Yolo

Détection d’objets

Plan

1. Introduction (YoloV1)
2. Priors (YoloV2)
3. Post-process
4. Entraînement
5. Evaluation
6. Bonus

Introduction

La détection c’est quoi?

&

L’architecture de YoloV1

Rappels sur la classification

Classifier

CNN + Dense

Vecteur de probas

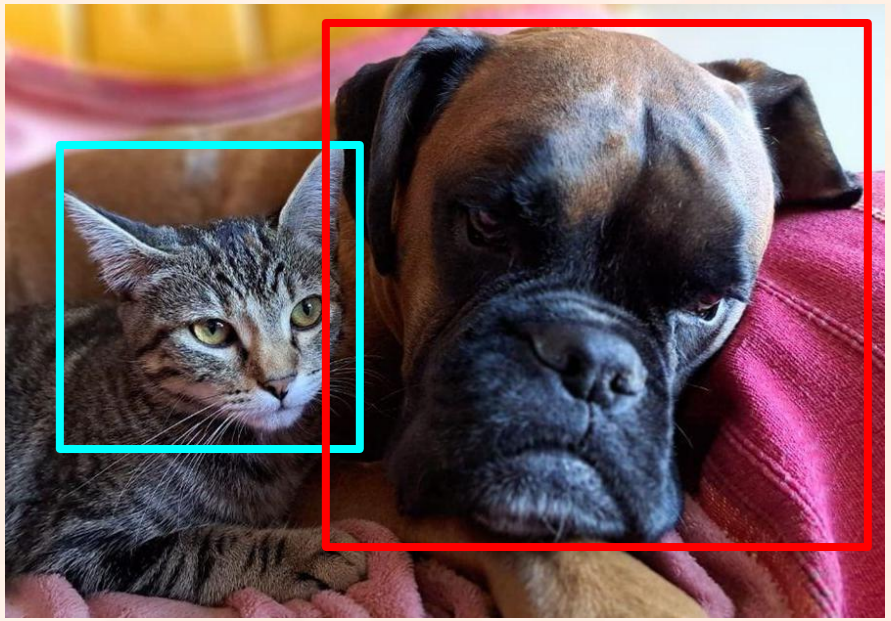
0.95	chat
0.03	chien
0.02	lapin

Sortie de taille fixe, peu importe l'image d'entrée

Formalisation de la détection

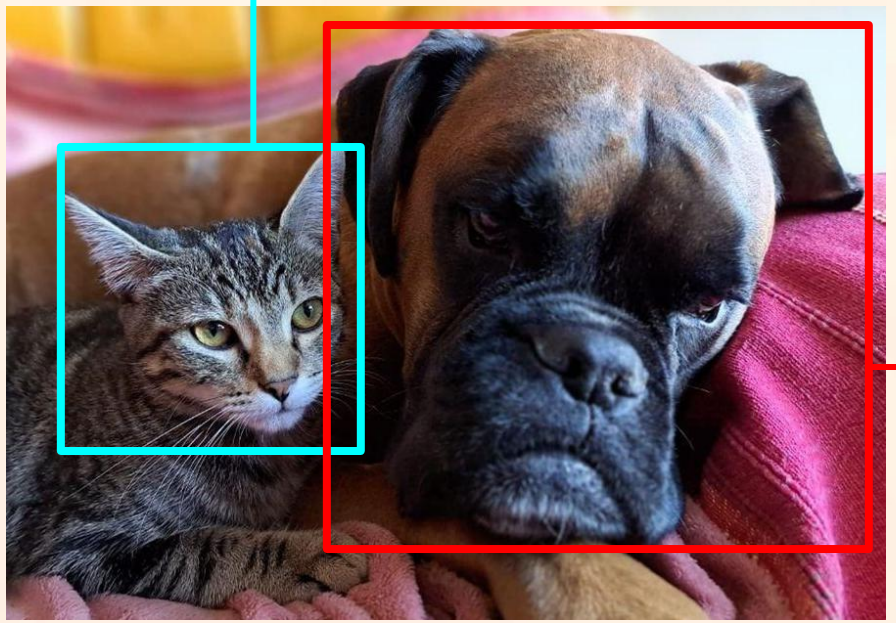
Détecteur

???



Objectif : trouver des « objets » en donnant leur taille, position et une classe (vecteur de probas)

Formalisation de la détection



Objet #0
- $x = 30$
- $y = 25$
- $w = 50$
- $h = 50$

0.95	chat
0.03	chien
0.02	lapin

Objet #1
- $x = 120$
- $y = 35$
- $w = 100$
- $h = 100$

0.04	chat
0.92	chien
0.04	lapin

Le nombre de sortie dépend du contenu de l'entrée

Feature extraction

Feature Extractor

Convolution
Pooling

Feature Maps

Tête de classif

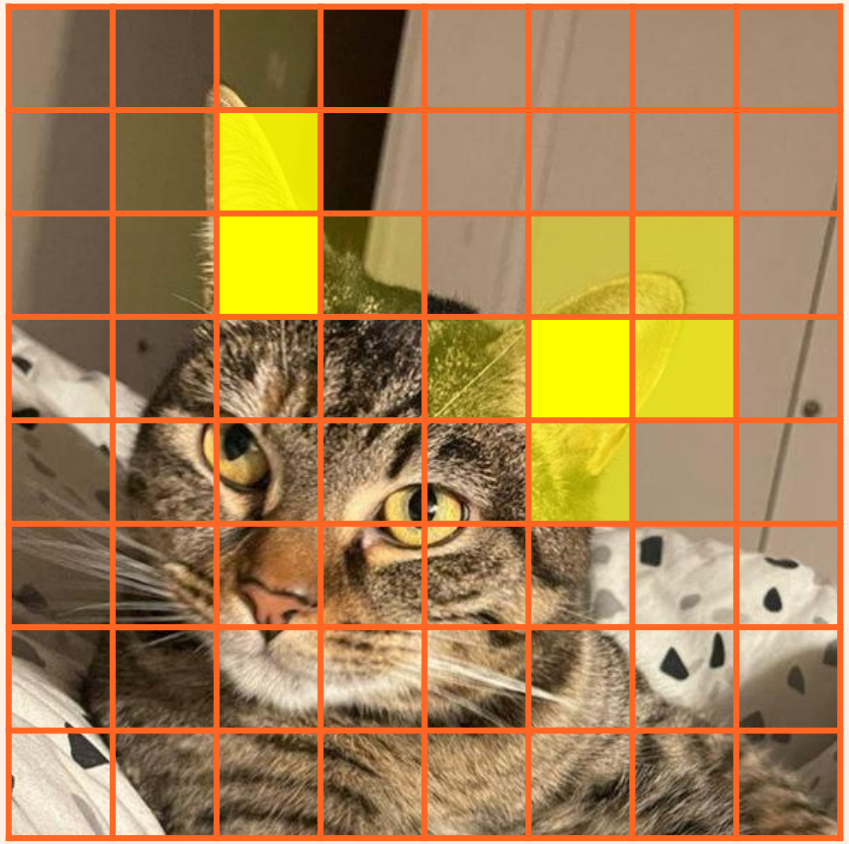
Flatten
Dense

Vecteur de probas

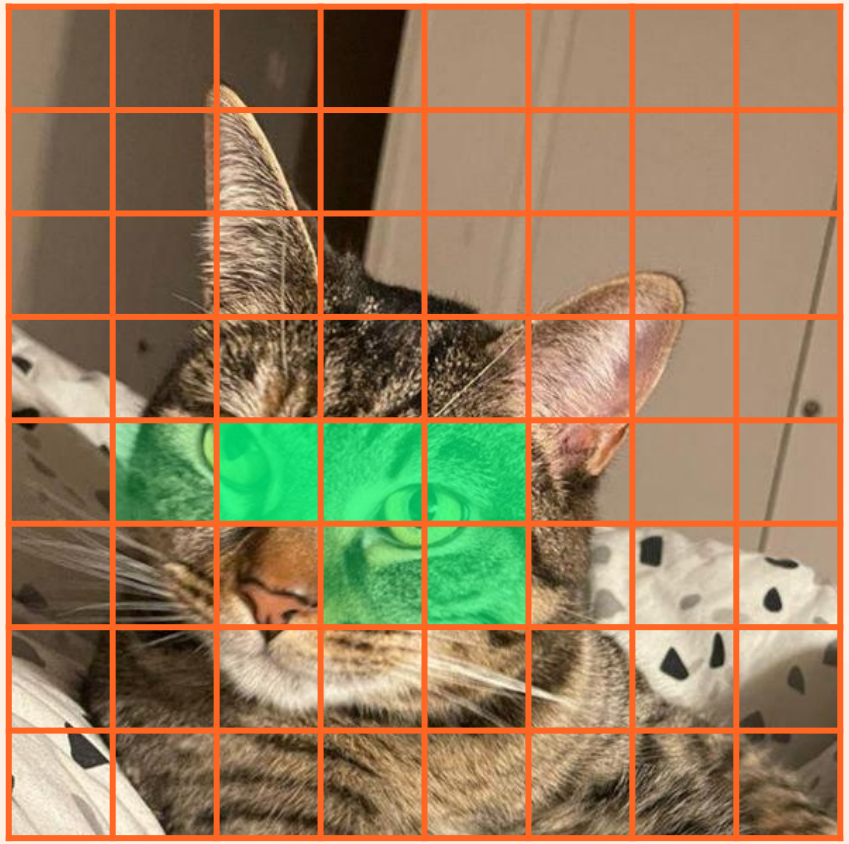
0.95	chat
0.03	chien
0.02	...

Interprétation de la feature map

Feature map des oreilles

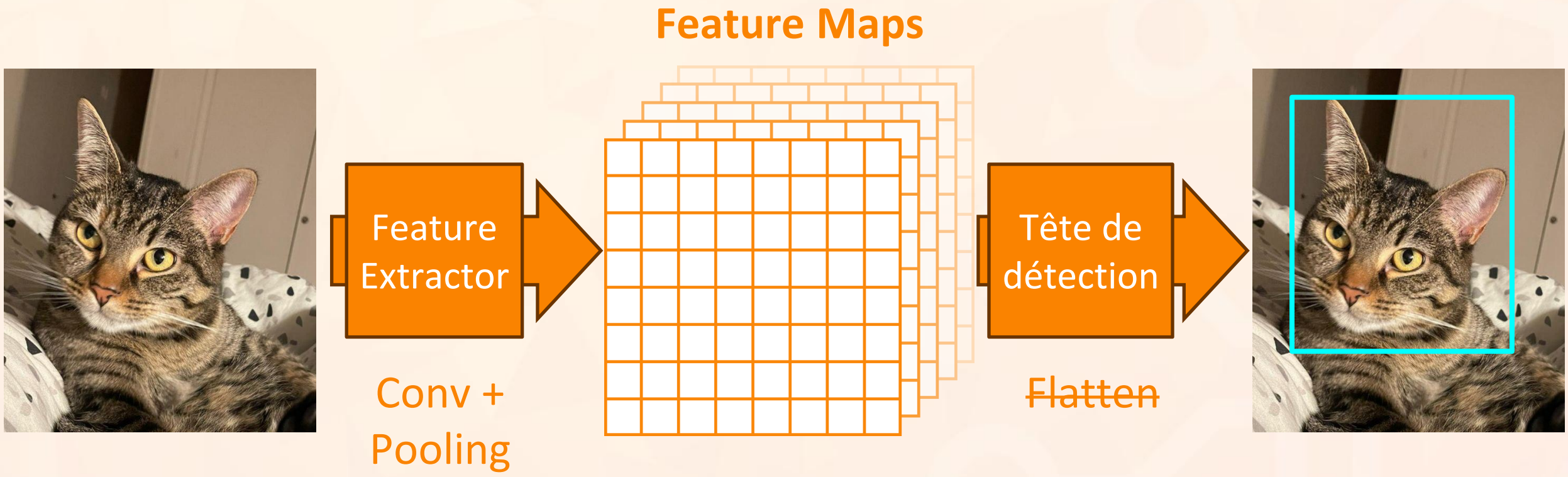


Feature map des yeux



Corrélation spatiale entre l'entrée et les feature maps

Et la détection dans tout ça?



Intuition : on veut garder l'**information spatiale**

L’architecture de base de Yolo

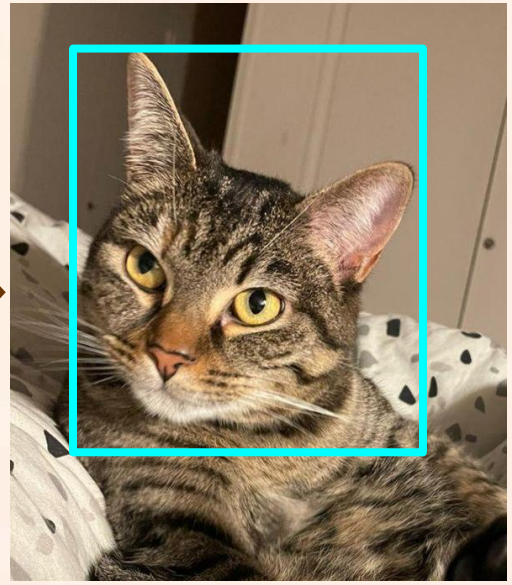
Feature Maps

Feature Extractor

Conv + Pooling

Conv 1x1

$5 + C$ filtres

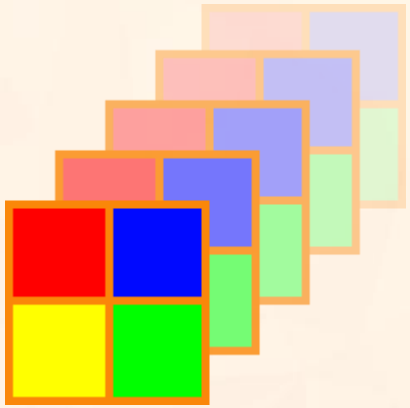


(avec un peu de post-processing)

