DÉTECTION D'OBJETS

TMI

Plan du cours

Une hiérarchie de problèmes

- Classification vs localisation vs détection vs segmentation
- Classification : CNN, notion de backbone

Détection

- Analyse du problème, évaluation des solutions [Metrics2020]
- Régression de masques [Szegedy2013]
- Fenêtres glissantes multiéchelles [Overfeat2014]
- Proposition de régions : [RCNN2013], [FastRCNN2015], [FasterRCNN2016]
- Single Shot detection : [SSD2016]
- YOLO : [YOLO2016], [YOLOReview2023]

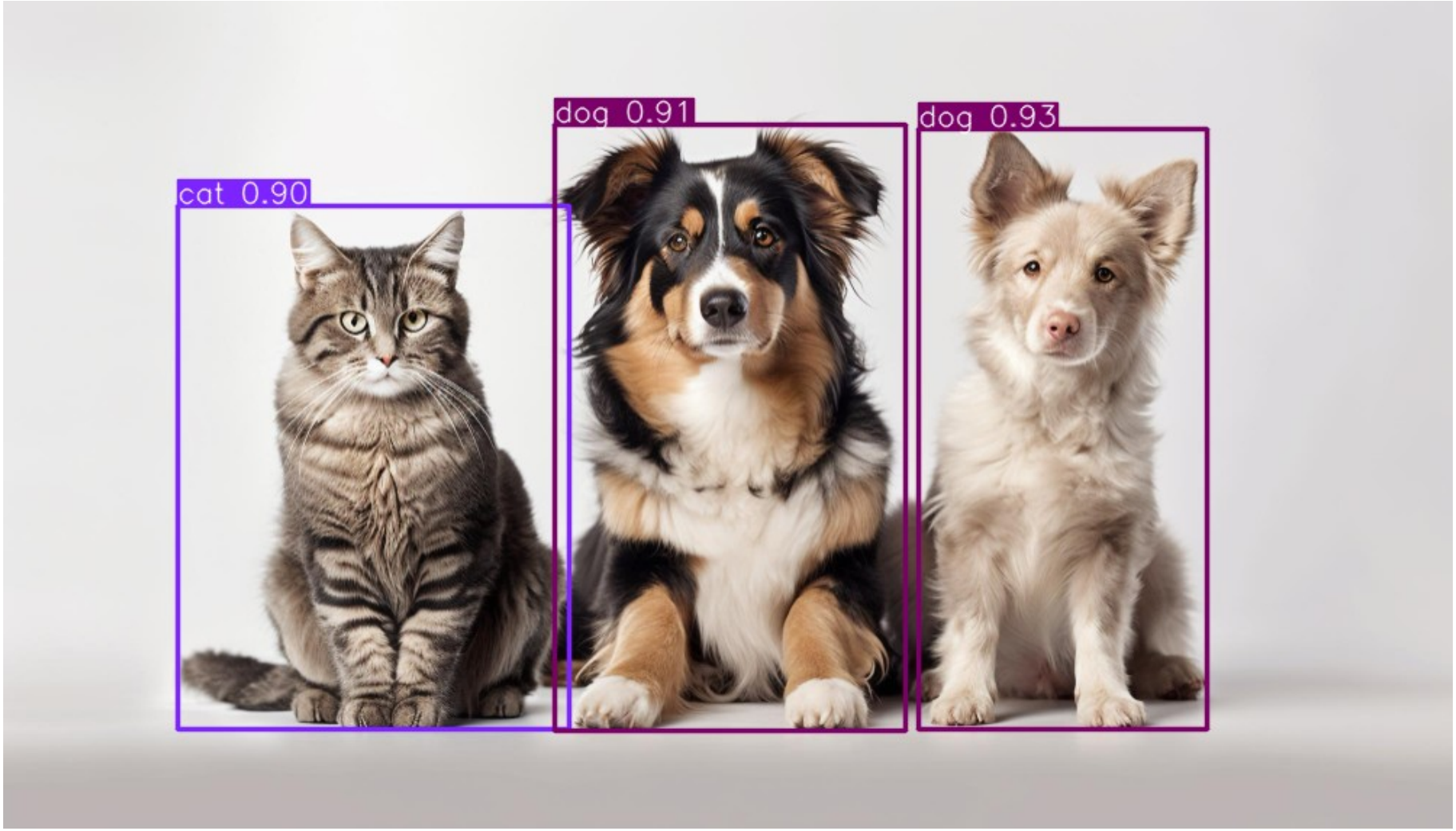
