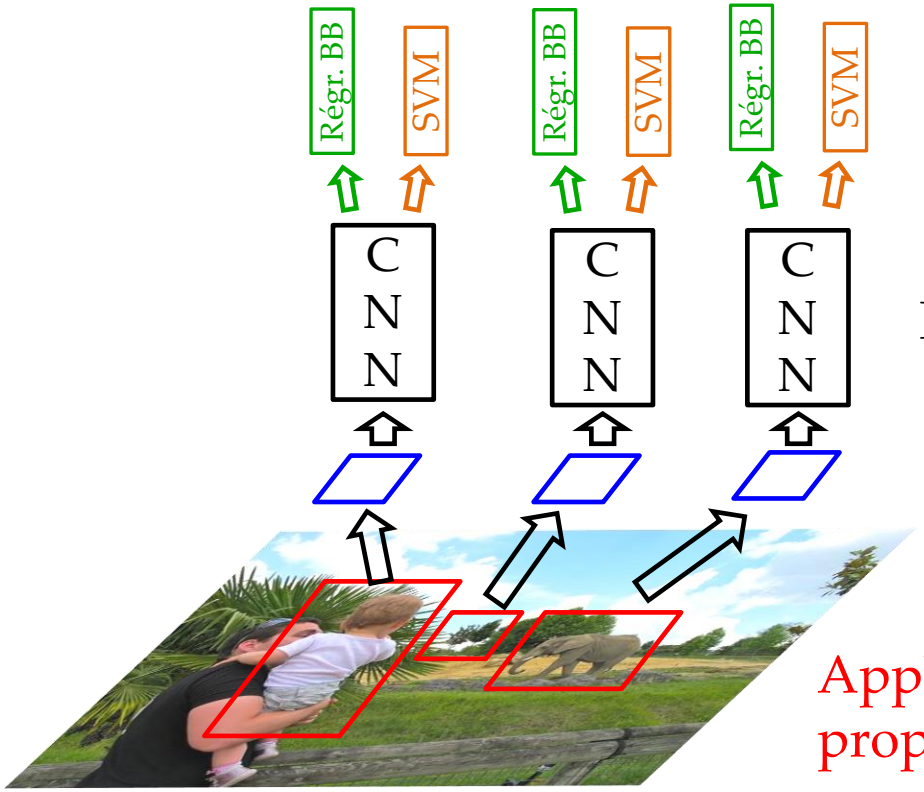
Region proposal (classique)

- Algorithmes qui proposent des régions **prometteuses** en terme de présence d'objets
- Basés parfois sur des heuristiques
  - Recherche de *blobs*
  - Distributions particulières de contours (*edge*)
- `Selective Search` propose 1000 régions en quelques seconds sur CPU *(pas temps-réel)*
- Voir aussi `Edge Boxes`



R-CNN (2014)

- RoI *(region of interest)* sont des rectangles
- Ajout d'une classe background
- Lent : inférence en 47 secondes par image, à cause de l'algorithme de proposition de régions