Intuition : on veut garder l’**information spatiale**

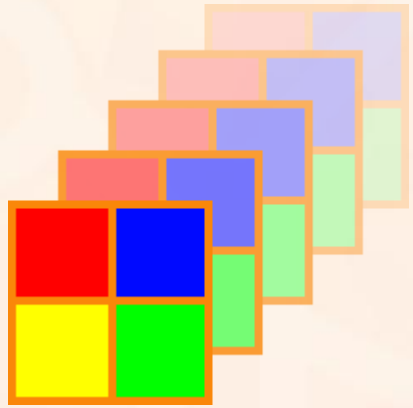
Interprétation d'une convolution 1x1

Feature Maps



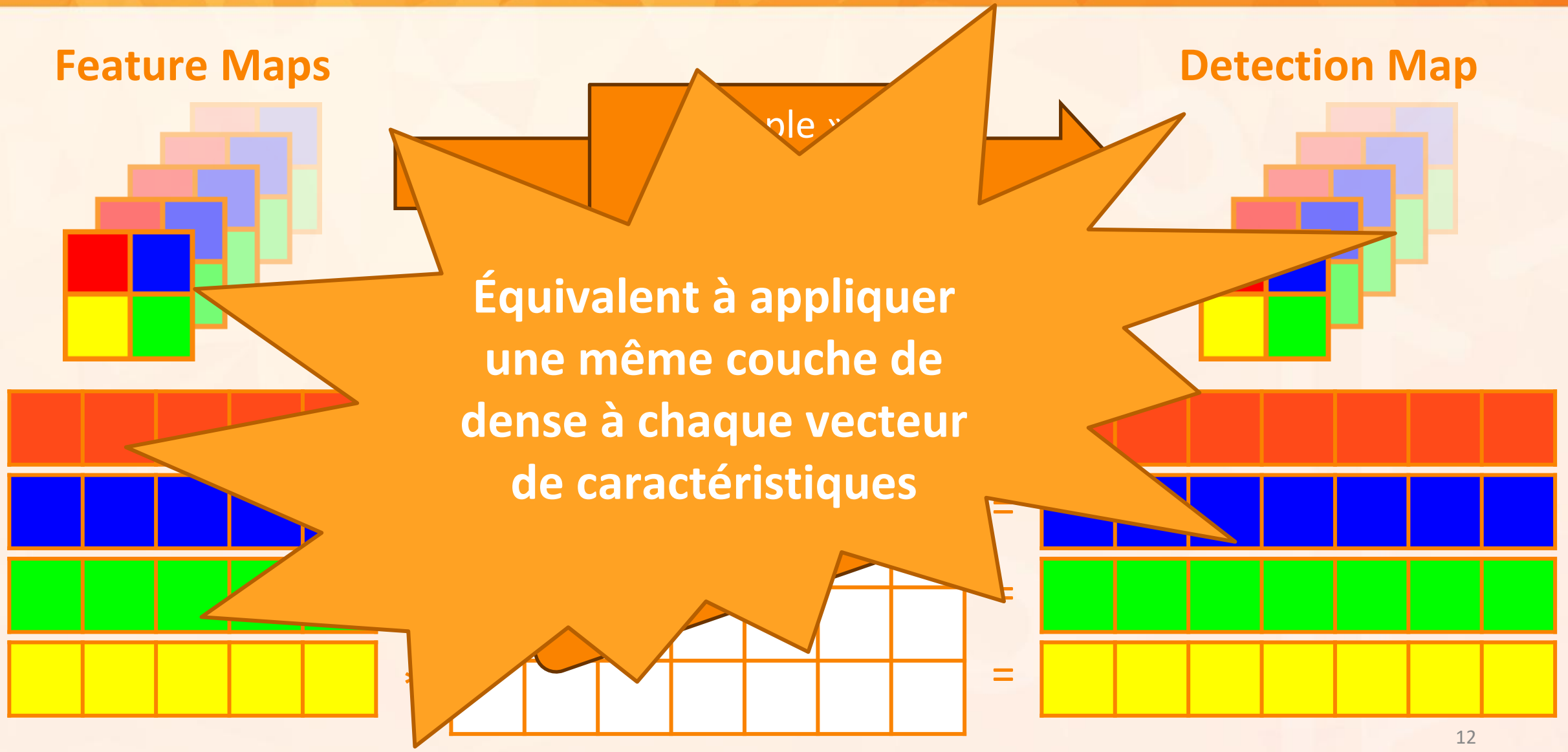
« simple »
Convolution
1x1

Detection Map

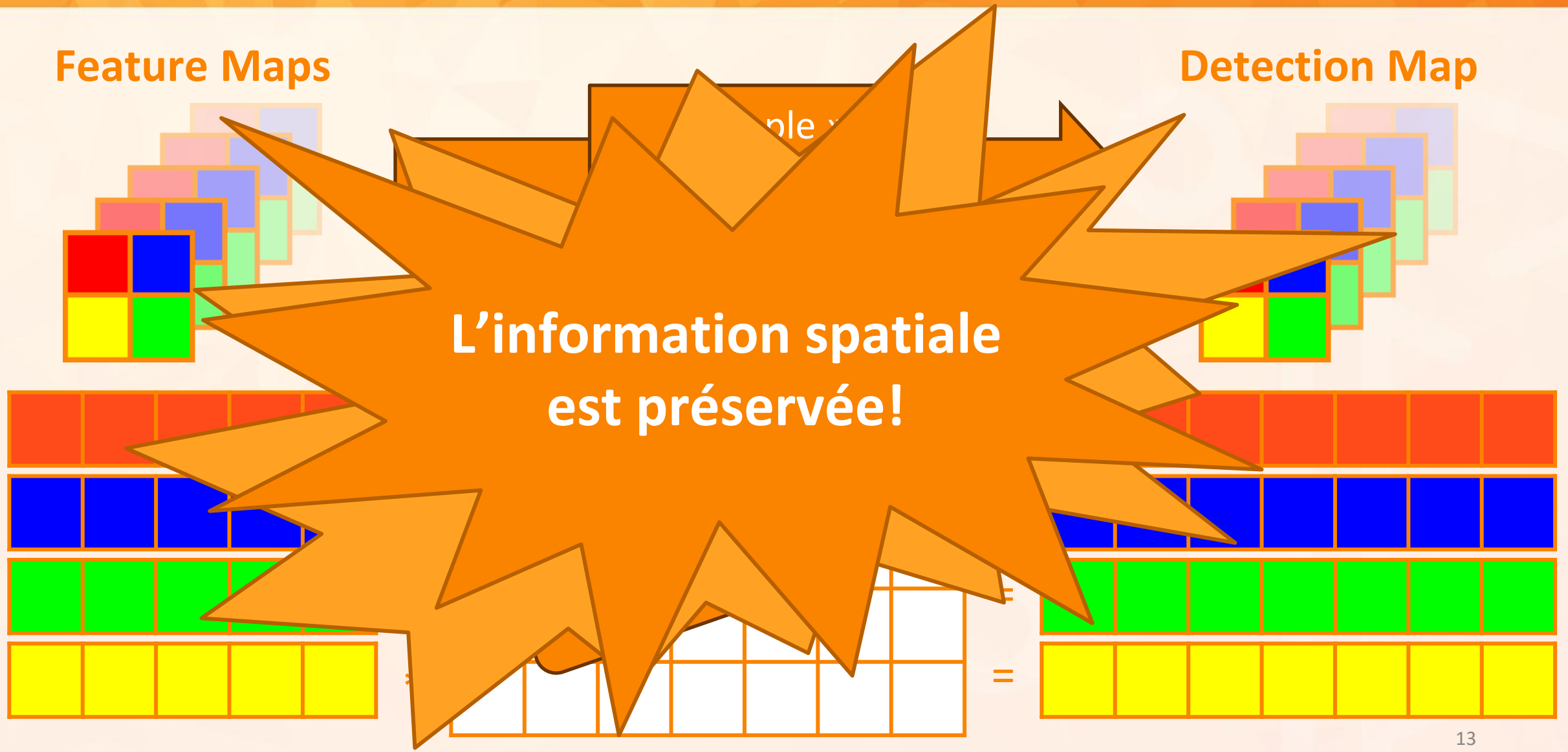


Nombre de filtres

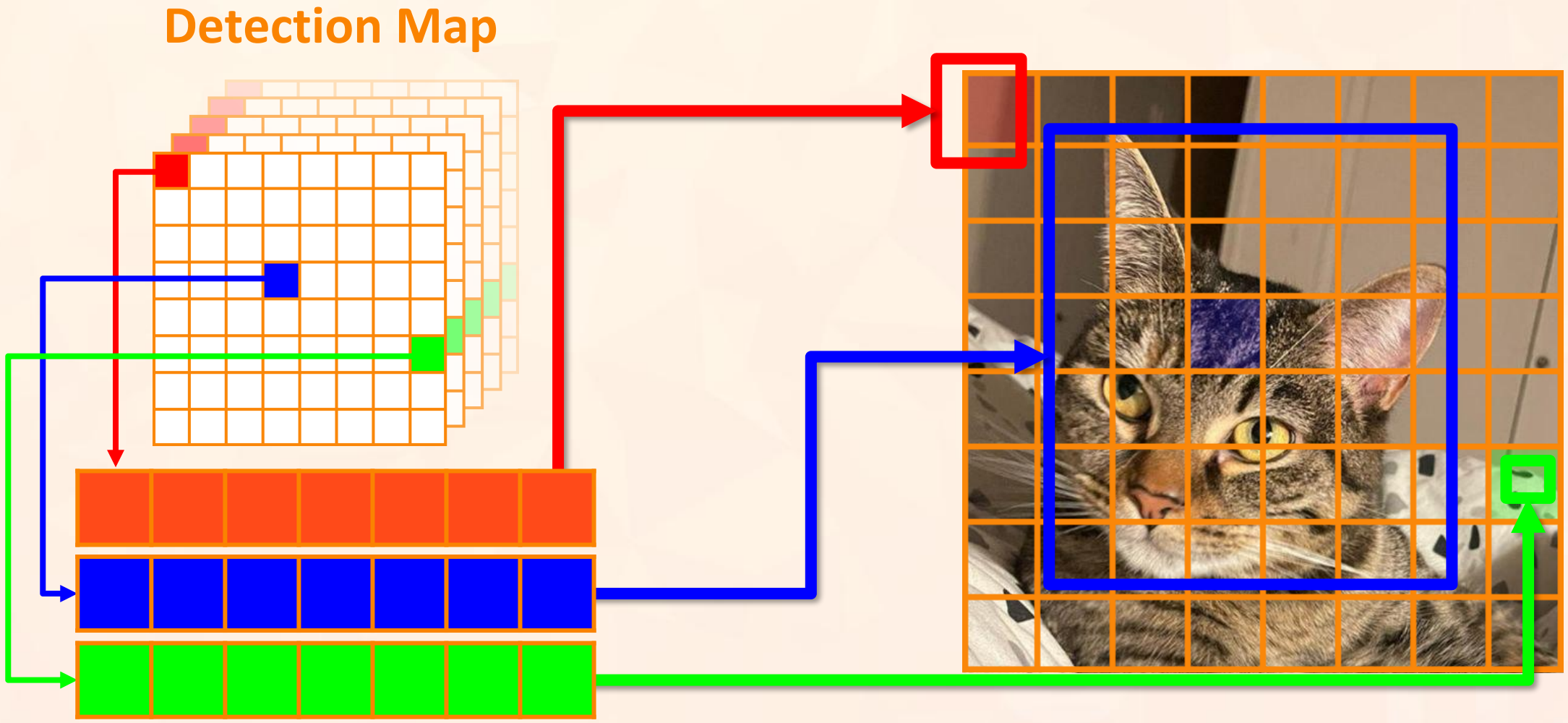
Interprétation d'une convolution 1x1



Interprétation d'une convolution 1x1

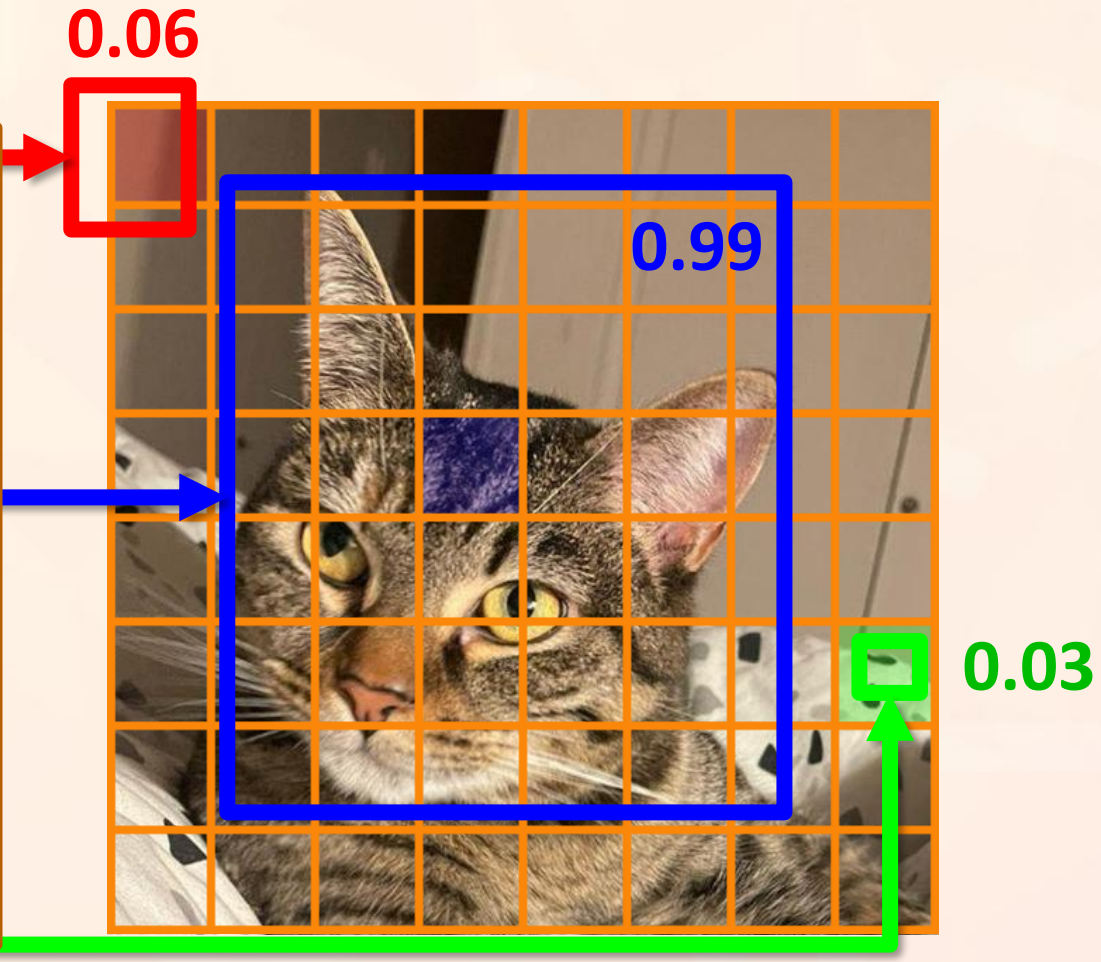


Interprétation de la detection map



Interprétation de la detection map

Detection Map

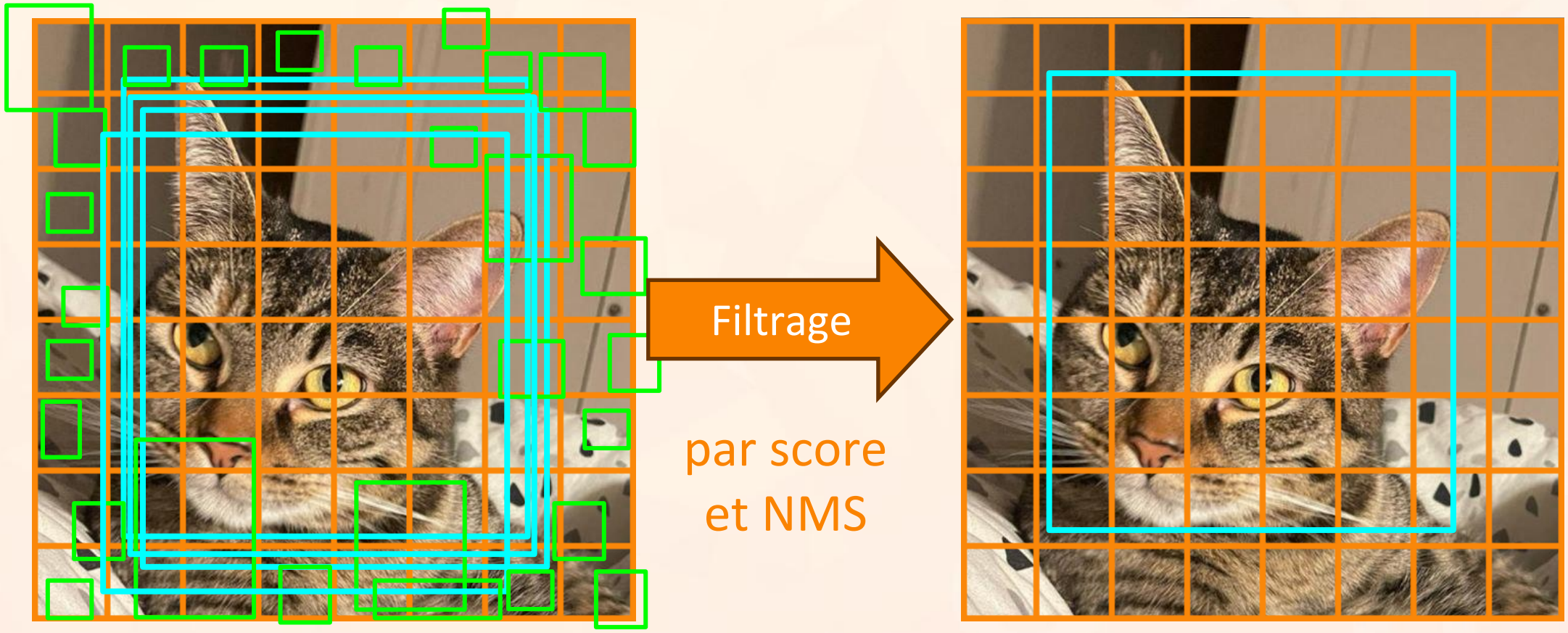


Remarques:

- En plus des coordonées et taille de la boîte, on prédit également un score pour la boîte de la cellule
- Niveau dimensions, les boîtes ne sont pas confinées à leur cellule, seulement les coordonées de leur centre le sont.