CLASSIFICATION UNE CLASSE

CLASSIFICATION MULTICLASSES

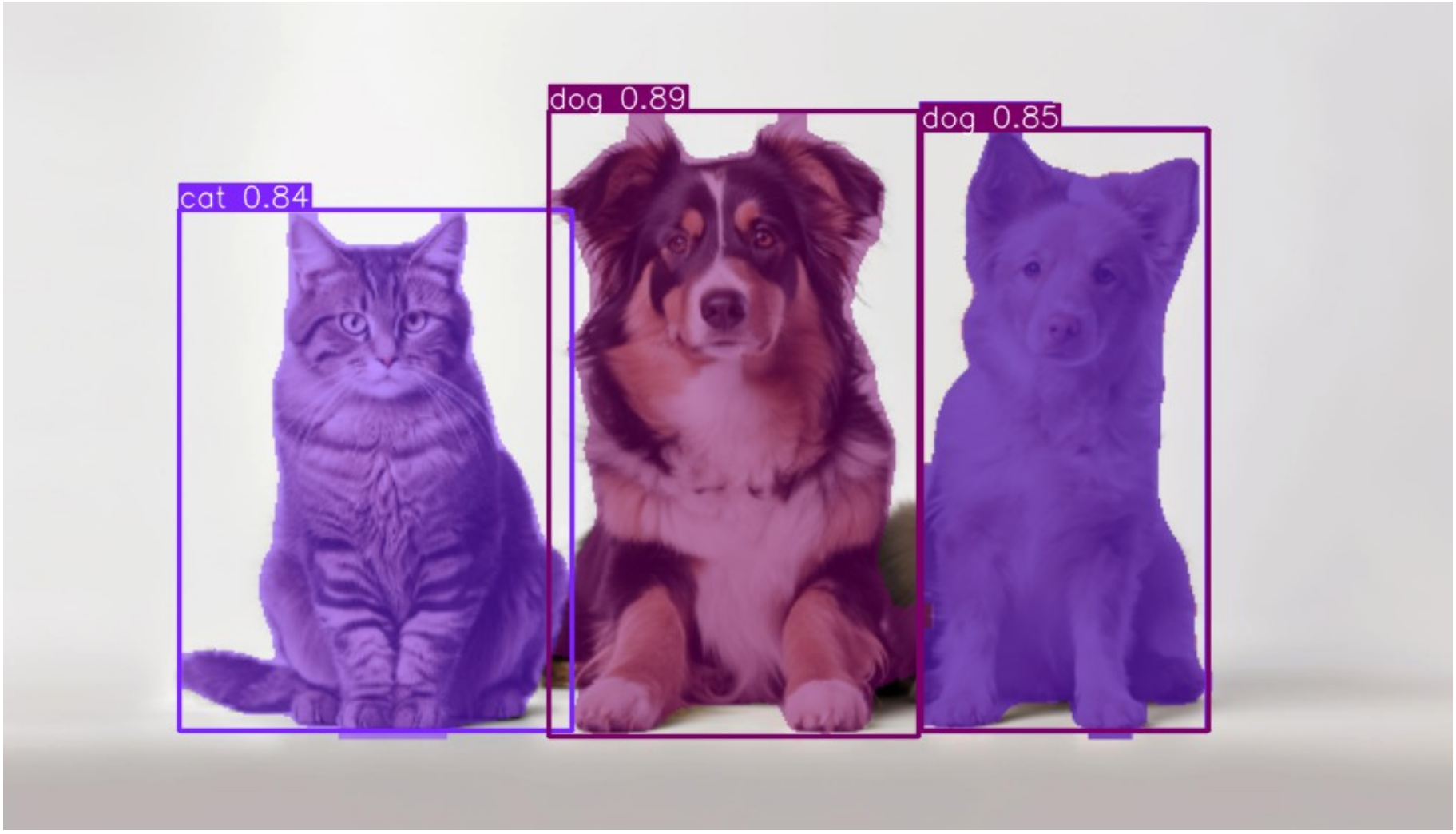
LOCALISATION

LOCALISATION MULTIINSTANCES

Détection



SEGMENTATION SÉMANTIQUE

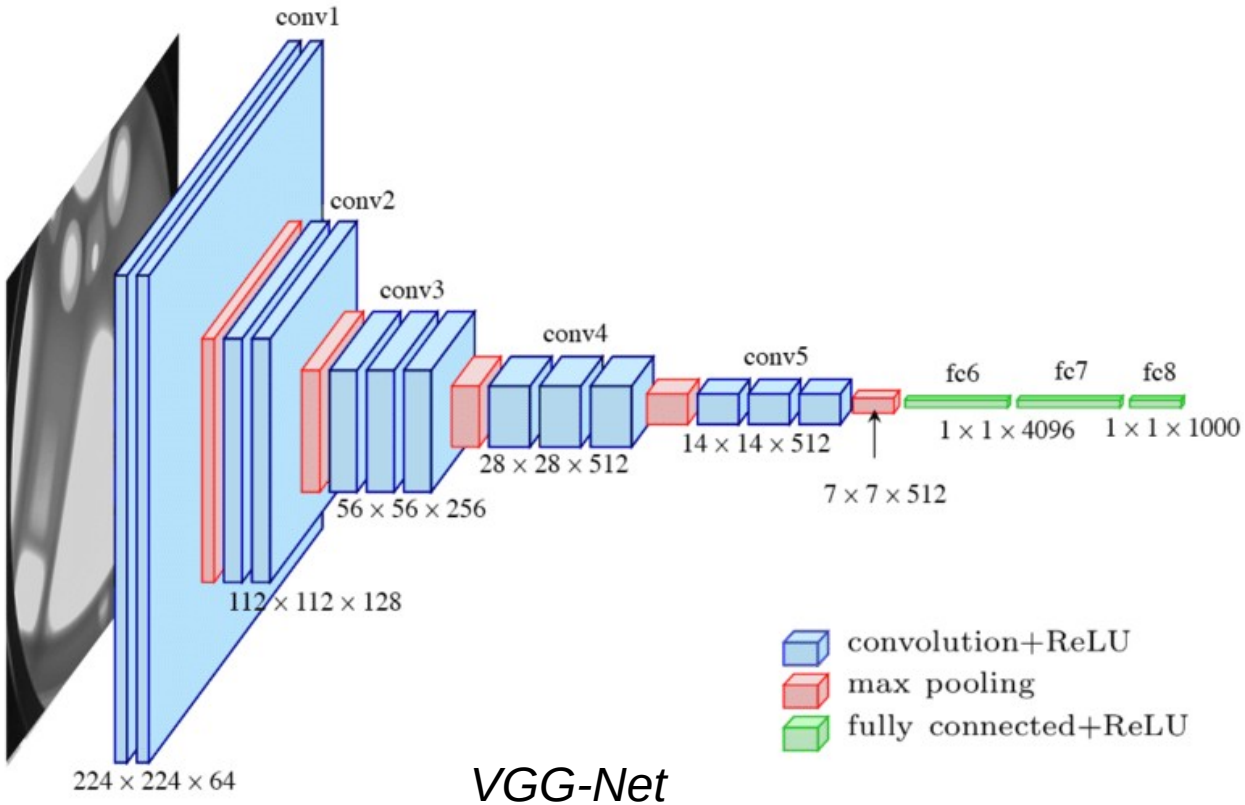


CLASSIFICATION : ARCHITECTURE

Deux blocs

Calcul de caractéristiques (*features, representation*) : CNN, transformer,...

Puis classification sur la base de ces caractéristiques : MLP



VGG-Net

CNN : CONVOLUTION

Définition

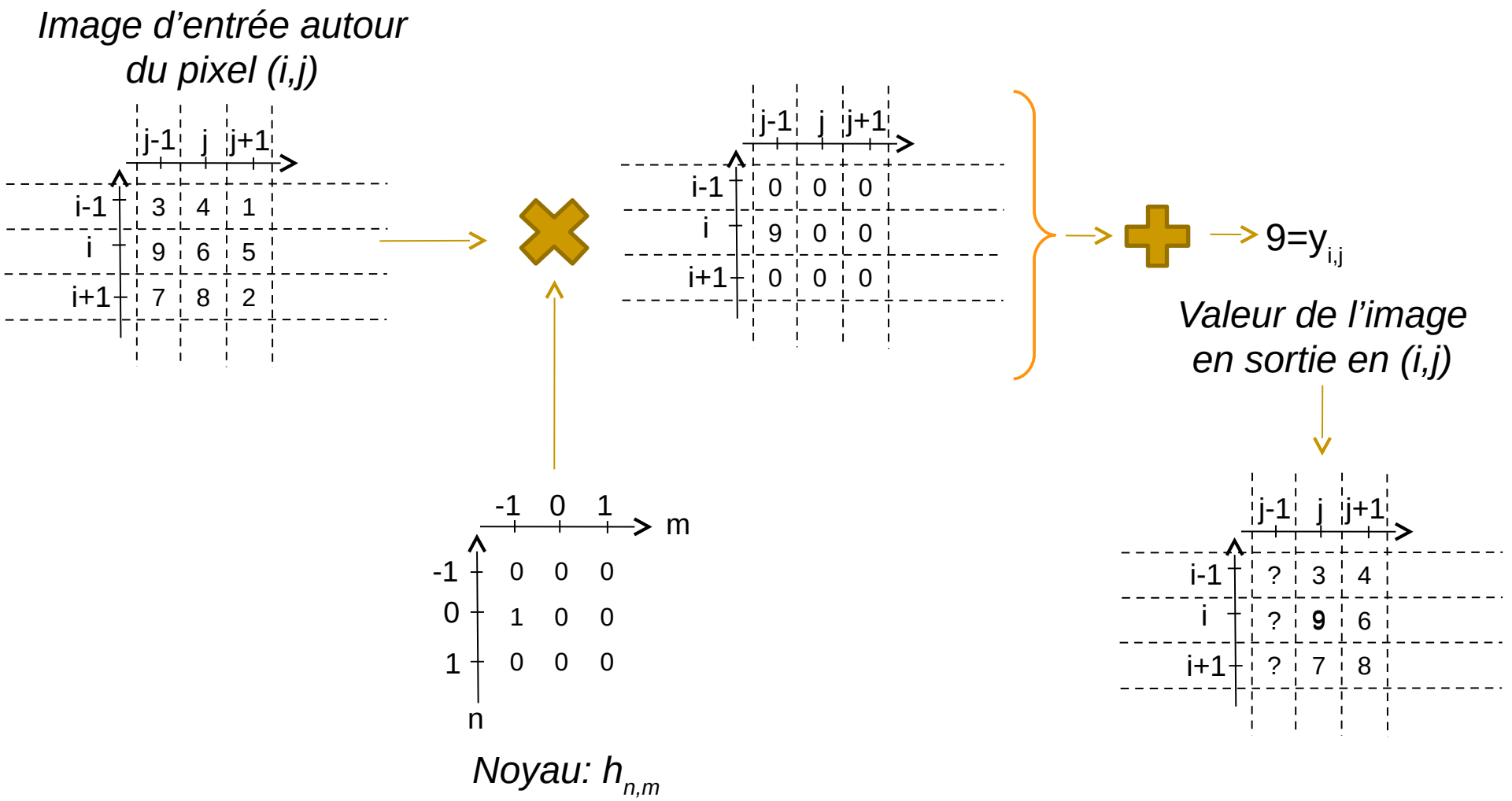
Pondération d'un pixel avec ses pixels voisins