Ajustement de la *bounding box* via tête régression

Classification avec SVM

Passe dans un backbone CNN

Déforme vers taille unique d'image d'entrée

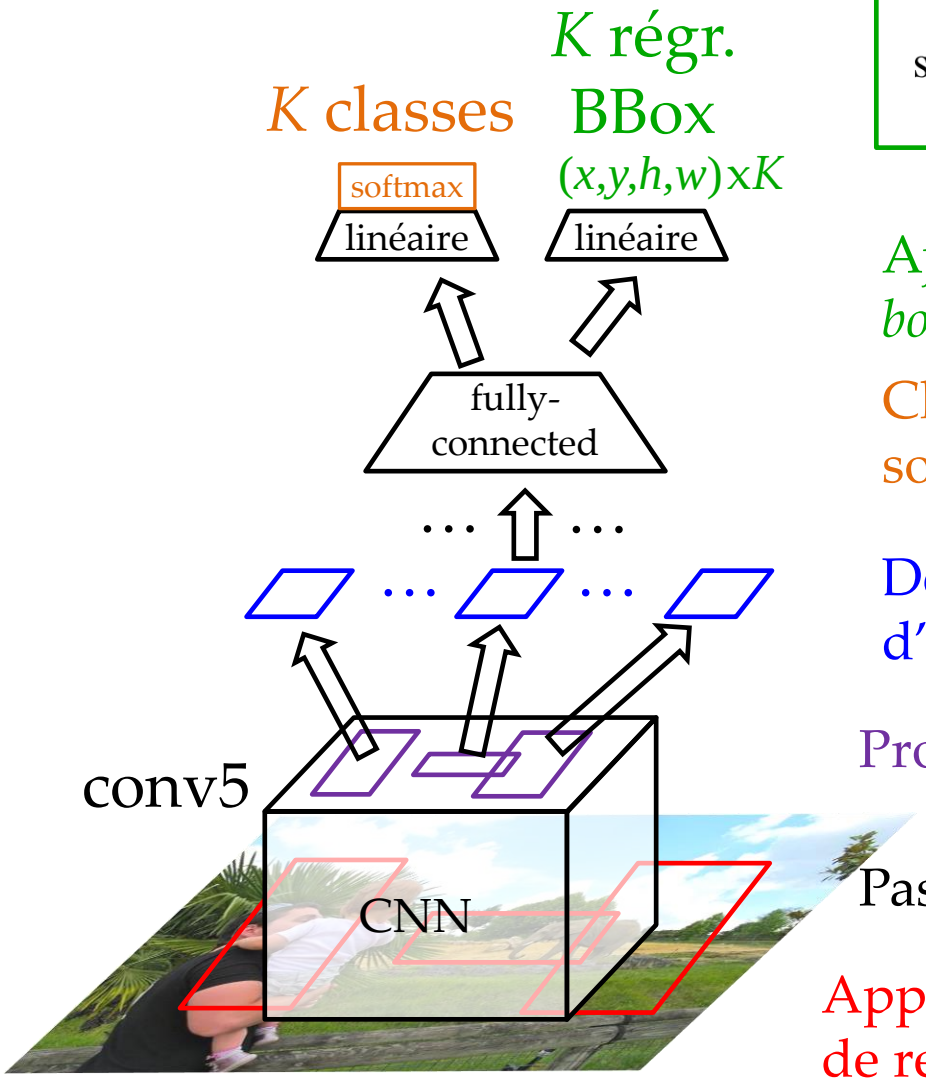
Applique un algorithme de proposition de régions (RoI) (2k)

adapté de cs231n

Fast R-CNN (2015)