1 vecteur = 1 boîte

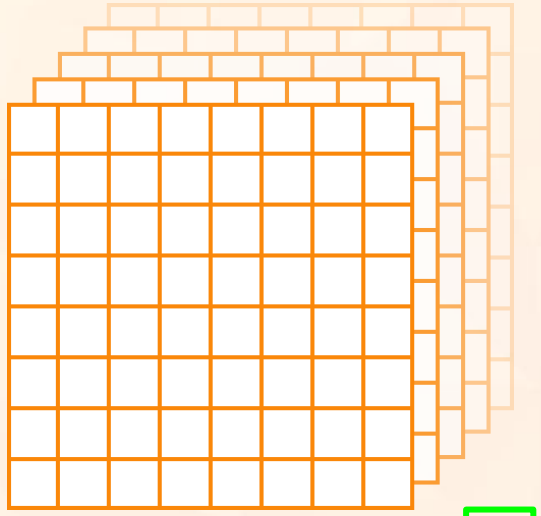
Post-process final



En résumé

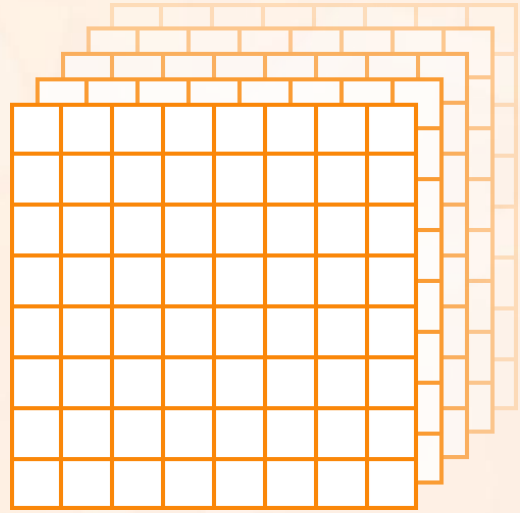
Feature Extractor

Feature Maps

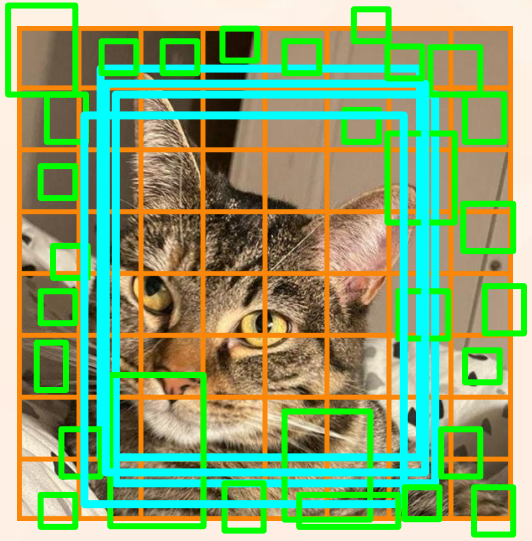


Conv 1x1

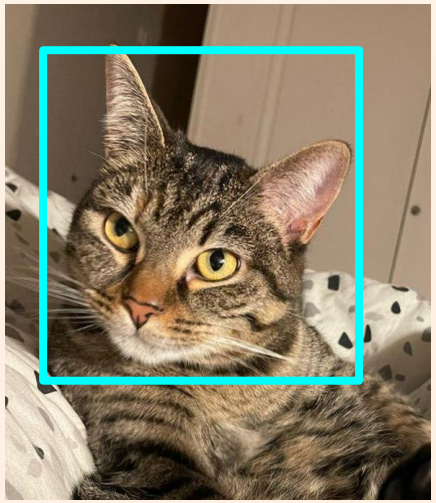
Detection Map



Décodage



Filtrage & NMS



En résumé

Modèle Yolo

Feature Extractor

Feature Maps

Conv 1x1

Detection Map

Sortie de taille fixe!

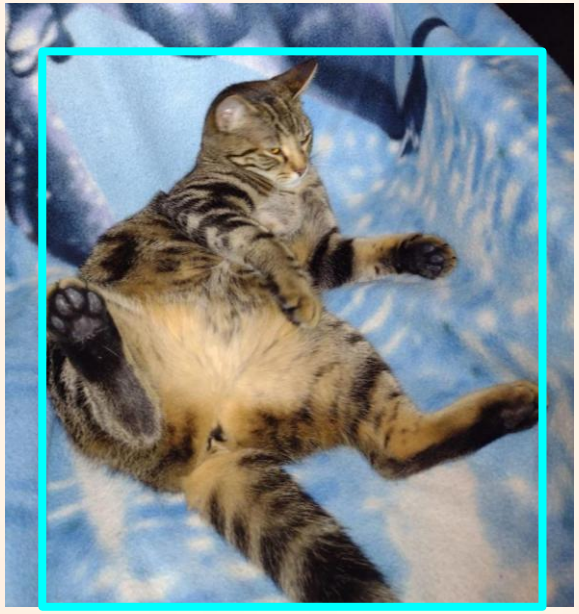
Décodage

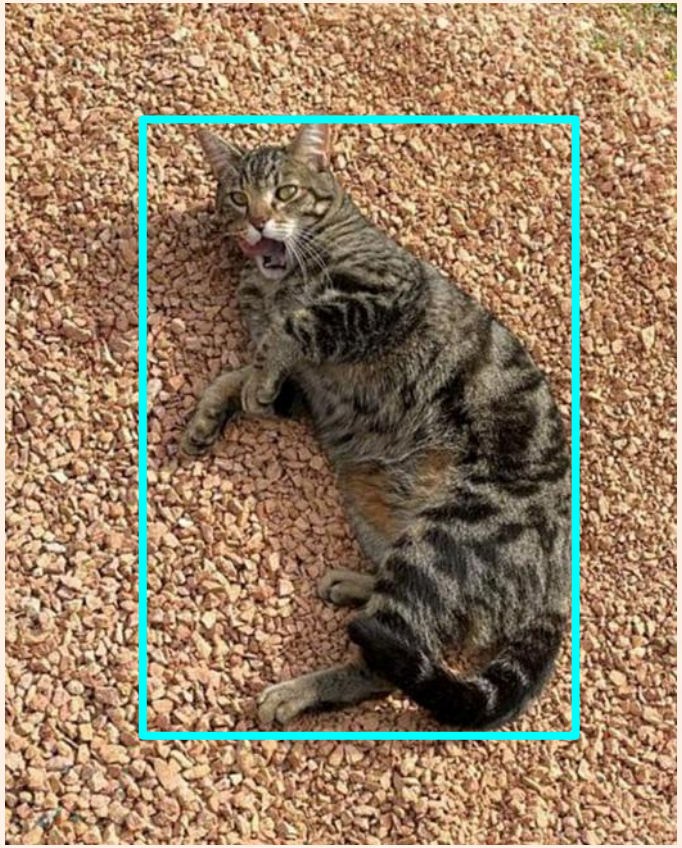
Questions?

Les priors

De YoloV1 a YoloV2

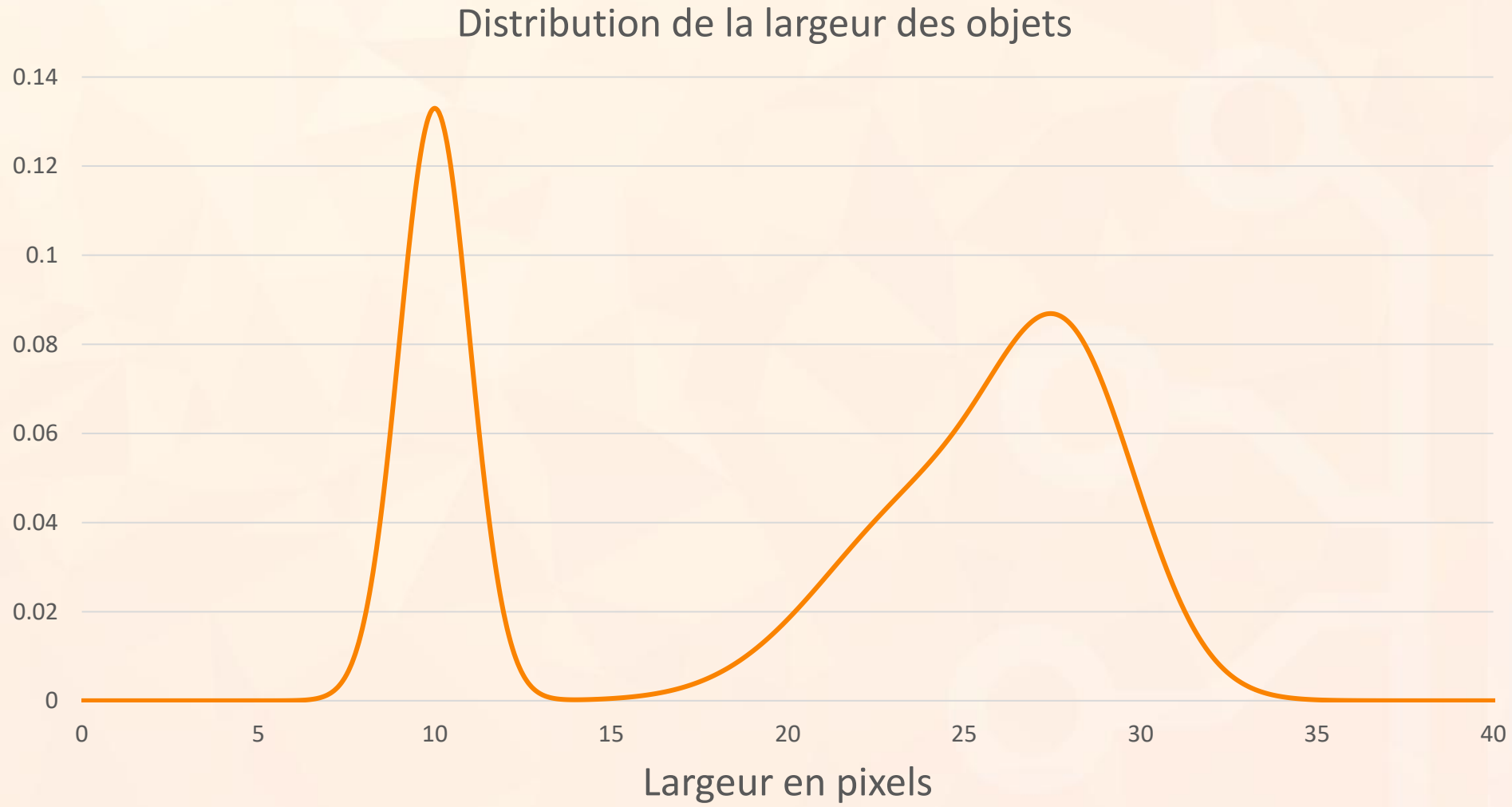
Le problème de YoloV1





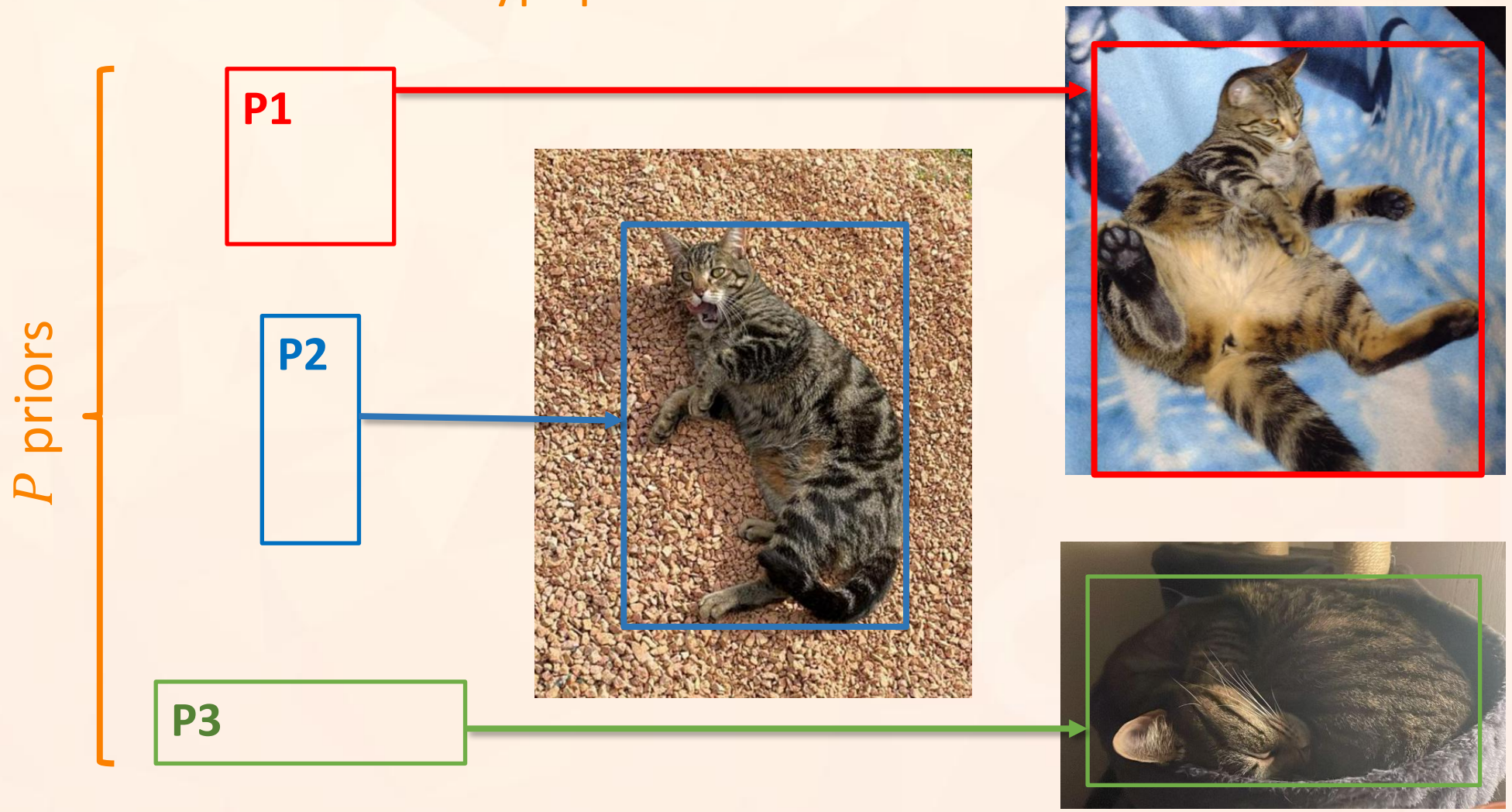
Les « objets » ont des formes et tailles variées