Taille (2N+1)x(2N+1) → n pixels dans chaque direction. Souvent 3x3 ou 5x5

Fenêtre glissante : chaque pixel a pour valeur en sortie, la somme pondérée de ses valeurs voisines

$$\forall i, j \quad y_{i,j} = \sum_{n=-N}^{N} \sum_{m=-N}^{N} x_{i+n, j+m} h_{n,m}$$

CONVOLUTION : EXEMPLE



CONVOLUTION : INTÉRÊT

Mêmes poids pour toute l'image

Réduction du nombre de paramètres

Ex : image 256x256 → 256x256x256x256+256x256 > 2^{32} poids (env. 4,3 milliards)

Si convolution 3x3 → 9 + 1 poids (noyau + biais)

En général, convolution KxK → K^2+1

Un noyau de convolution pour chaque feature en entrée → C_{in}*K^2 +1

1 jeu de paramètres pour chaque feature en sortie (=canal=image) → C_{out}*(C_{in}*K^2+1)

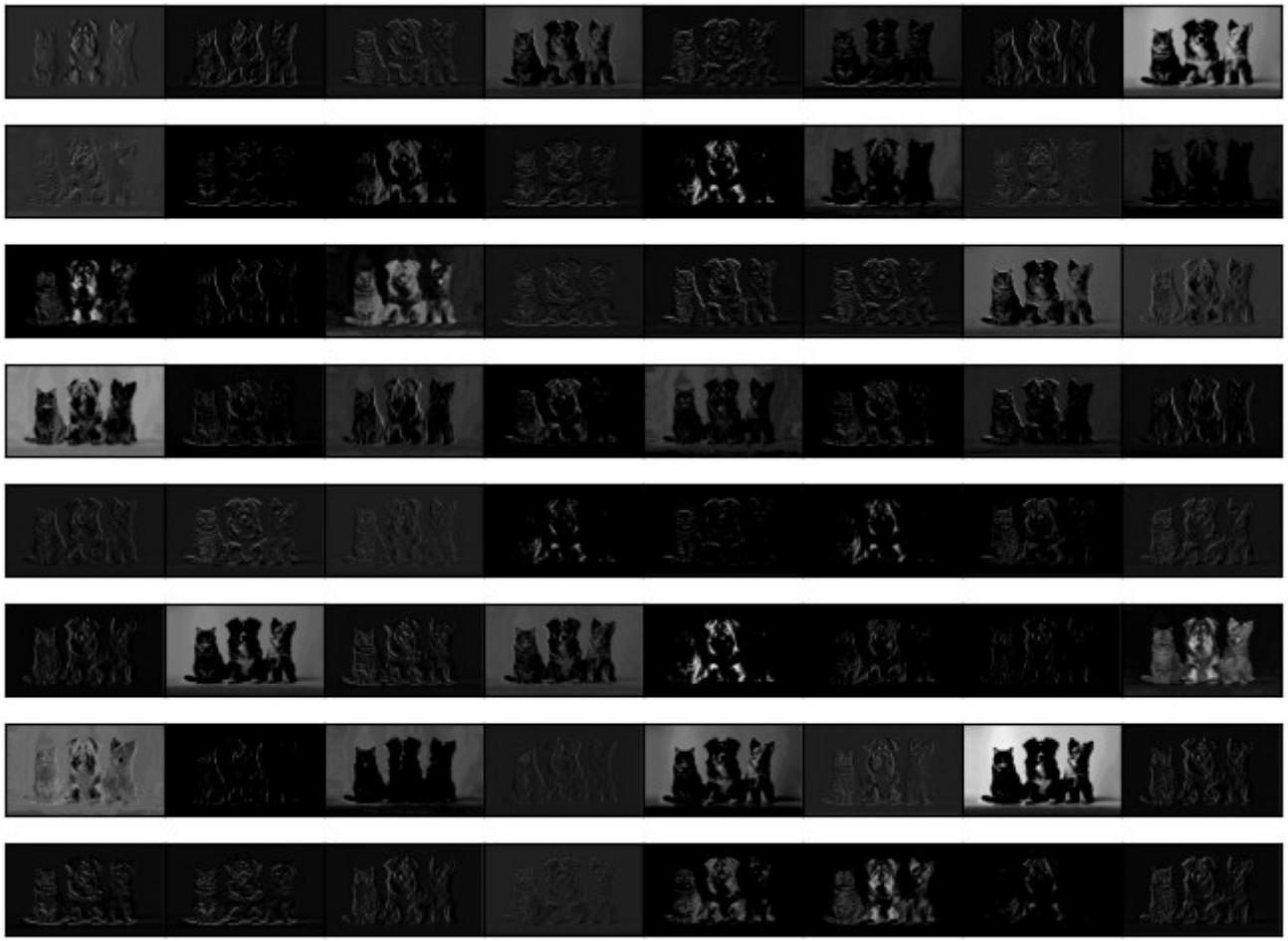
Vocabulaire

kernel_size : taille du noyau (K)

stride : pas entre deux centres de convolution → réduction de la taille de l'image en sortie

padding : valeur à donner aux pixels extérieurs à l'image (convolution aux bords, sur une bande (K-1)/2)

COUCHE DE CONVOLUTION



Exemple de première couche de features pour VGG16

COUCHE DE CONVOLUTION

Exemple de quatrième couche de features pour VGG16

Backbone

Rôle des couches de convolution

- Calculer un (grand) ensemble de features à partir de l'image
- Suivies de couches denses (MLP) : la tête du réseau

Modularité

- On peut remplacer les premières couches de convolution par d'autres
- Usage des couches apprises sur d'autres réseaux
  - de classification : AlexNet, GoogLeNet, ResNet50,…
  - Ou pour d'autres tâches: auto-encodeurs (débruitage), Unet (segmentation)
- Ces couches peuvent incorporer des modules (e.g. inception pour GoogLeNet)

Backbone : réseau dont on prend les premières couches

Détection

Deux grands problèmes

- Nombre d’objets inconnu a priori
- Encodage de la bounding box

Problèmes corollaires

- Gestion des détections multiples
- À fois classification et régression : loss ?
- Comment évaluer un résultat de détection ?

Analyse du problème

Tête = MLP

- Sortie = tenseur de taille fixe
- Mais résultat de taille variable
- Notion clé : la confiance → paramètre à estimer
  - Associée à chaque détection
  - Détections en très nombreux sites a priori, mais filtrées par le niveau de confiance (NMS=Non Maxima Suppression)

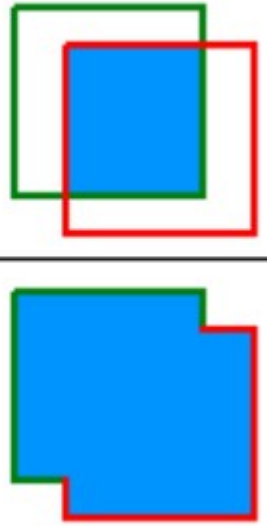
Que veut dire détecter ?