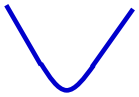
Temps de calcul

- **2 sec Prop RoI**
- 0.3 sec partie réseau



Perte régression

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$



Ajustement des la *bounding box* via tête de régression

Classification avec réseau linéaire + softmax

Déforme vers taille unique d'image d'entrée

Projette les RoI dans le feature map

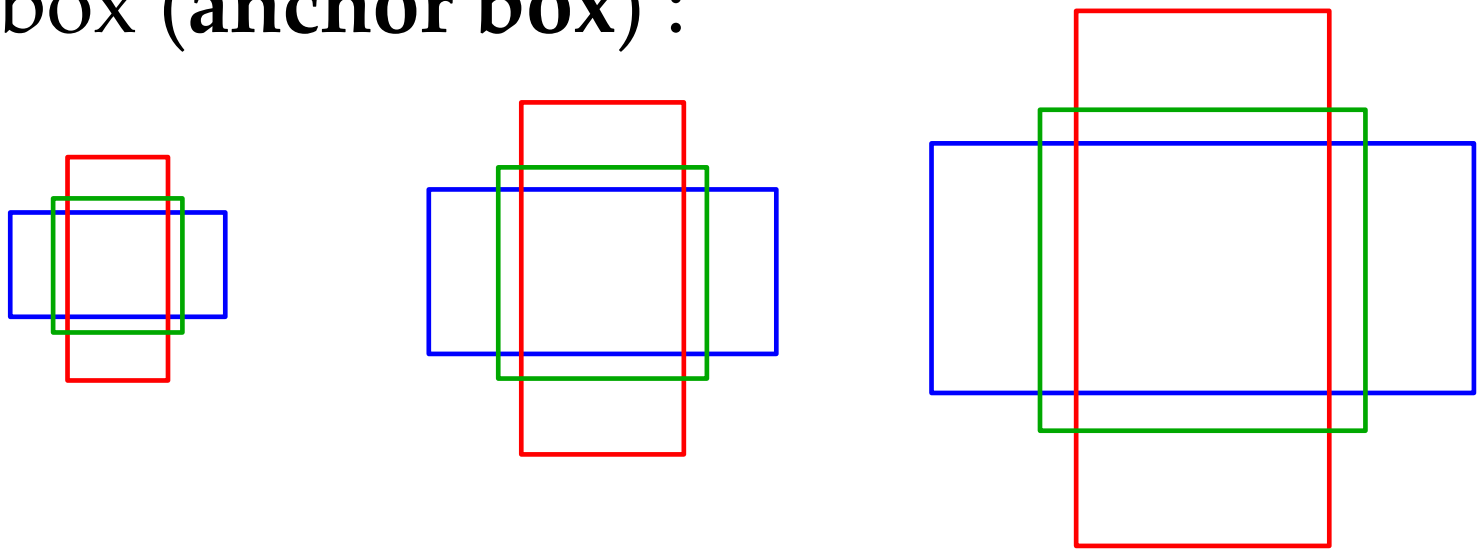
Passe dans un backbone CNN

Applique un algorithme de proposition de régions (RoI) (Selective Search)

adapté de cs231n

Faster R-CNN

- Laisser le réseau faire les propositions : **RPN (Region Proposal Network)**
- Accélère grandement le processus
- RPN s'améliore via backprop des pertes
- Introduction de 9 prototypes de bounding box (**anchor box**) :

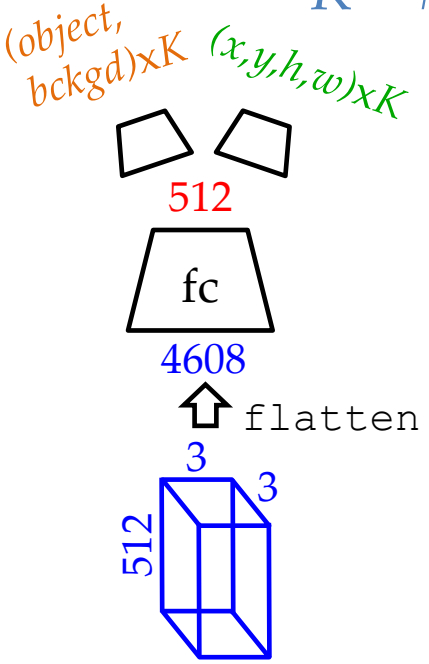


Faster R-CNN : Fully-convolutional

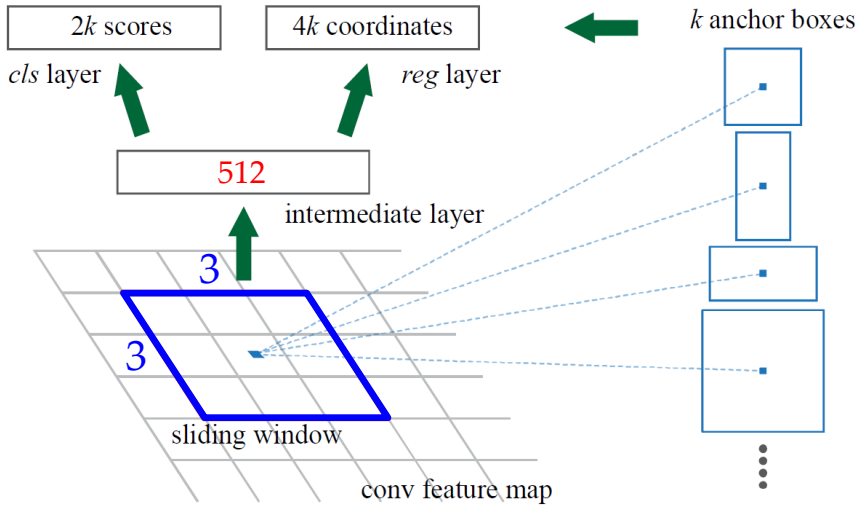
(néglige biais)
paramètres

512x(2+4)xK

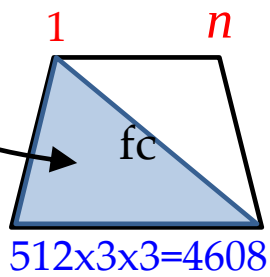
512x3x3x512



K = # bounding boxes



filtre conv 3x3 !