Un problème de distribution

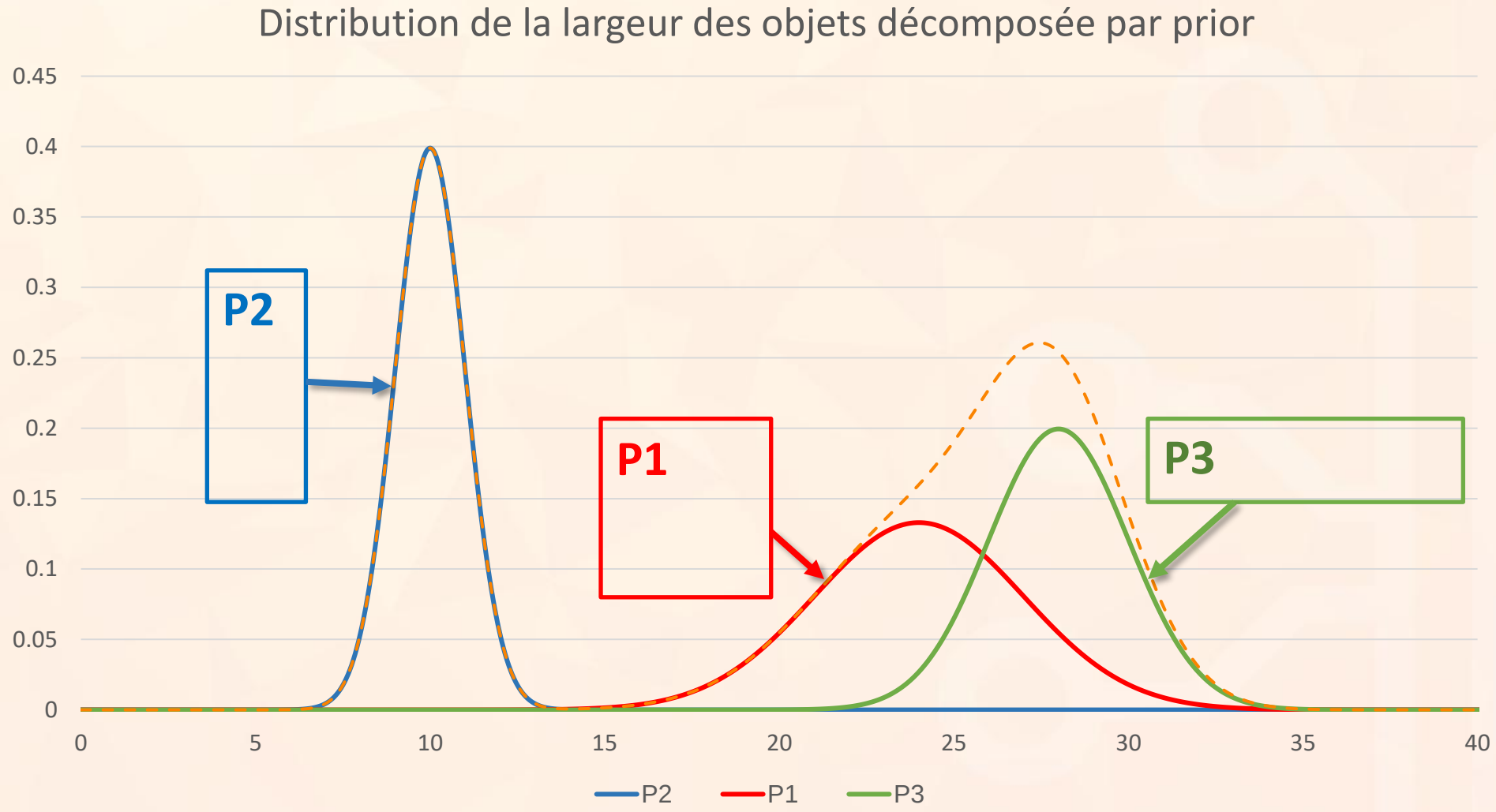


Les priors, c’est quoi?

Les priors sont des tailles typiques de boîtes

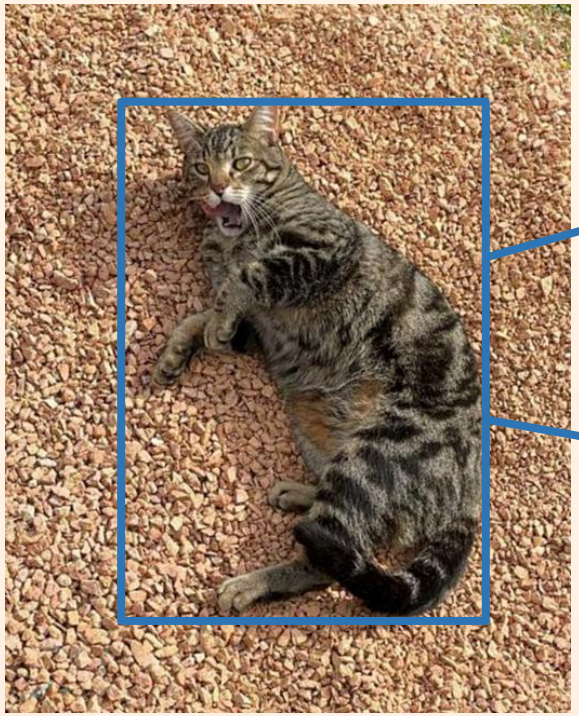


Une distribution plus sympa



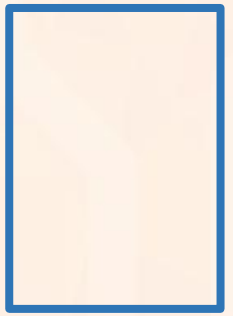
A quoi servent les priors?

Le modèle doit être capable de générer toutes les tailles de boîtes



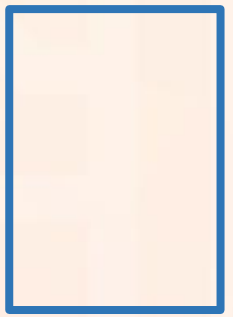
YoloV1

Prédiction
« from scratch »



YoloV2

Déformation



Choix du meilleur prior

A quoi ser[illegible]t l[illegible]iors?

Le mod[illegible]

[illegible]les de boîtes

Moins de variance en sortie
⇒ moins d'overfitting et
entraînement plus stable
⇒ meilleure généralisation

[illegible]rédiction
« from scratch »

[illegible]oV2

P2

Déformation

Choix du meilleur prior

A quoi ser[illegible]t l[illegible]iors?

Le mod[illegible]

[illegible]les de boîte[illegible]

[illegible]rédi[illegible]

[illegible]m

oV2

Moins de variance en sortie
⇒ moins d'overfitting et
entraînement plus stable
⇒ meilleure généralisation

MAIS

Mauvais choix des priors
⇒ entraînement inefficace
Trop de priors
⇒ complexité du modèle élevée
⇒ redondance des priors
⇒ overfitting

A quoi ser[illegible]t l[illegible]iors?

Le mod[illegible]

Moins de variance en sortie
⇒ moins d'overfitting et
entraînement plus stable
⇒ m[illegible]

Comment (bien) choisir ces priors?

[illegible]es de boîte[illegible]

[illegible]rédi[illegible]

[illegible]m

[illegible]riors

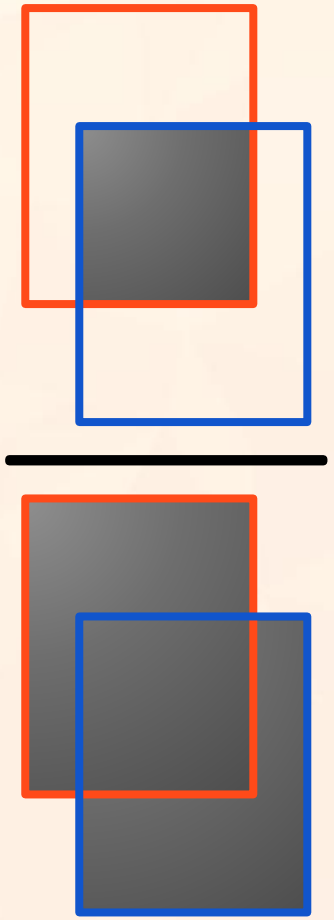
⇒ entraînement inefficace

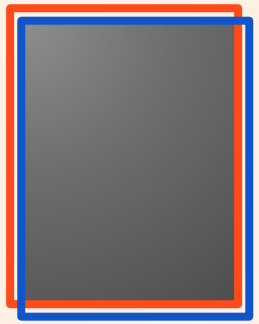
Trop de priors
⇒ complexité du modèle élevée
⇒ redondance des priors
⇒ overfitting

Intersection over Union (IoU)

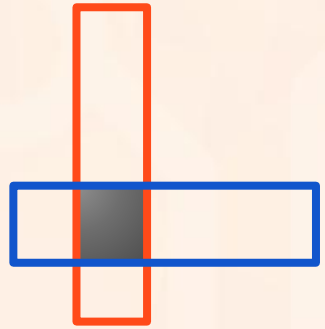
Simplement le rapport entre l'aire de l'intersection et de l'union entre 2 boîtes

$$IoU(b_1; b_2) =$$





$IoU \simeq 1$



$IoU \simeq 0$

Distance de Jaccard

$$d_J(b_1; b_2) = 1 - IoU(b_1; b_2)$$

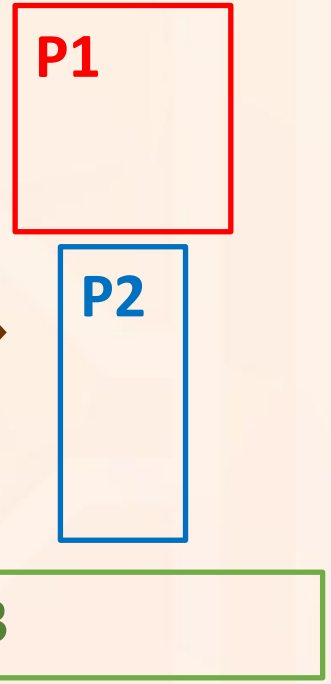
Calcul des priors

On effectue un clustering sur les dimensions $(w; h)$ avec la distance de Jaccard en partant du principe que les boîtes sont centrées entre elles.

Boîtes Train set

K-means

$d_{Jaccard}$



P priors

Calcul des priors

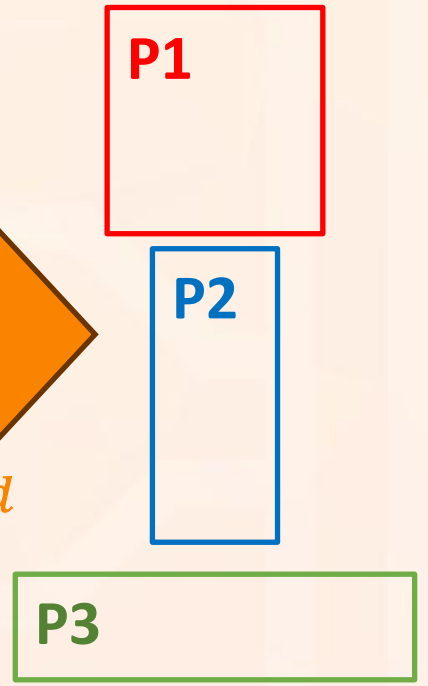
On effectue un clustering sur les dimensions $(w; h)$ avec la distance de Jaccard en partant du principe que les boîtes sont centrées entre elles.

On considère **TOUTES** les boîtes du train set, peu importe les classes

Boîtes Train set

K-means

$d_{Jaccard}$



P priors

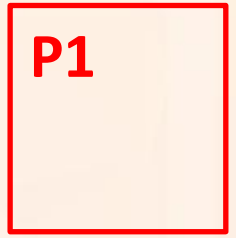
Calcul des priors

On effectue un clustering sur les dimensions $(w; h)$ avec la distance de Jaccard en partant du principe que les boîtes sont centrées entre elles.