- Quand est-ce qu'un objet est détecté ?
- Bounding boxes pas exactes par rapport à vérité terrain
- Usage de l'IoU pour évaluer une détection
- Usage de la mAP comme métrique

IoU ET MAP [Metrics2020]

IoU : Intersection over Union

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\quad}{\quad}$$



Aussi appelé index de Jaccard

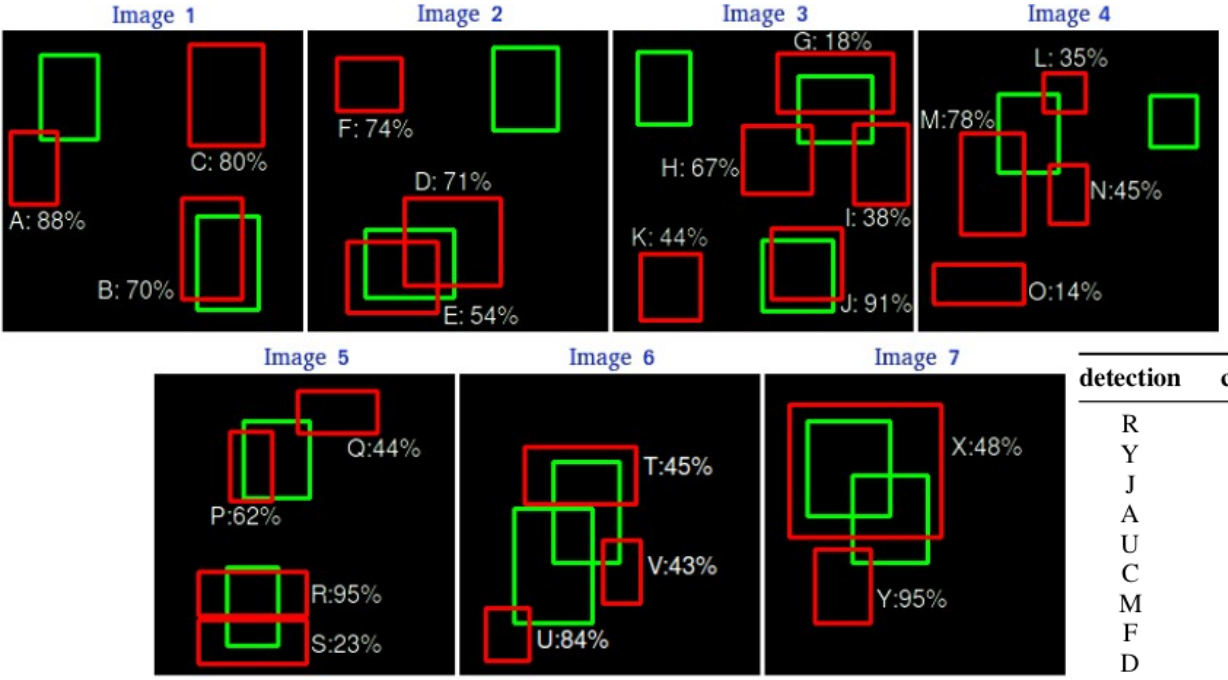
Usage d'un seuil t : IoU > t => VP, sinon FP

Mais : quel seuil sur la confiance pour les FP et TP ?

mAP

Exploitation conjointe de l'IoU et de la confiance

MAP [Metrics2020]



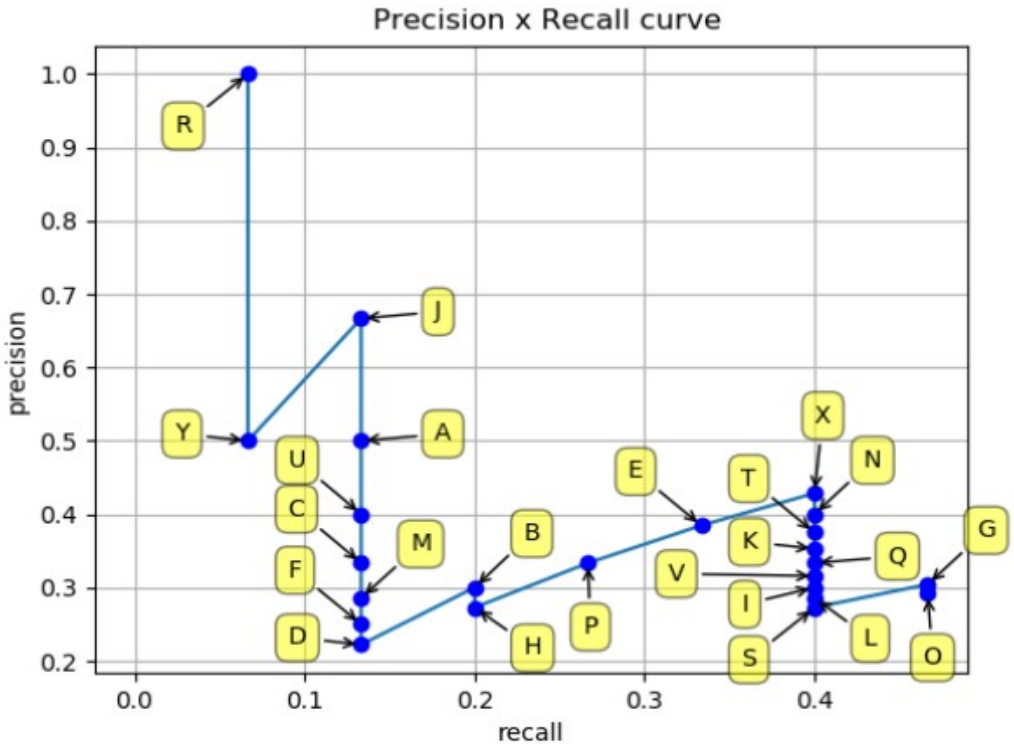
1 classe, 15 objets GT, 24 détections
IoU > 30 %

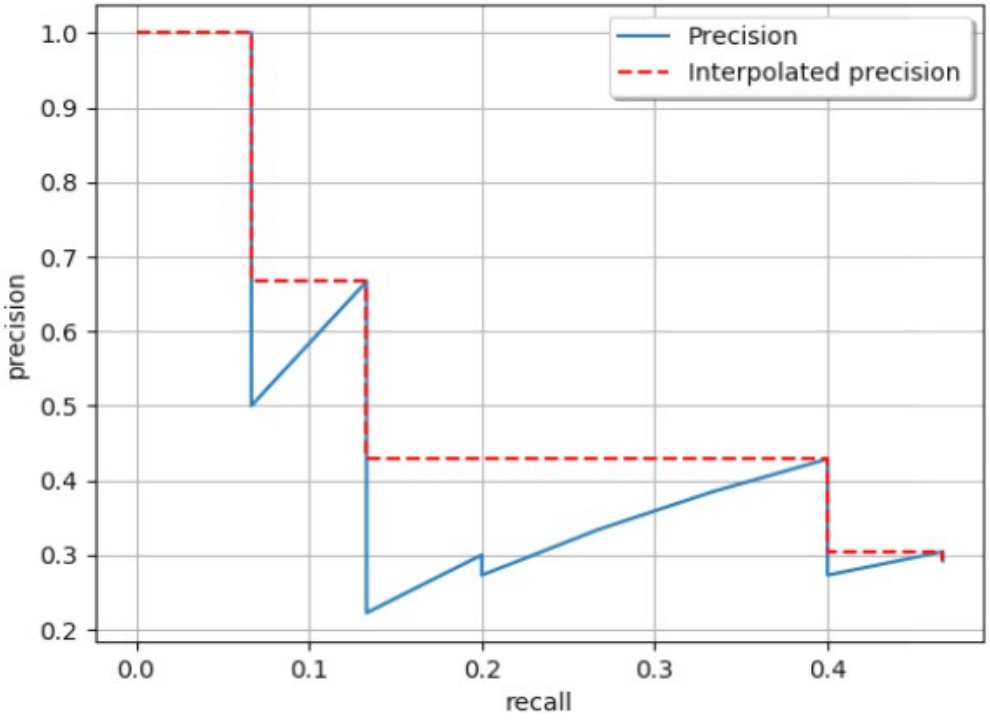
$$P = \frac{VP}{VP + FP} = \frac{VP}{\#\ \text{détections}}$$

$$R = \frac{VP}{VP + FN} = \frac{VP}{\#\ \text{objets GT}}$$

detection	confidence	TP	FP	acc TP	acc FP	precision	recall
R	95%	1	0	1	0	1	0.0666
Y	95%	0	1	1	1	0.5	0.0666
J	91%	1	0	2	1	0.6666	0.1333
A	88%	0	1	2	2	0.5	0.1333
U	84%	0	1	2	3	0.4	0.1333
C	80%	0	1	2	4	0.3333	0.1333
M	78%	0	1	2	5	0.2857	0.1333
F	74%	0	1	2	6	0.25	0.1333
D	71%	0	1	2	7	0.2222	0.1333
B	70%	1	0	3	7	0.3	0.2
H	67%	0	1	3	8	0.2727	0.2
P	62%	1	0	4	8	0.3333	0.2666
E	54%	1	0	5	8	0.3846	0.3333
X	48%	1	0	6	8	0.4285	0.4
N	45%	0	1	6	9	0.4	0.4
T	45%	0	1	6	10	0.375	0.4
K	44%	0	1	6	11	0.3529	0.4
Q	44%	0	1	6	12	0.3333	0.4
V	43%	0	1	6	13	0.3157	0.4
I	38%	0	1	6	14	0.3	0.4
L	35%	0	1	6	15	0.2857	0.4
S	23%	0	1	6	16	0.2727	0.4
G	18%	1	0	7	16	0.3043	0.4666
O	14%	0	1	7	17	0.2916	0.4666