Appliquer un réseau *fully-connected* avec *n* sorties, de manière coulissante = Appliquer *n* filtres de convolution

Fully-convolutional network (FCN)

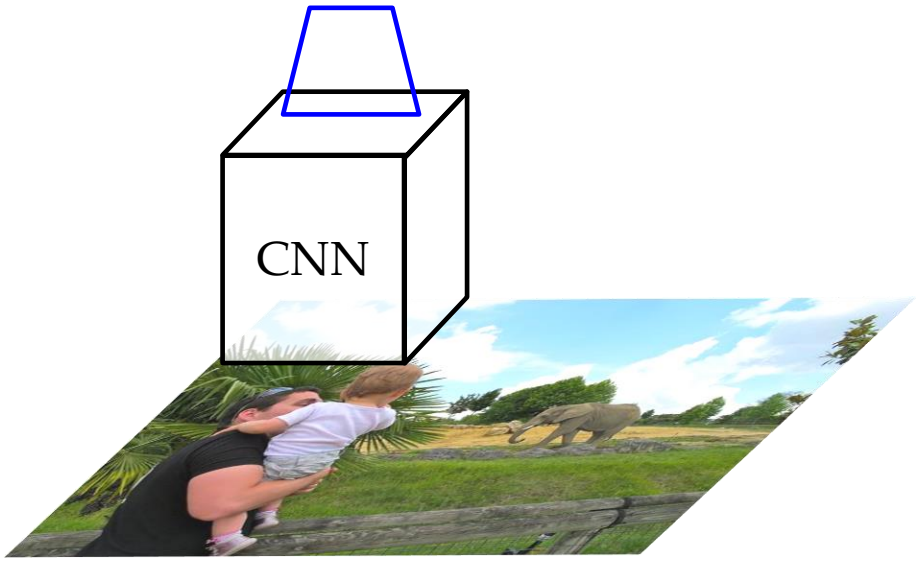
Approche détection par fenêtre coulissante mais **très efficace**!

Faster R-CNN : Fully-convolutional

- Coulissage via convolution : très rapide

Détection par coulissage « traditionnel »