On considère **TOUTES** les boîtes du train set, peu importe les classes

Boîtes Train set

K-means



P priors

P est un **hyper-paramètre** (à trouver empiriquement ☺)

Architecture de YoloV2

Feature Extractor

Feature Maps

Prior #1

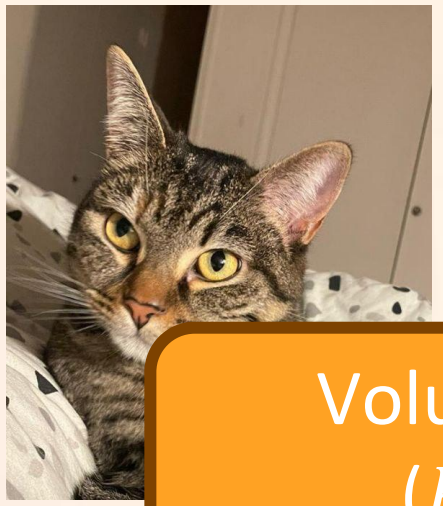
Conv 1x1

Conv 1x1

Prior #2

Pour chaque prior, on génère une detection map

Architecture de YoloV2



Feature Extractor

Feature Maps

Prior #1

Conv 1x1

Conv 1x1

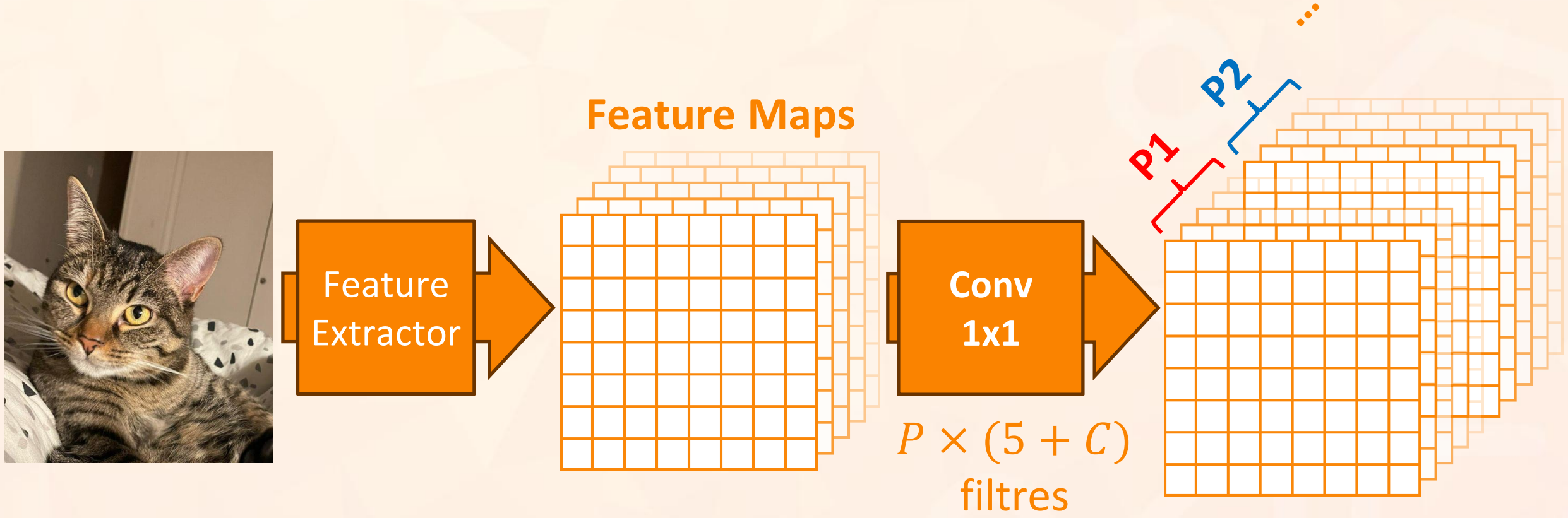
Prior #2

Volume de sortie en **4D**!
(P detection maps)

un peu relou...

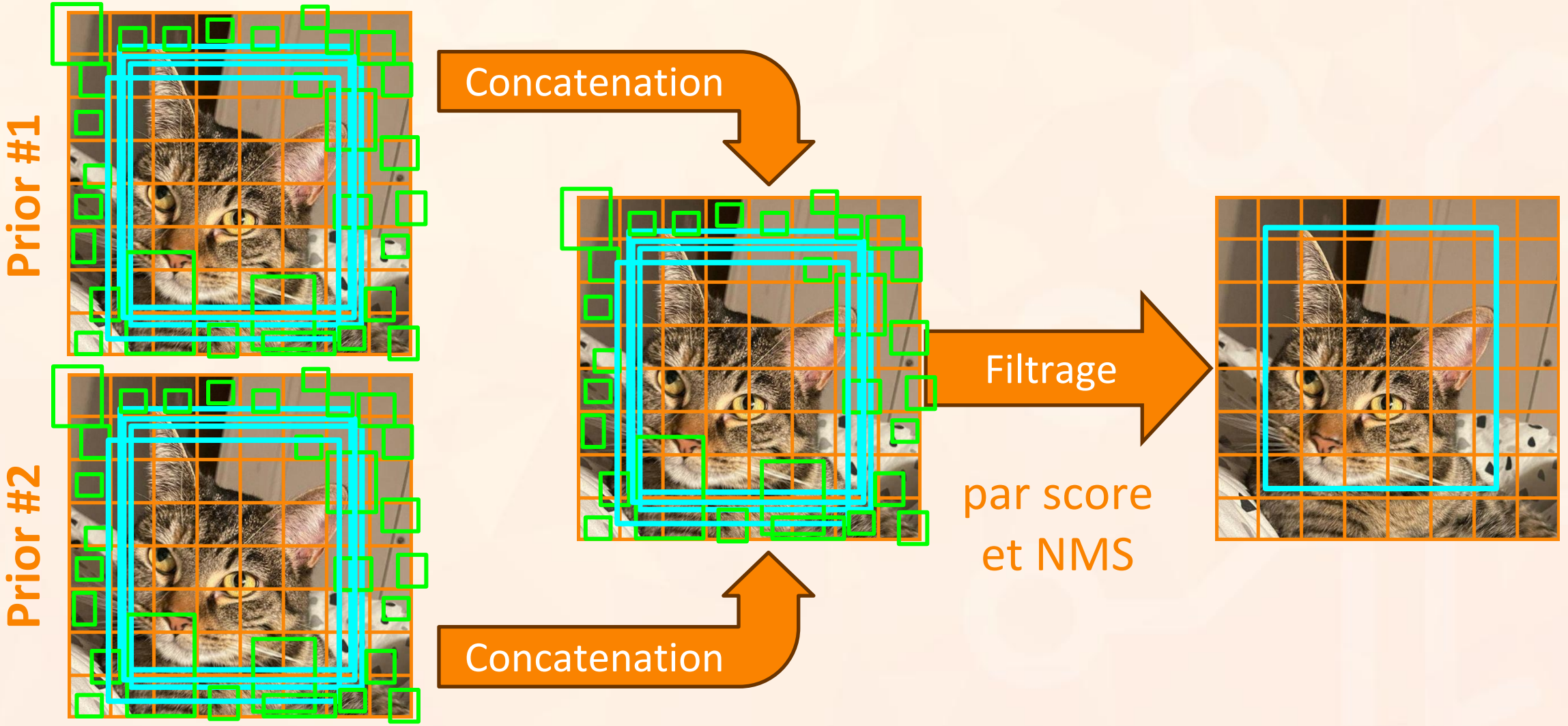
Pour chaque prior, on génère une detection map

En pratique

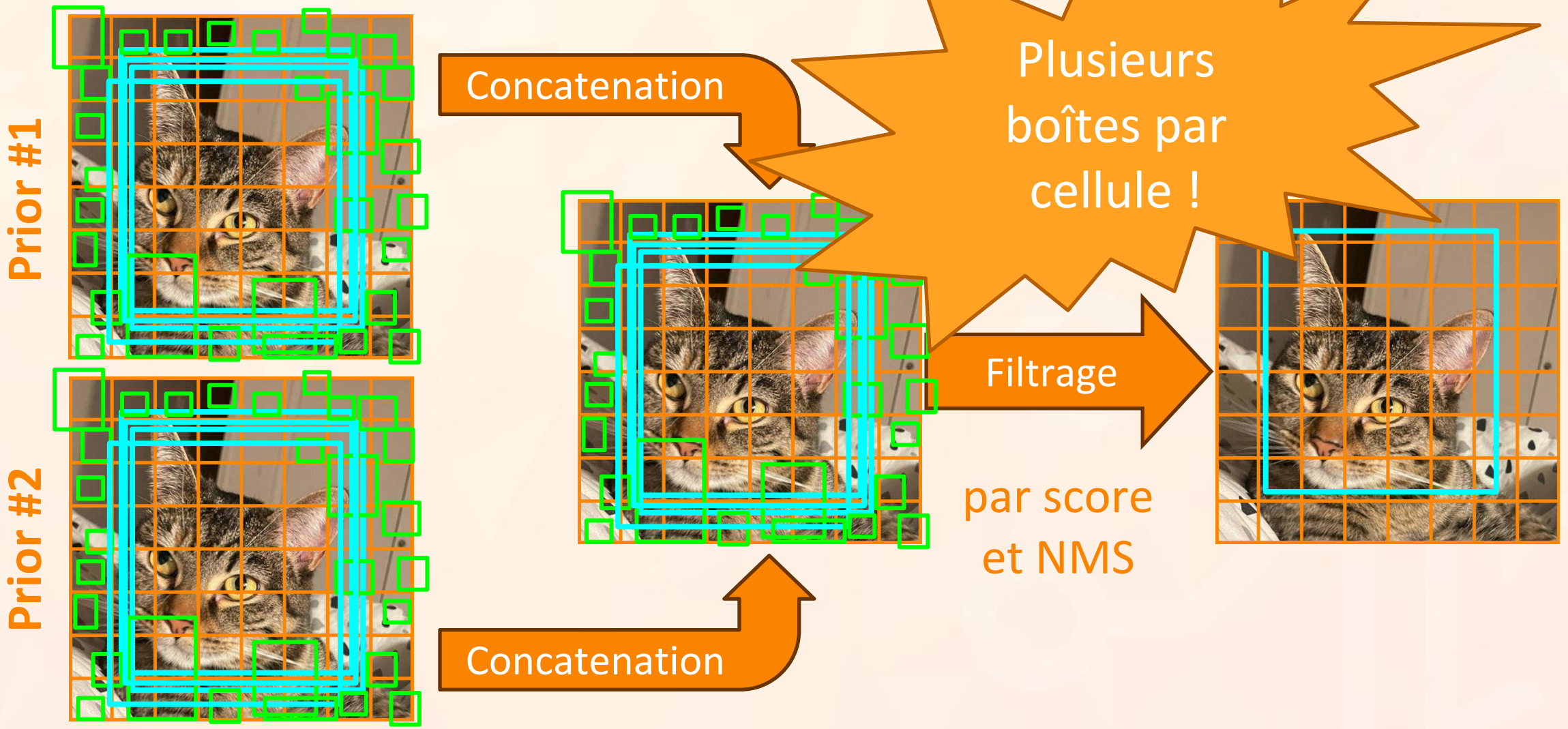


On peut factoriser les convolutions en une seule « grosse » convolution

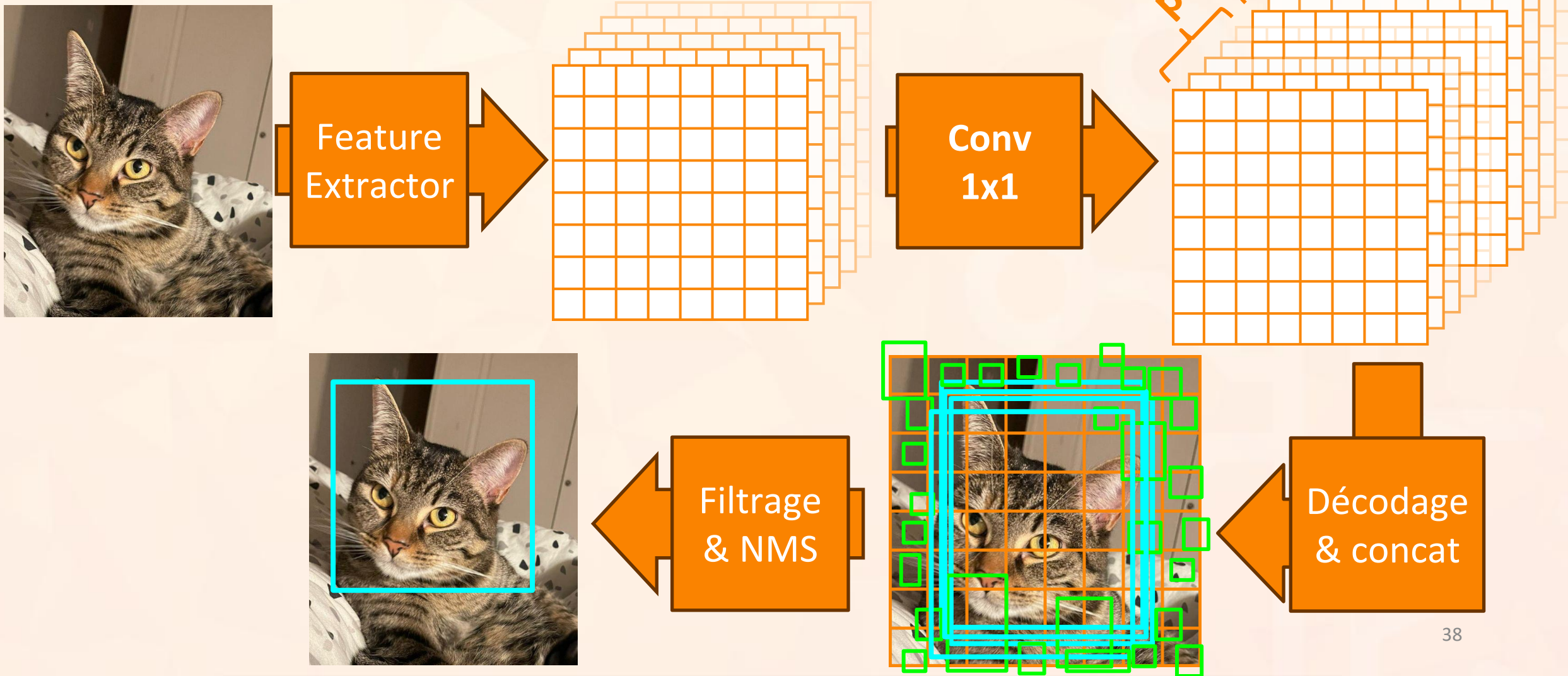
Post-process : même principe



Post-process : même principe



En résumé

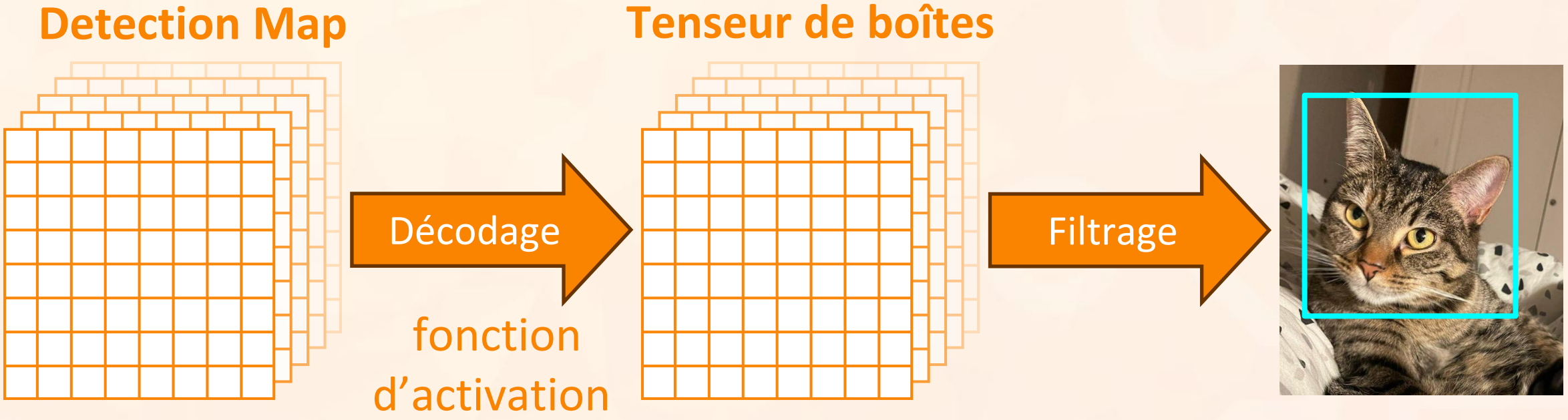

Feature Maps
priors
Feature Extractor
Conv 1x1
Décodage & concat
Filtrage & NMS

Questions?