MAP [Metrics2020]





AP (Average Precision)=aire sous la courbe rouge en pointillés

mAP (mean AP)=moyenne des AP sur toutes les classes

Suppression des non maxima (NMS) [YOLOReview2023]

Algorithm 1 Non-Maximum Suppression Algorithm

Require: Set of predicted bounding boxes B, confidence scores S, IoU threshold τ, confidence threshold T
Ensure: Set of filtered bounding boxes F

$F \leftarrow \emptyset$
Filter the boxes: $B \leftarrow \{b \in B \mid S(b) \geq T\}$
Sort the boxes B by their confidence scores in descending order
while $B \neq \emptyset$ **do**
    Select the box b with the highest confidence score
    Add b to the set of final boxes F: $F \leftarrow F \cup \{b\}$
    Remove b from the set of boxes B: $B \leftarrow B - \{b\}$
    for all remaining boxes r in B **do**
        Calculate the IoU between b and r: $iou \leftarrow IoU(b, r)$
        if $iou \geq \tau$ **then**
            Remove r from the set of boxes B: $B \leftarrow B - \{r\}$
        end if
    end for
end while

Bounding Boxes : régression de masques [Szegedy2013]

Base

- Backbone=AlexNet
- Modification de la sortie : une image binaire (24x24)=masque

Problèmes

- Résolution grossière pour le masque
  - Encodage de la GT par pixel partiel
  - Plusieurs échelles de résolution
  - Répétition (détection, crop) pour affiner la taille de la boîte
- Objets qui se touchent ne peuvent être discriminés → prédiction des moitiés haute, basse, droite et gauche du masque (5 masques prédits)
- Nombreuses détections
  - Seuil sur score de superposition à la GT
  - Seuil de classification (AlexNet)
  - Suppression des non maxima locaux (scores de classification et de superposition)

[Szegedy2013]

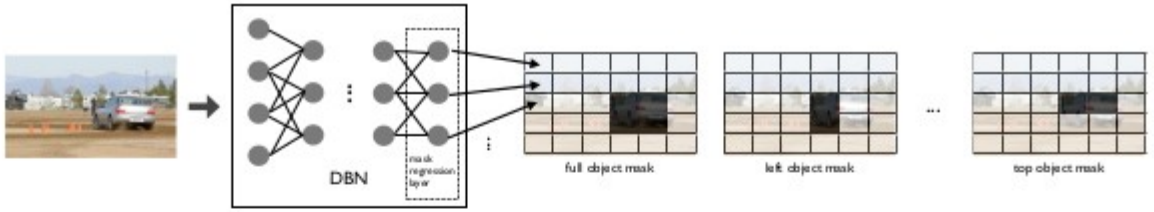


Figure 1: A schematic view of object detection as DNN-based regression.

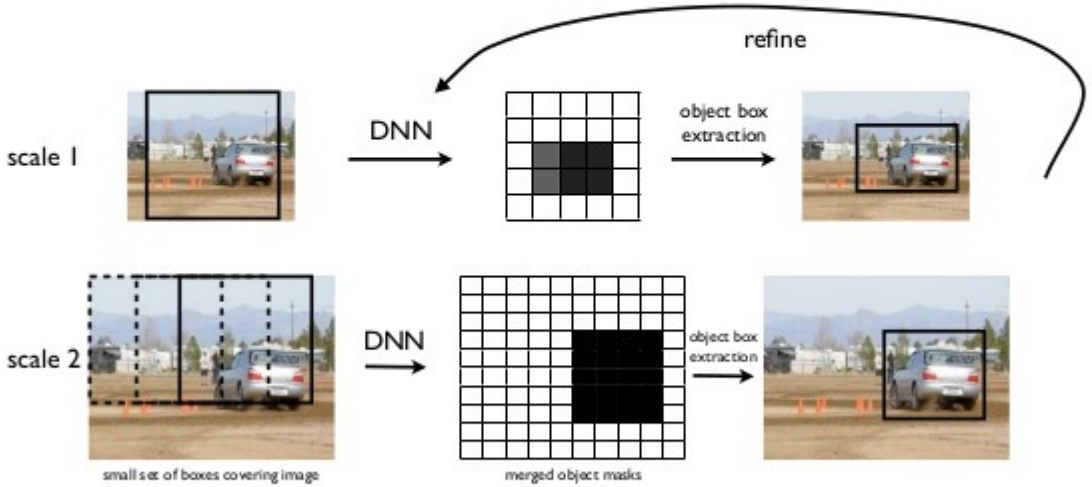


Figure 2: After regressing to object masks across several scales and large image boxes, we perform object box extraction. The obtained boxes are refined by repeating the same procedure on the sub images, cropped via the current object boxes. For brevity, we display only the full object mask, however, we use all five object masks.

FENETRES GLISSANTES [OverFeat2014]

Idée de base

- Fenêtre glissante → crop → classification → seuil
- Fenêtre varie selon : position, taille, échelle…

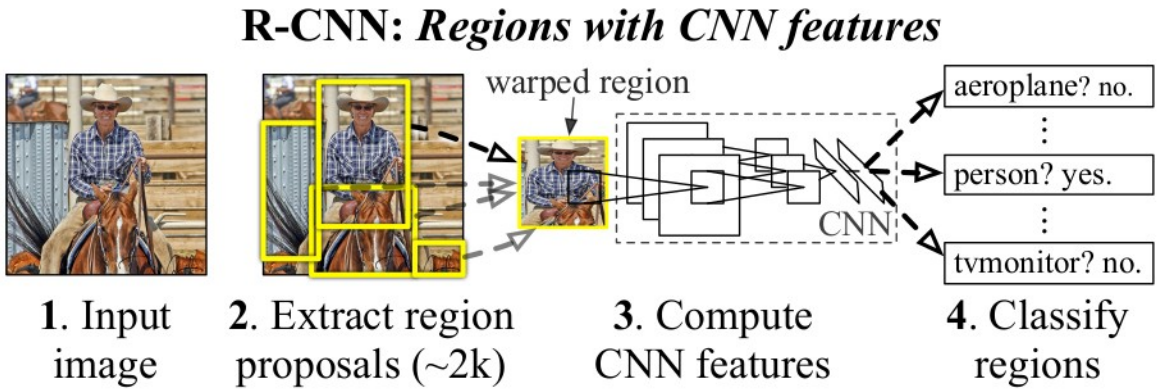
Idée originale

- Convolution → invariance à translation, échelle facile
- Beaucoup de calculs en commun → partage des premières couches
- Têtes : classification+confiance, et régression des 4 coins (x_{min},y_{min},x_{max},y_{max})
- Cf Fig. 6, p. 9

Critique

- Scores perfectibles, détection d'un seul objet en pratique (comme Szegedy)

PROPOSITION DE RÉGIONS [RCNN2014]



5 étapes

- Extraire des régions candidates
- Appliquer un CNN sur chaque région → on ne garde que les features
- Appliquer un classifieur sur les features
- Ne garder que celles avec un score suffisant + NMS
- Ajustement des boîtes englobantes candidates par régression