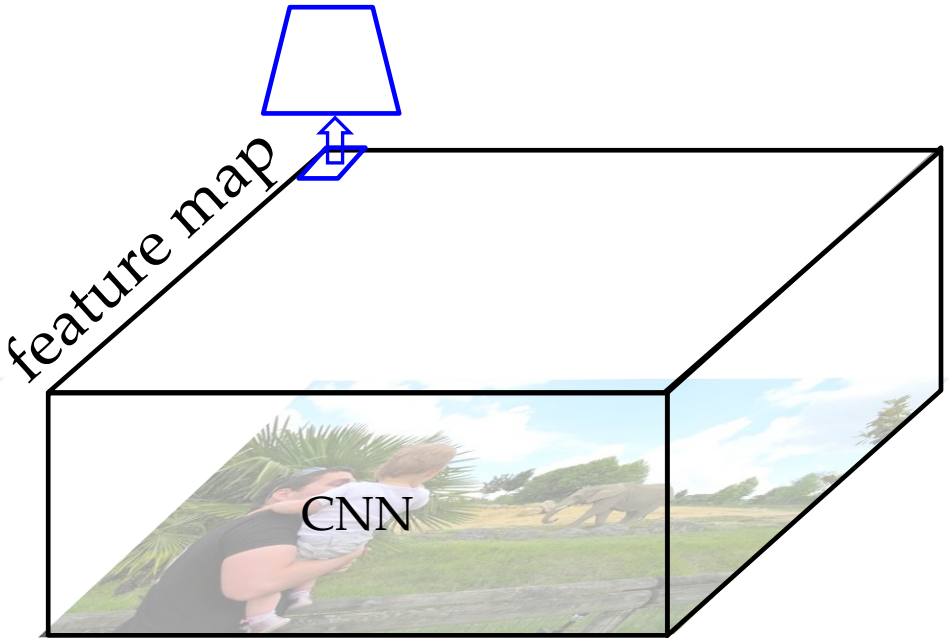
têtes de prédictions



Faster R-CNN

…simplement en ajoutant quelques couches de convolution en guise de têtes

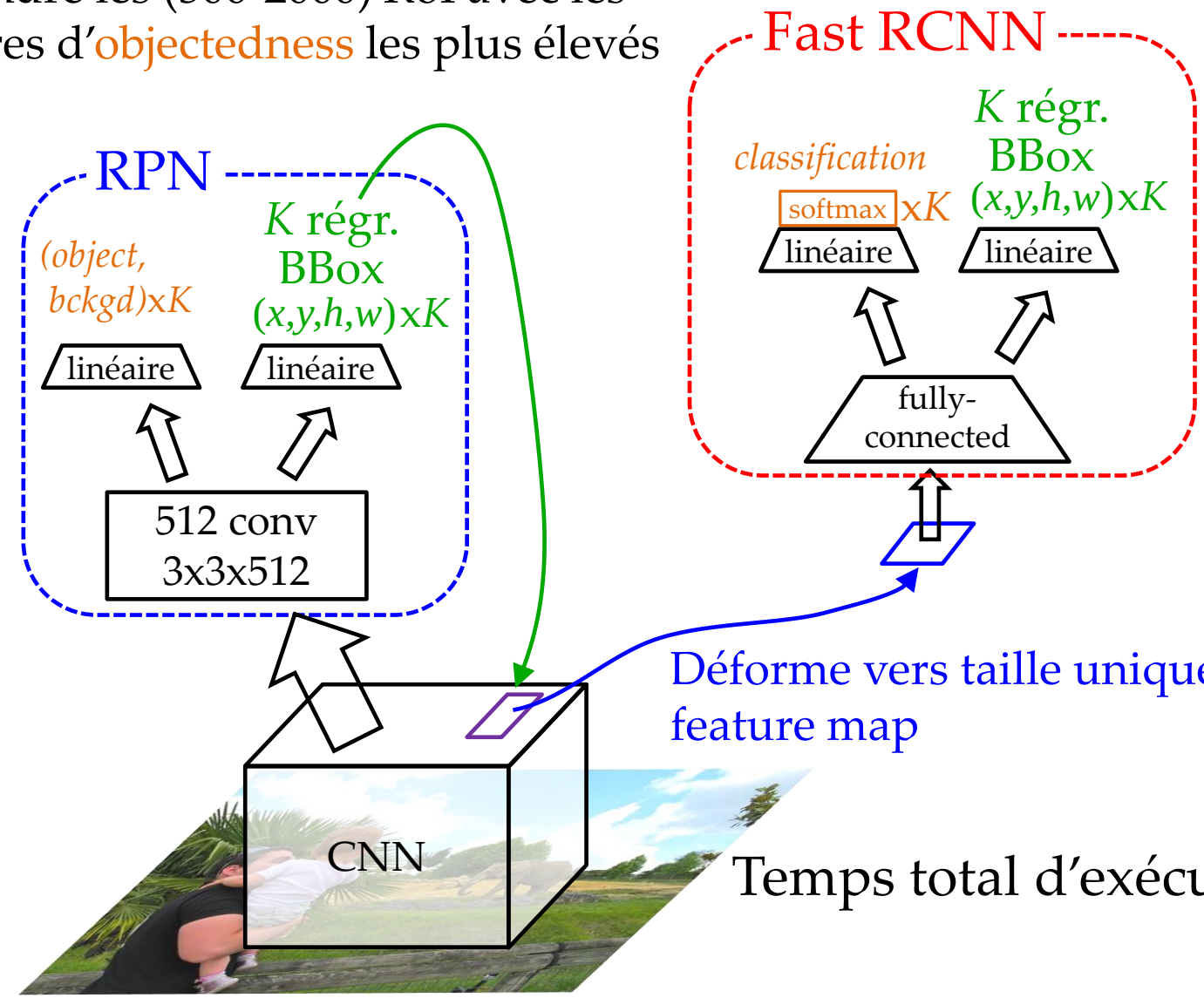
têtes de prédictions



Faster R-CNN

Gradients sur les pertes vont aider les deux réseaux

Prendre les (300-2000) RoI avec les scores d'objectedness les plus élevés



Temps total d'exécution : 200 ms!

Approche sans proposal (Yolo v3)

- Grille régulière (7x7 → feature map)
- Pour chaque position centrale de la grille
  - pour chaque anchor box, prédire :
    - classe (incluant la classe background)
    - régression (b_x, b_y, b_w, b_h) sur les paramètres de l'anchor box + confiance (objectedness)
- Prédictions obtenues via Fully-convolutional (FCN) d'une 1x1 : très rapide! (20 ms)