Décodage des boîtes

Une fonction d’activation pas comme les autres

Récap rapide



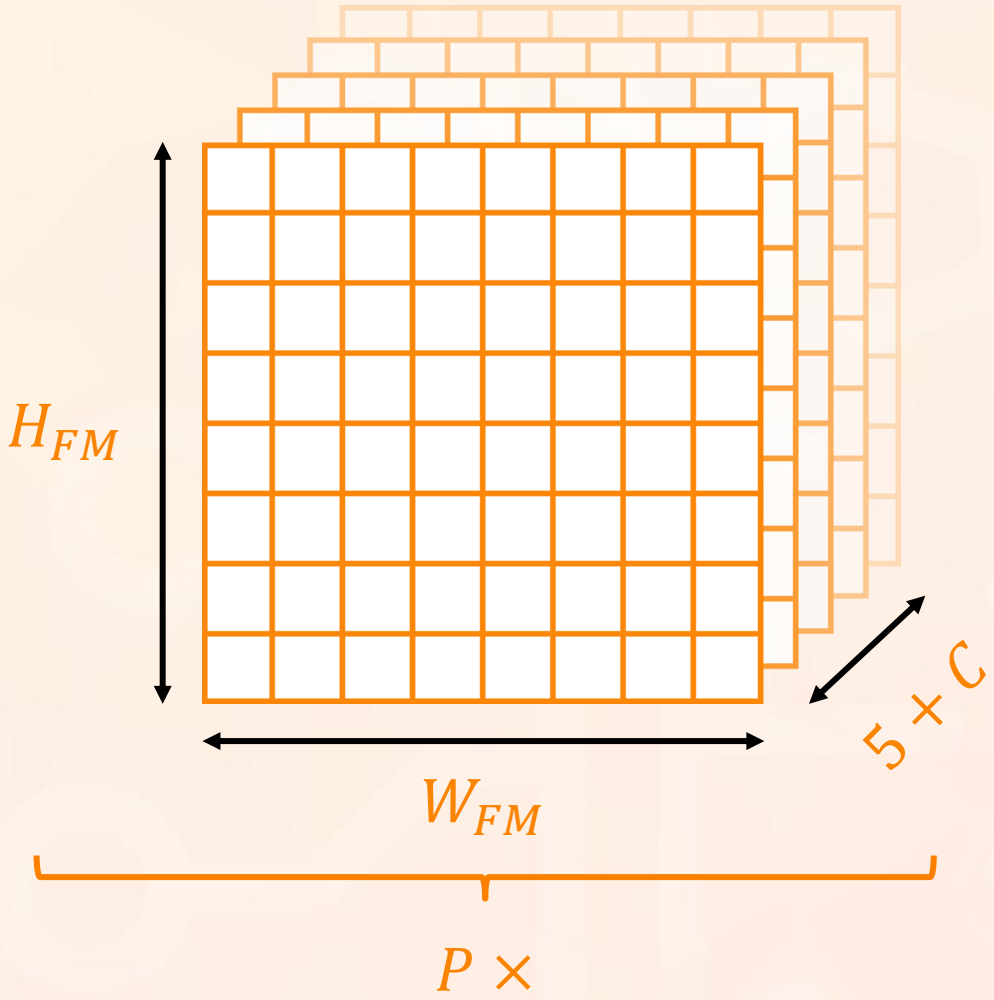
Notations

On introduit les constantes suivantes :

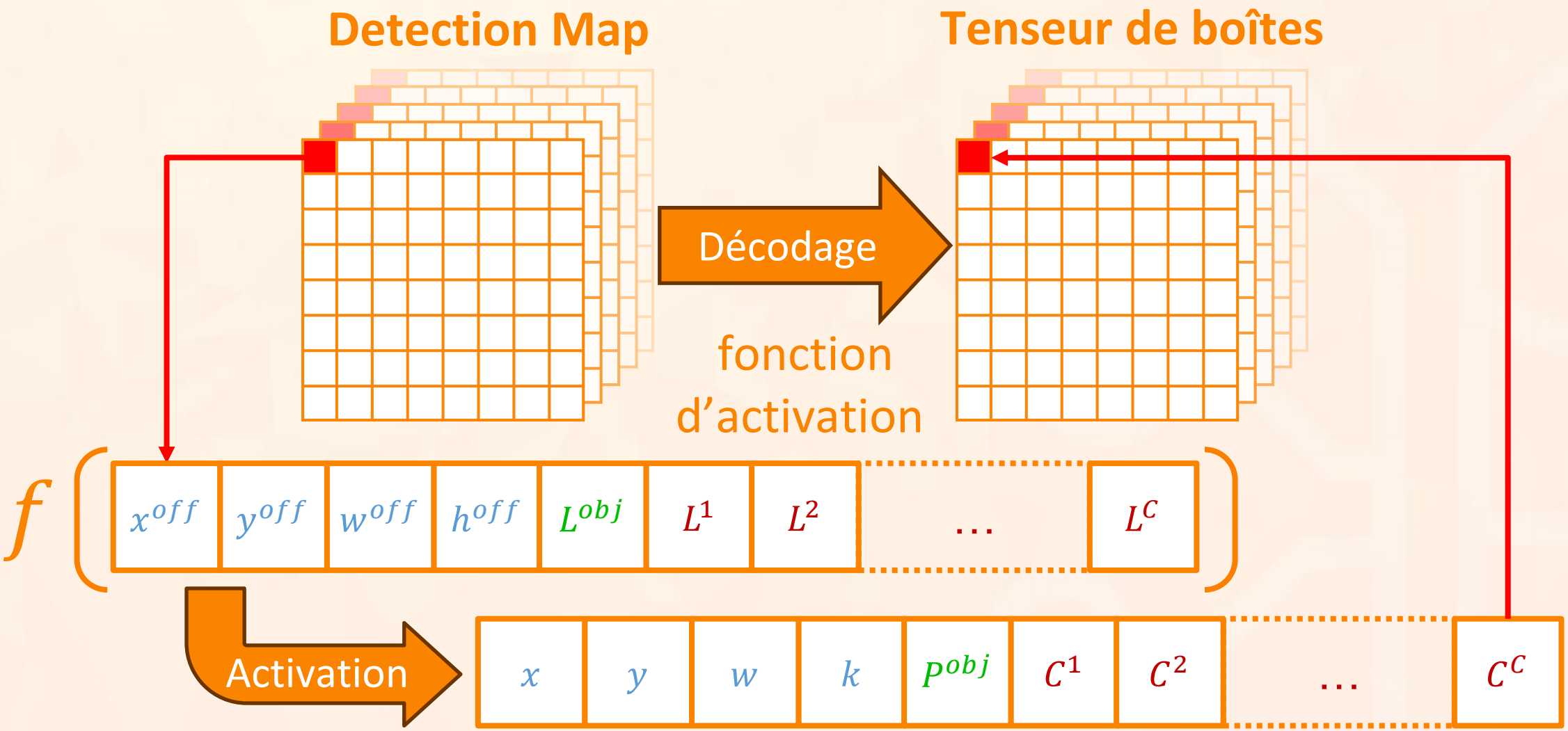
- P le nombre de priors
- W_{FM} et H_{FM} les dimensions de la feature map
- C le nombre de classes

Les indices utilisés par la suite :

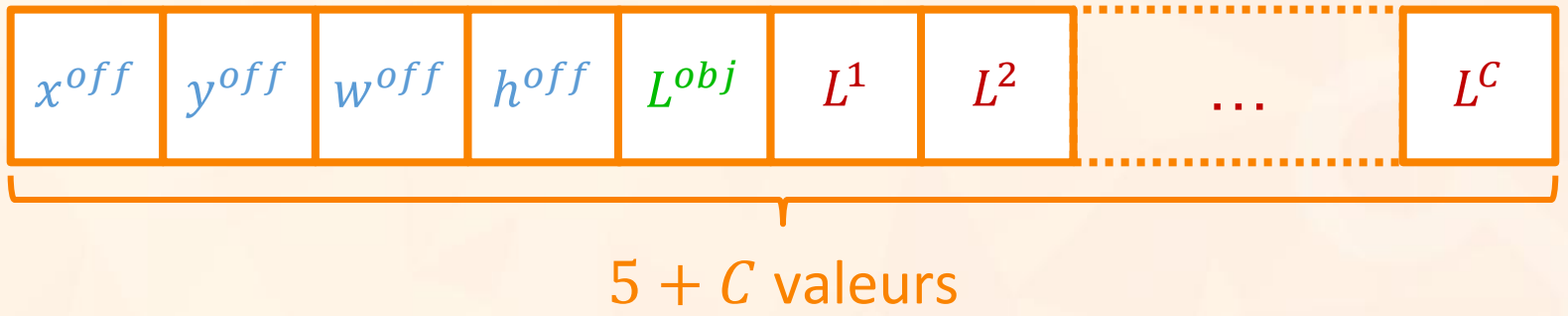
- k le prior considéré
- i, j la position de la cellule dans la feature map
- c l'indice de la classe considérée



Décodage d'une boîte



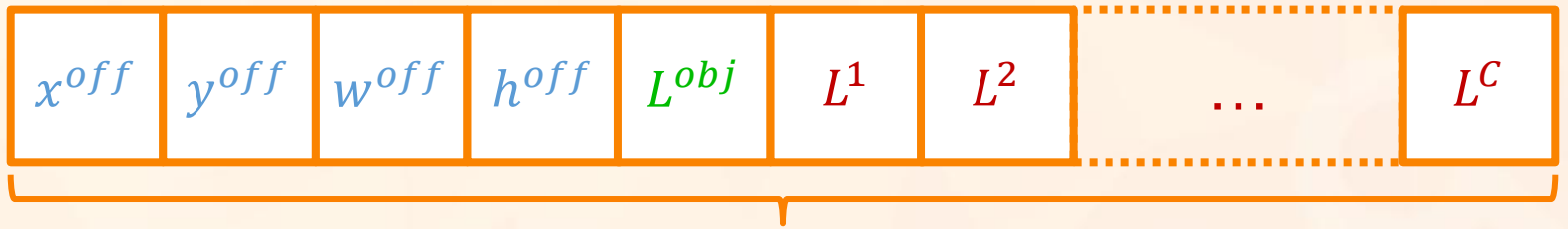
Décodage d'une boîte



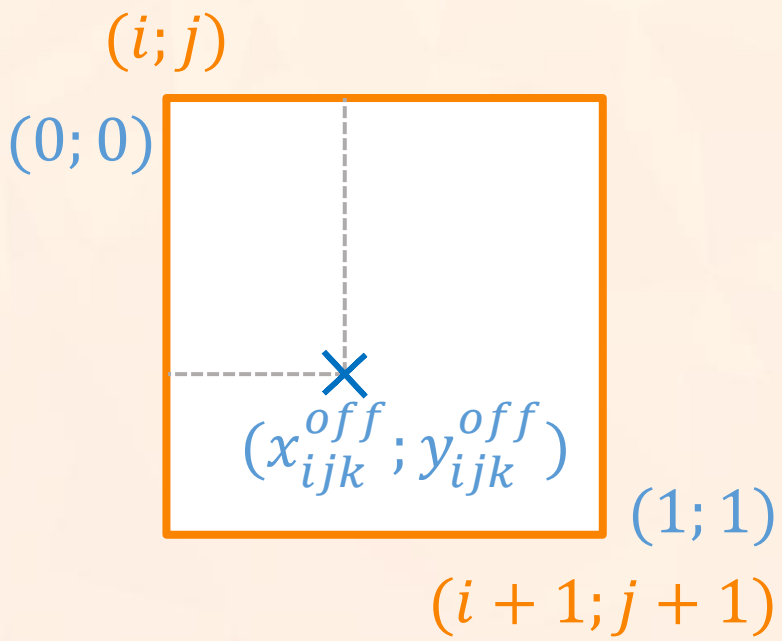
Pour une cellule (i, j) et un prior k

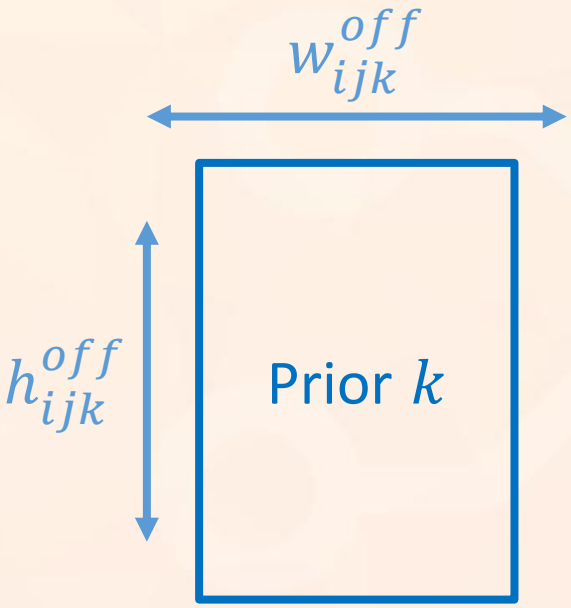
- x_{ijk}^{off} et y_{ijk}^{off} : position du centre de la boîte par rapport a la **cellule**
- w_{ijk}^{off} et h_{ijk}^{off} : taille de boîte relative au prior k
- L_{ijk}^{obj} : probabilité que la boîte contienne un objet (logit)
- L_{ijk}^{c} : probabilité conditionelle que la boîte contienne c (logit)