Briques

- Régions candidates : selective search (segmentation puis regroupement hiérarchique)
- Features : AlexNet
- Classifieur : SVM
- NMS basée sur IoU (seuil appris)
- Utilisation de la couche 5 du réseau pour apprendre les poids de la régression des boîtes

VARIANTE [FastRCNN2015]

Ce qui est gardé

Toujours des régions candidates

Ce qui est nouveau

VGG16 comme feature extractor, pré-entraînée sur ImageNet

ROI pool : crop des features

Prédiction classe/bbox (affinée)xK

Multi-task loss (u=classe (K+1), P=proposed, G=Ground Truth)

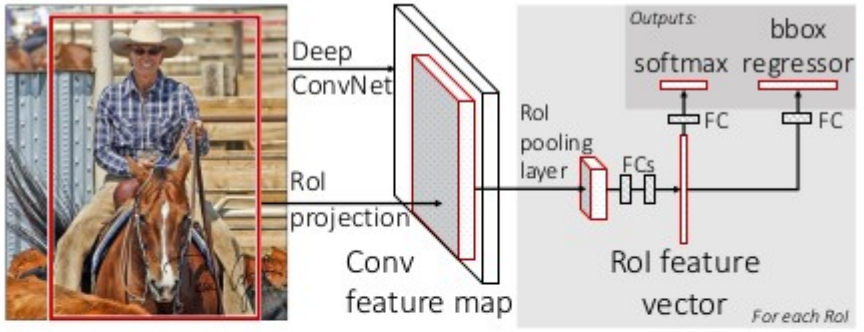


Figure 1. Fast R-CNN architecture. An input image and multiple regions of interest (RoIs) are input into a fully convolutional network. Each RoI is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected layers (FCs). The network has two output vectors per RoI: softmax probabilities and per-class bounding-box regression offsets. The architecture is trained end-to-end with a multi-task loss.

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1] L_{\text{loc}}(t^u, v)$$

$$L_{\text{cls}}(p, u) = -\log p_u$$

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t_i^u - v_i) \quad \text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

$$t_x = (G_x - P_x)/P_w$$
$$t_y = (G_y - P_y)/P_h$$
$$t_w = \log(G_w/P_w)$$
$$t_h = \log(G_h/P_h).$$

VARIANTE [FasterRCNN2015]

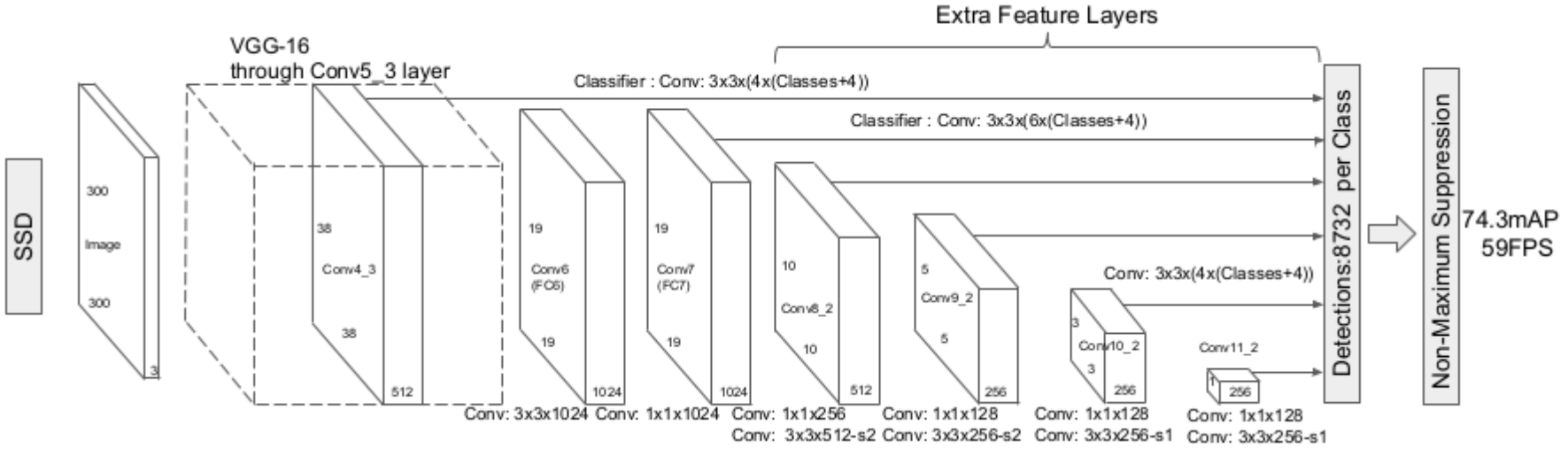
RPN : region proposal network

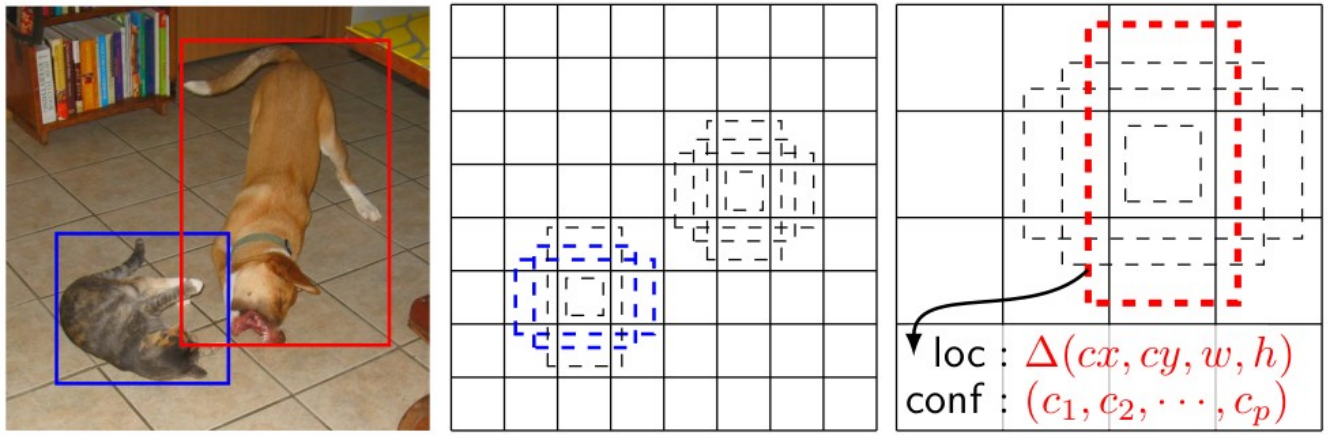
- Partage de poids avec CNN de FastRCNN
- Fenêtre glissante sur la dernière couche de convolution (3x3)
- Notion d'ancres : k=9 formes prédéfinies de fenêtres (3 échelles et 3 rapports d'aspect)
- Prédiction :
  - 1 vecteur de probabilités de classe pour chaque ancre
  - 4 coordonnées (Bbox) relatives à chaque ancre : facteurs correctifs, amplitude plus homogène et restreinte

Critique de ces approches

- Dépendantes de la proposition de régions → BB trop nombreuses et/ou inadaptées
- Compromis formes de BB vs temps de calcul → approches lentes et coûteuses

SINGLE-SHOT DETECTOR [SSD2016]





(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

SINGLE-SHOT DETECTOR [SSD2016]

Loss

Matching (x) via index de Jaccard (IoU) > 0.5

α estimé à 1 par validation croisée

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^{k} \text{smooth}_{\text{L1}}(l_i^m - \hat{g}_j^m)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

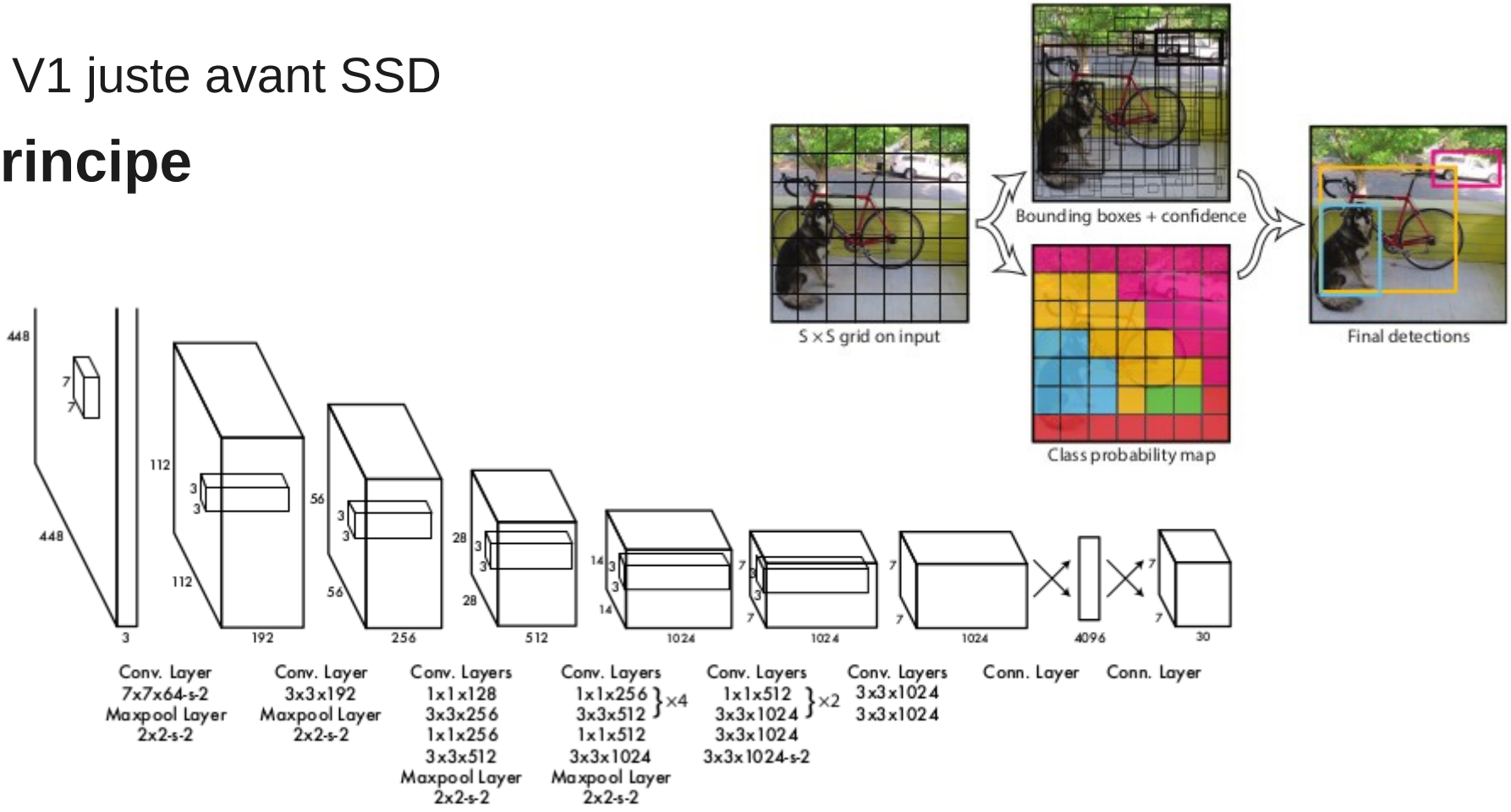
$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right)$$

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^p log(\hat{c}_i^p) - \sum_{i \in Neg} log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

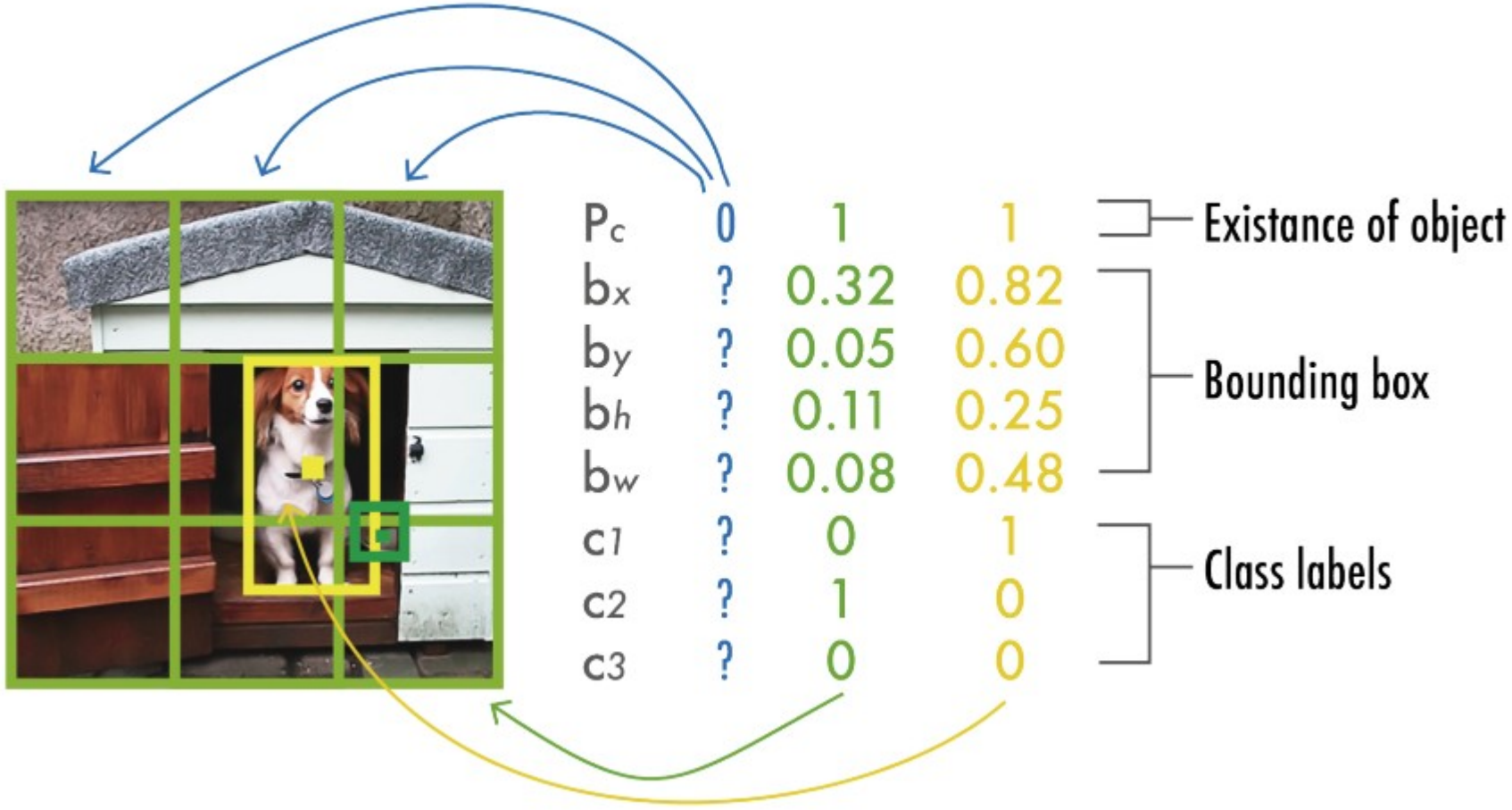
YOU ONLY LOOK ONCE [YOLO2016]

V1 juste avant SSD

Principe



YOU ONLY LOOK ONCE [YOLO2016]



YOLOv1 [YOLO2016]

Prédiction (par cellule)