Décodage d'une boîte



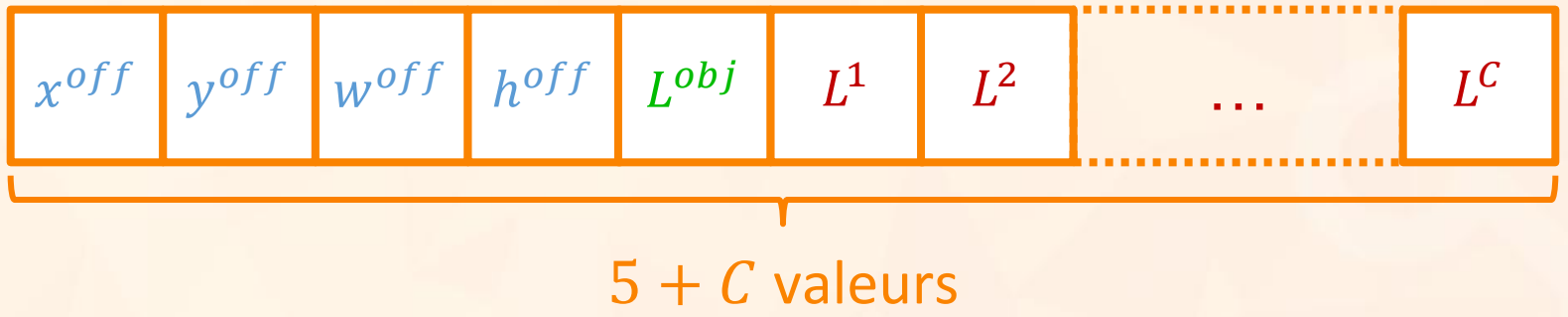
Pour une cellule (i, j) et un prior k





Déformation du prior k

Décodage d'une boîte

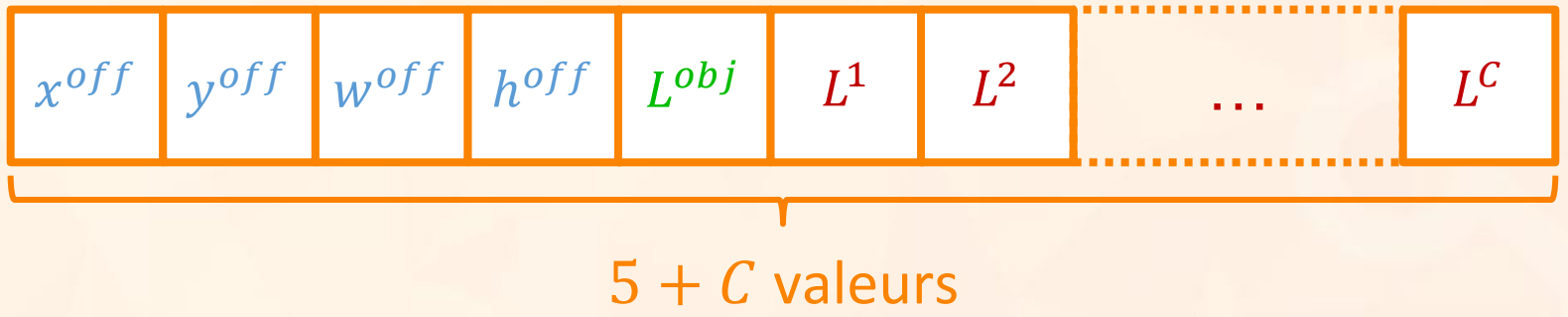


Pour une cellule (i, j) et un prior k

$$x_{ijk} = i + \sigma\left(x_{ijk}^{off}\right) \qquad w_{ijk} = w_k^p \times \exp\left(w_{ijk}^{off}\right)$$

$$y_{ijk} = j + \sigma\left(y_{ijk}^{off}\right) \qquad h_{ijk} = h_k^p \times \exp\left(h_{ijk}^{off}\right)$$

Décodage d'une boîte



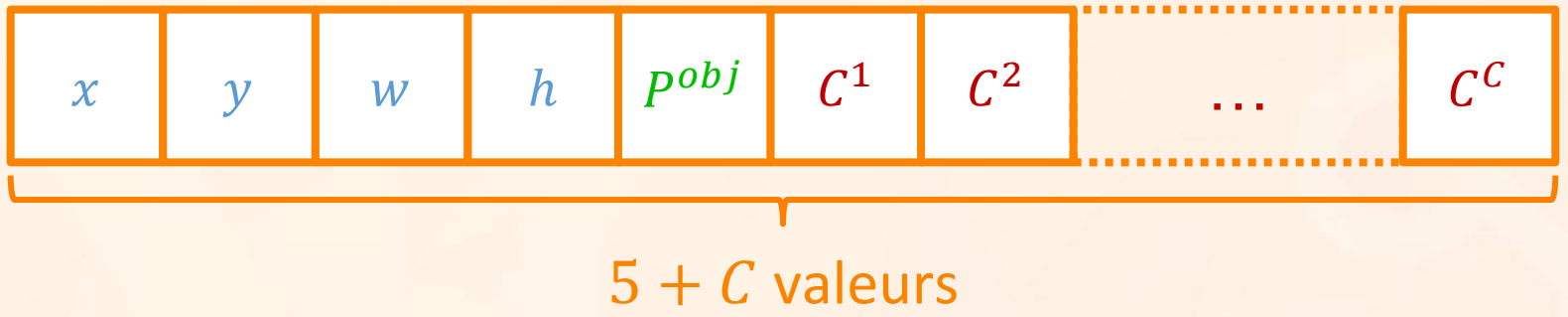
Pour une cellule (i, j) et un prior k

$$P_{ijk}^{obj} = \sigma\left(L_{ijk}^{off}\right)$$

$$C_{ijk}^{c} = \underset{c}{\text{softmax}}[L_{ijk}^{c}]$$

$$P_{ijk}^{c} = P_{ijk}^{obj} \times C_{ijk}^{c}$$

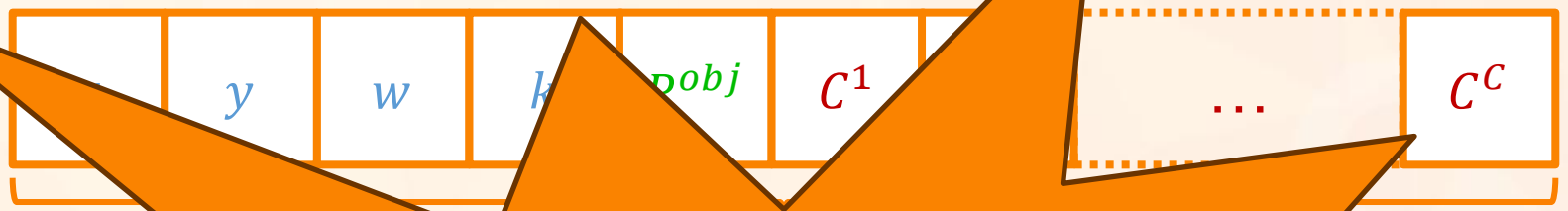
Décodage d'une boîte



Pour une cellule (i, j) et un prior k

- x_{ijk} et y_{ijk}: position du centre de la boîte par rapport a la **feature map**
- w_{ijk} et h_{ijk}: taille de la boîte par rapport a la **feature map**
- P_{ijk}^{obj} : probabilité que la boîte contienne un objet
- C_{ijk}^{c} : probabilité conditionelle que la boîte contienne c

Décodage d'une boîte



L'information spatiale est maintenant directement contenue dans les **vecteurs post-activation**

Pour une

- x_{ijk} et y_{ijk} : p
- w_{ijk} et
- P_{ijk}^{obj} : probabilité
- C_{ijk}^{c} : probabilité conditi e que la îte contienne c

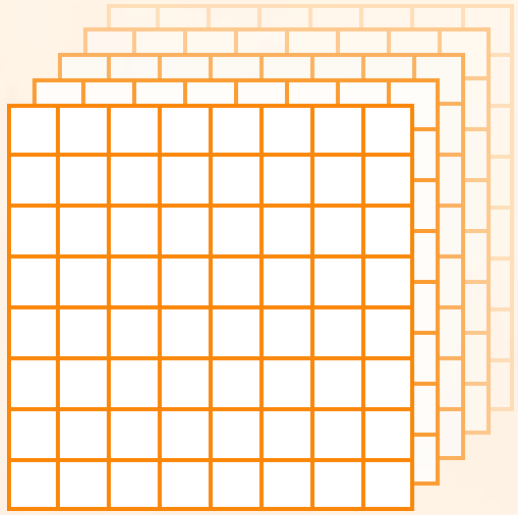
la **feature map**

map

un obj

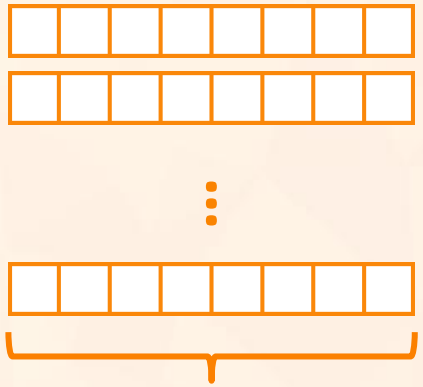
Filtrage des boîtes

Tenseur de boîtes



« Flatten »

$P \times W_{FM} \times H_{FM}$ boîtes

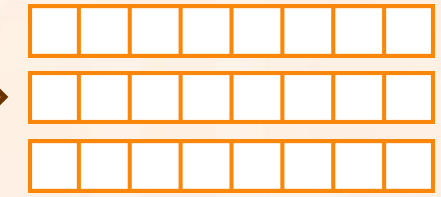


$5 + C$

Filtrage

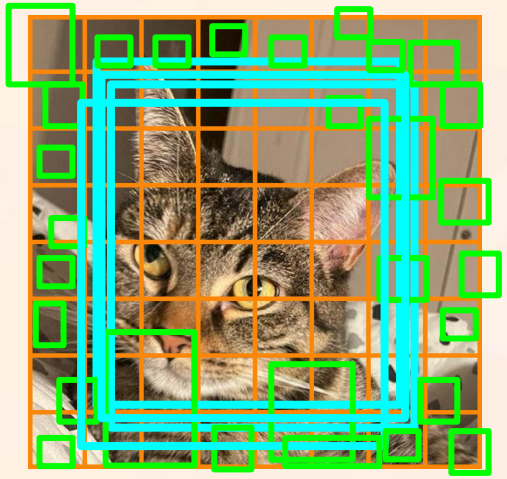
Par rapport à P_{ijk}^{obj}

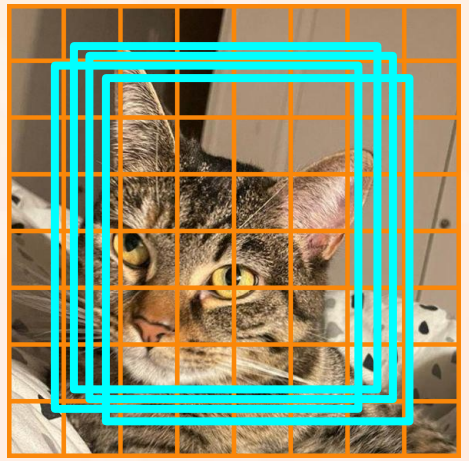
N boîtes



On conserve les boîtes ssi.

$$P_{ijk}^{obj} > \alpha$$





Non-Maximum Suppression (NMS)

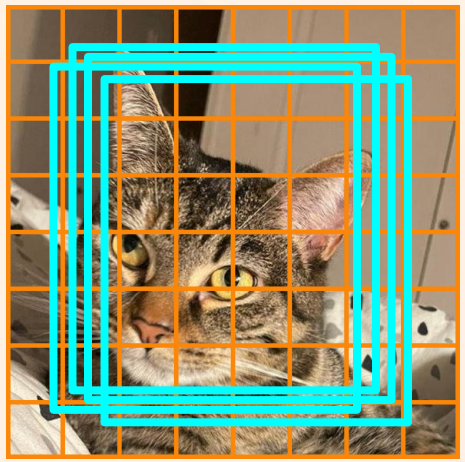
Plusieurs boîtes peuvent être prédite pour un même « objet »

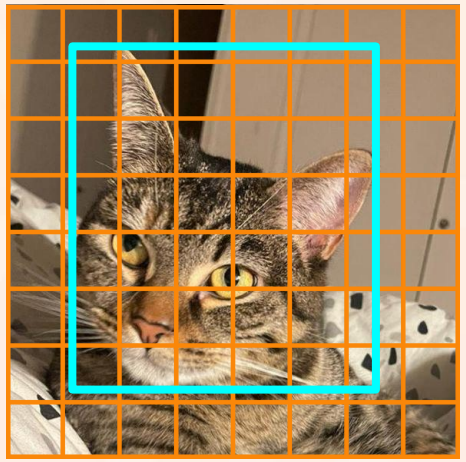
On considère que 2 boîtes b_1 et b_2 sont superposées si :

$$C_1 = C_2$$

$$IoU(b_1; b_2) > \beta$$

On garde uniquement la boîte avec le score le plus élevé





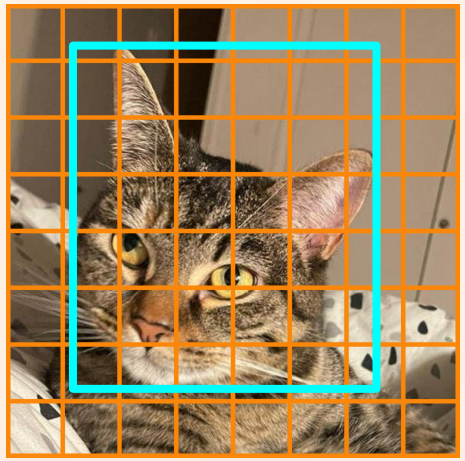
Dernière étape!