B(=2) Bbox (=x,y,w,h,c) + C(=20) classes

Loss

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

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

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

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

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

Voir aussi [YOLOReview2023] pour une explication

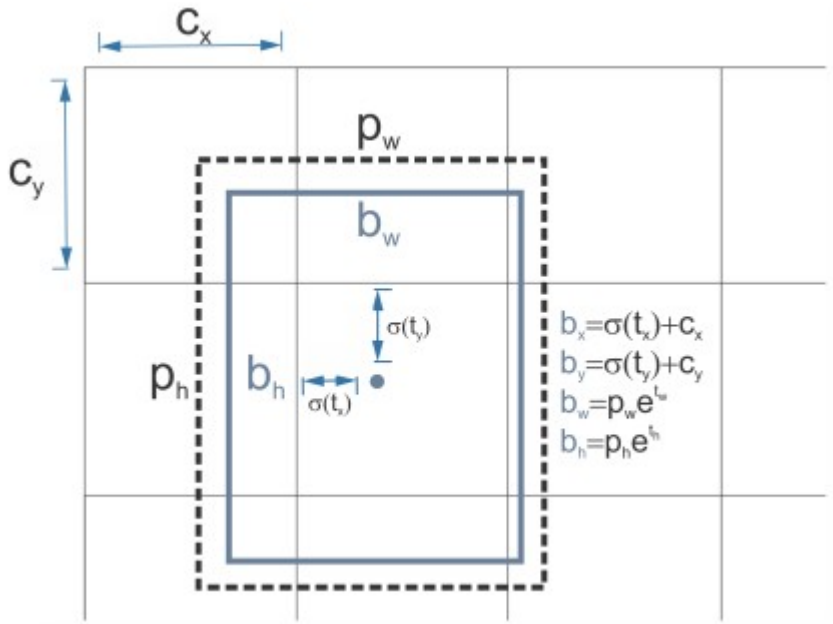
YOLOV2 [YOLOReview2023]

Nouveautés

Suppression de la couche FD → fully convolutional → multiéchelles

Usage d'ancres (classication k-means sur base d'apprentissage)

Nouvelle paramétrisation des BB



YOLOV3 [YOLOReview2023]

Nouveautés

- Objectness score (au lieu de la confiance) → régression logistique (Binary Cross Entropy - BCE)
- BCE indépendantes pour les probabilités de classes (pas de normalisation SoftMax)
- Différents backbones testés (Feature extractors)

YOVOV+... [YOLOReview2023]

De YOLOv4...

Intégration de modules plus évolués (attention, pyramid networks…) → Bag-of-Specials

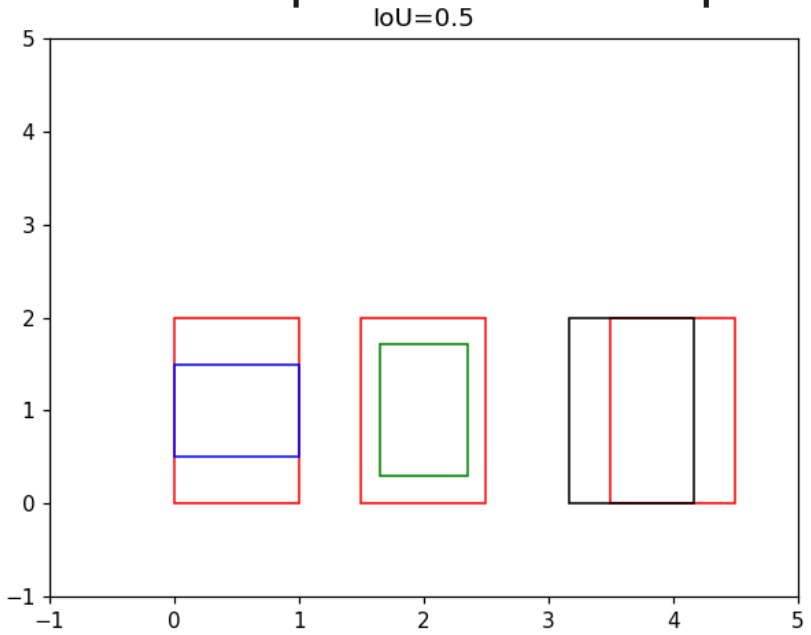
Bag-of-Freebies → e.g. lissage de classes, CIoU

Disparition des ancres…

À YOLOv12 !

IoU : CRITIQUES

- **Problèmes de IoU**
  - Pas de gradient pour les BB qui ne se superposent pas (fonction constante=0)
  - Pas de notion de distance : plusieurs cas équivalents



CIoU : Complete IoU

Trois critères

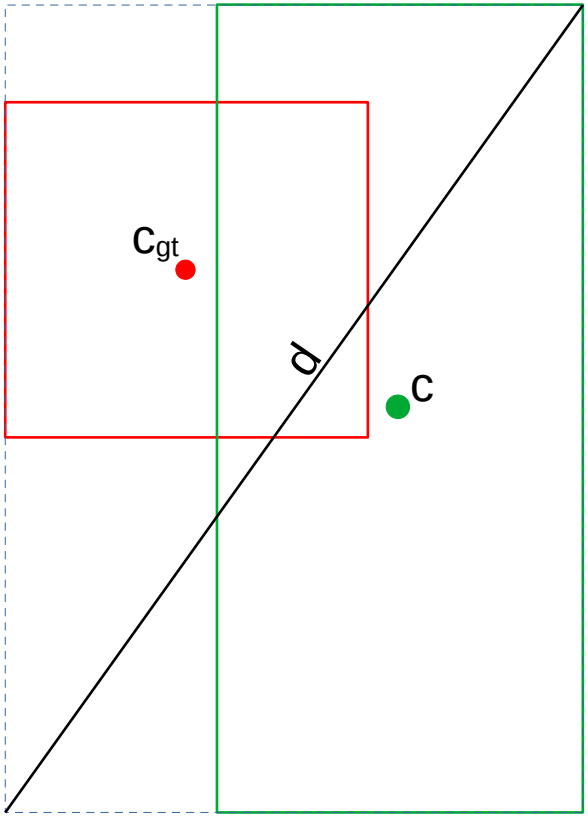
- Grande superposition
- Centres proches
- Aspect similaire (rapport de forme)

Objectif

- Minimisation

Formule : somme de 3 termes

- Grande superposition : 1- IoU
- Centres proches : $|c\text{-}c_{gt}|^2/d^2$
- Aspect similaire : α $(\text{atan}(w_{gt}/h_{gt}) - \text{atan}(w/h))$ $4/\pi^2$



RÉFÉRENCES

[Szedegy2013] C. Szegedy, A. Toshev, and D. Erhan « Deep Neural Networks for Object Detection ». Advances in Neural Information Processing Systems (NIPS'13), vol 26, 2013.

[RCNN2014] R. Girshick, J. Donahue, T. Darrell, and J. Malik « Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation », IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14) pp. 580-587, 2014.

[OverFeat2014] P. Sermanet, D. Eigen, X. Zhiang, M. Mathieu, R. Fergus, and Y. LeCun « OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks » International Conference on Learning Representations (ICLR'14), 2014.

[FastRCNN2015] R. Girshick « Fast R-CNN », IEEE International Conference on Computer Vision (ICCV'15), pp. 1440-1448, 2015.

[FasterRCNN2015] S. Ren, K. He, R. Girshick, and J. Sun. « Faster R-CNN : towards real-time object detection with region proposal networks ». International Conference on Neural Information Processing Systems (NIPS'15), vol 1, pages 91-99, 2015.

[SSD2016] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A.C. Berger « SSD: Single Shot Multibox Detector ». European Conference on Computer Vision (ECCV'16), LNCS vol 9905, 2016.

[YOLO2016] J. Redmon, S. Divvala, R. Girschick, A. Farhadi « You Only Look Once : Unified, Real-Time Object Detection » IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16), 2016.

[YOLOReview2023] J. Terven, D.-M. Cordova-Esparza, and J.-A. Romero-Gonzalez « A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS » Machine Learning and Knowledge Extraction 5(4), 1680-1716, 2023.

[Metrics2020] R. Padilla, S.L. Netto, and E.A.B. Da Silva. « A survey on performance metrics for object-detection algorithms » International Conference on Systems, Signals and Image Processing (IWSSIP'20), pages 237–242, 2020.