M boîtes

Scaling des boîtes

Objet #0
- $x = 30$
- $y = 25$
- $w = 50$
- $h = 50$

0.95	chat
0.03	chien
0.02	…

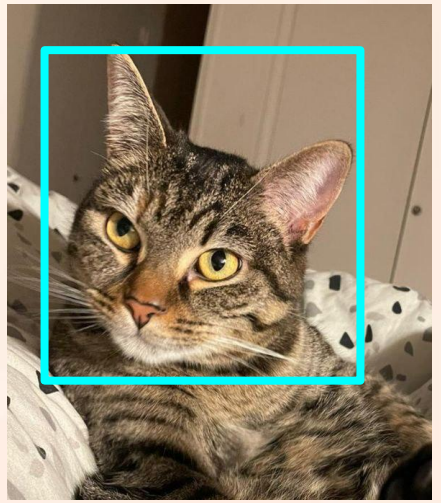


Coordonées relatives a la feature map

↓

Coordonées relative a l'image

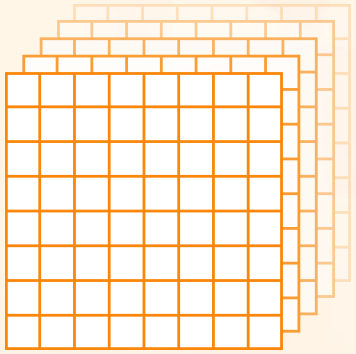
Le facteur de scaling depend de la taille de la feature map, et donc du feature extractor utilisé



Gros récap

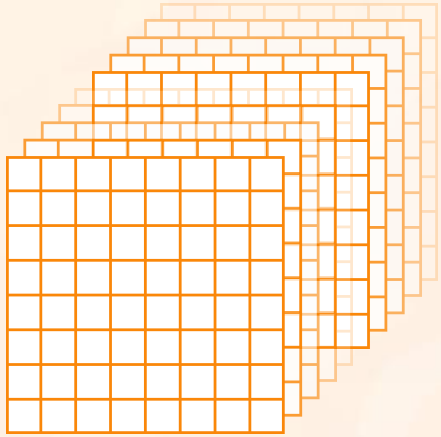
FE

Feature Maps



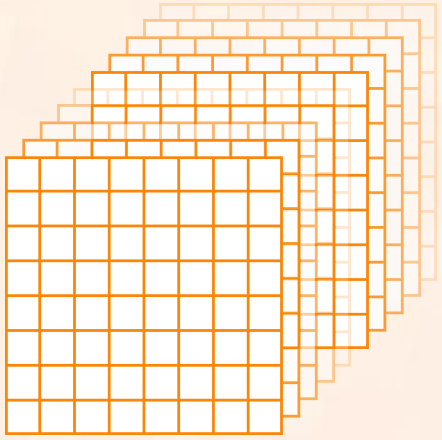
Conv

Detection Maps



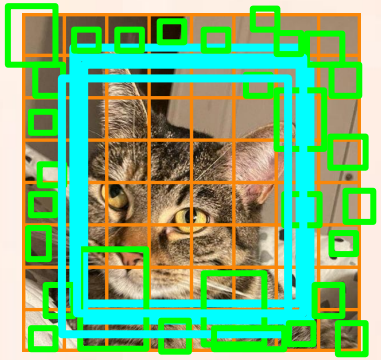
Décodage

Tenseur de boîtes



Modèle

Post-process

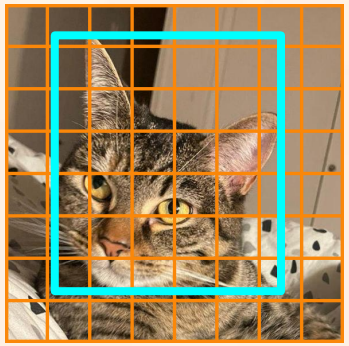


Filtrage

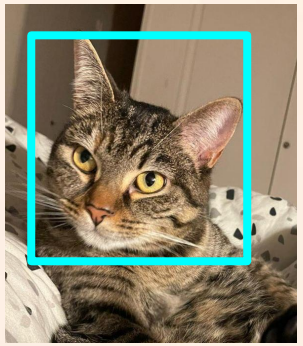
α

NMS

β



Scaling

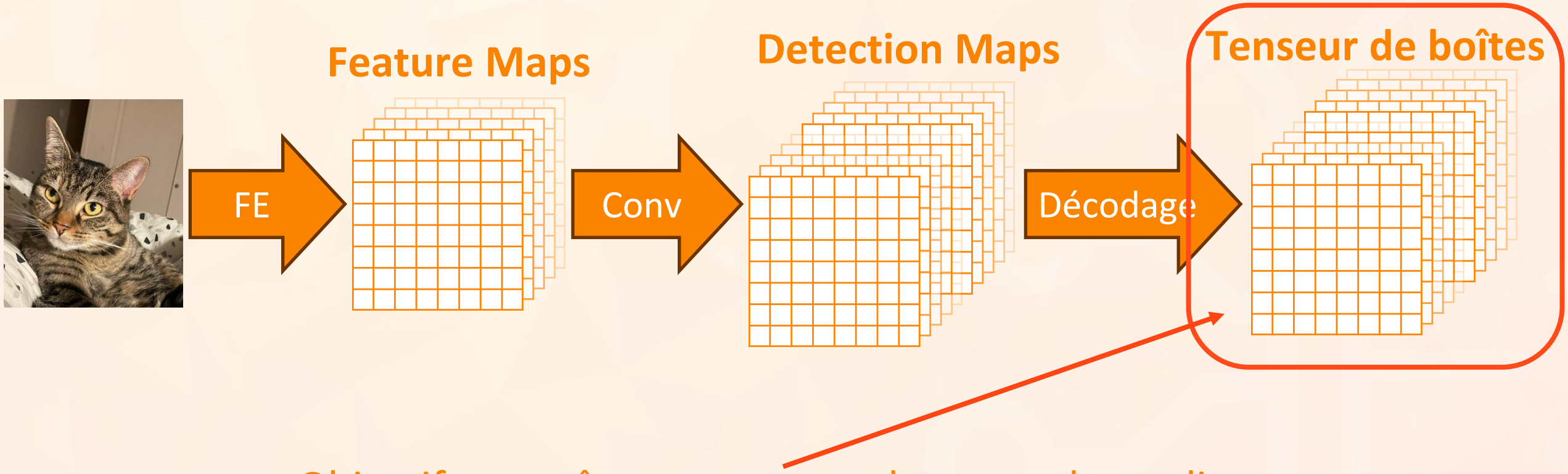


Questions?

Entraînement

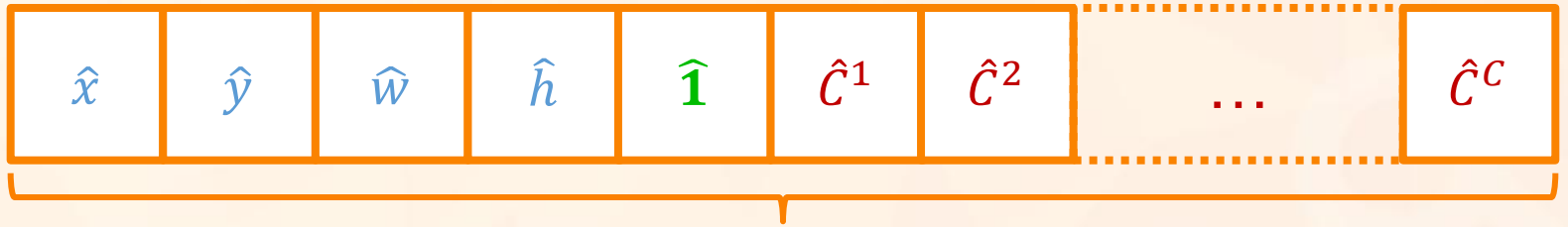
Tenseur de GT & Loss

Qu’est qu’on entraîne dans tout ça?



Objectif : entraîner ce truc par descente de gradient

Génération du tenseur de ground truth



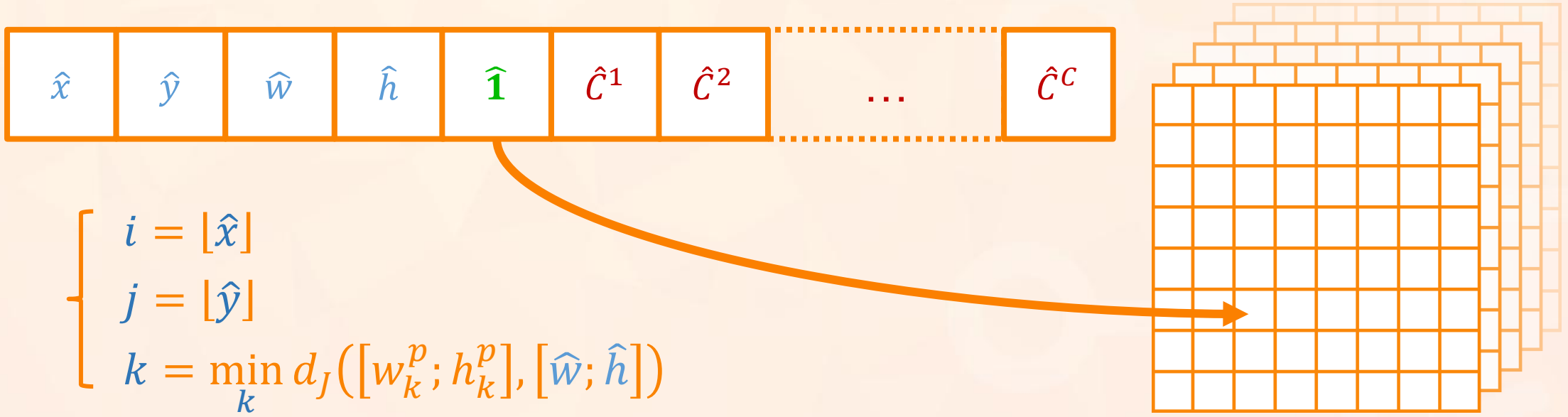
Pour chaque boîte de notre train set, on construit un vecteur de GT

- $\hat{x}$ et $\hat{y}$: position du centre de la boîte par rapport a la **feature map**
- $\hat{w}$ et $\hat{h}$: taille de la boîte par rapport a la **feature map**
- $\hat{\mathbf{1}}$: probabilité que la boîte contienne un objet, donc toujours 1
- $\hat{C}^c$: la classe de l'objet, sous la forme d'un vecteur one-hot

Génération du tenseur de ground truth

On initialise un tenseur de taille $W_{FM} \times H_{FM} \times P \times (5 + C)$ avec des 0.
Pour chaque boîte :

- On trouve le meilleur prior k avec la distance de Jaccard.
- On place le vecteur dans la cellule $(i; j)$ du volume du prior k.



$$\begin{cases} i = \lfloor \hat{x} \rfloor \\ j = \lfloor \hat{y} \rfloor \\ k = \min_k d_J\left(\left[w_k^p; h_k^p\right], \left[\hat{w}; \hat{h}\right]\right) \end{cases}$$

Intuition sur la loss function

Cas 1 : la cellule contient effectivement une boîte : regression sur la boîte

$$\mathcal{L}\left(\begin{array}{|c|c|c|c|c|c|c|c|c|}\hline x & y & w & h & P^{obj} & C^1 & C^2 & \dots & C^C \\ \hline \hat{x} & \hat{y} & \hat{w} & \hat{h} & \hat{\mathbf{1}} & \hat{C}^1 & \hat{C}^2 & \dots & \hat{C}^C \\ \hline\end{array}\right)$$

Cas 2 : la cellule ne contient pas de boîte : regression sur l'objectness

$$\mathcal{L}\left(\begin{array}{|c|c|c|c|c|c|c|c|c|}\hline x & y & w & h & P^{obj} & C^1 & C^2 & \dots & C^C \\ \hline 0 & 0 & 0 & 0 & \hat{\mathbf{0}} & 0 & 0 & \dots & 0 \\ \hline\end{array}\right)$$

Loss

$$\mathcal{L} = \lambda_1 \sum_{i=1}^{W_{FM}} \sum_{j=1}^{H_{FM}} \sum_{k=1}^{P} \widehat{\mathbf{1}}_{ijk} \left[\left(x_{ijk} - \hat{x}_{ijk} \right)^2 + \left(y_{ijk} - \hat{y}_{ijk} \right)^2 \right]$$

$$+ \lambda_1 \sum_{i=1}^{W_{FM}} \sum_{j=1}^{H_{FM}} \sum_{k=1}^{P} \widehat{\mathbf{1}}_{ijk} \left[\left(\sqrt{w_{ijk}} - \sqrt{\hat{w}_{ijk}} \right)^2 + \left(\sqrt{h_{ijk}} - \sqrt{\hat{h}_{ijk}} \right)^2 \right]$$

$$+ \lambda_2 \sum_{i=1}^{W_{FM}} \sum_{j=1}^{H_{FM}} \sum_{k=1}^{P} \widehat{\mathbf{1}}_{ijk} \left(P_{ijk}^{obj} - \mathrm{IoU}\left(b_{ijk}; \hat{b}_{ijk} \right) \right)^2$$

$$+ \lambda_3 \sum_{i=1}^{W_{FM}} \sum_{j=1}^{H_{FM}} \sum_{k=1}^{P} \left(1 - \widehat{\mathbf{1}}_{ijk} \right) \left(P_{ijk}^{obj} - 0 \right)^2$$

$$+ \lambda_4 \sum_{i=1}^{W_{FM}} \sum_{j=1}^{H_{FM}} \sum_{k=1}^{P} \widehat{\mathbf{1}}_{ijk} \sum_{c=1}^{C} \left(C_{ijk}^{c} - \hat{C}_{ijk}^{c} \right)^2$$

wtf?

Loss

$$\mathcal{L} = \lambda_1 \sum_{i=1}^{W_{FM}} \sum_{j=1}^{H_{FM}} \sum_{k=1}^{P} \hat{\mathbf{1}}_{ijk} \left[\left(x_{ijk} - \hat{x}_{ijk}\right)^2 + \left(y_{ijk} - \hat{y}_{ijk}\right)^2 \right]$$

$$+ \lambda_1 \sum_{i=1}^{W_{FM}} \sum_{j=1}^{H_{FM}} \sum_{k=1}^{P} \hat{\mathbf{1}}_{ijk} \left[\left(\sqrt{w_{ijk}} - \sqrt{\hat{w}_{ijk}}\right)^2 + \left(\sqrt{h_{ijk}} - \sqrt{\hat{h}_{ijk}}\right)^2 \right]$$

Loss sur la position et la taille

$$+ \lambda_2 \sum_{i=1}^{W_{FM}} \sum_{j=1}^{H_{FM}} \sum_{k=1}^{P} \hat{\mathbf{1}}_{ijk} \left(P_{ijk}^{obj} - \mathrm{IoU}\left(b_{ijk}; \hat{b}_{ijk}\right)\right)^2$$

$$+ \lambda_3 \sum_{i=1}^{W_{FM}} \sum_{j=1}^{H_{FM}} \sum_{k=1}^{P} \left(1 - \hat{\mathbf{1}}_{ijk}\right) \left(P_{ijk}^{obj} - 0\right)^2$$

Loss sur l'objectness

$$+ \lambda_4 \sum_{i=1}^{W_{FM}} \sum_{j=1}^{H_{FM}} \sum_{k=1}^{P} \hat{\mathbf{1}}_{ijk} \sum_{c=1}^{C} \left(C_{ijk}^{c} - \hat{C}_{ijk}^{c}\right)^2$$

Loss de classification

Loss : position et taille des boîtes

Pénalisation sur la position : écarts au carré

$$E_{ijk}^{xy} = \hat{\mathbf{1}}_{ijk} \left[\left(x_{ijk} - \hat{x}_{ijk} \right)^2 + \left(y_{ijk} - \hat{y}_{ijk} \right)^2 \right]$$

Pénalisation sur la taile : écarts au carré des racines carrées

$$E_{ijk}^{hw} = \hat{\mathbf{1}}_{ijk} \left[\left(\sqrt{w_{ijk}} - \sqrt{\hat{w}_{ijk}} \right)^2 + \left(\sqrt{h_{ijk}} - \sqrt{\hat{h}_{ijk}} \right)^2 \right]$$

Loss : position et taille des boîtes

Pén[illegible]ion [illegible]écarts au carré

$$\left[\left(x_{ijk} - \hat{x}_{ijk}\right)^2 + \left(y_{ijk} - \hat{y}_{ijk}\right)^2\right]$$

Racines carrées
⇔
pénalisation plus forte pour les petites boîtes

[illegible]écarts au carré des racines carrées

$$E_{ijk}^{hw} = \hat{\mathbf{1}}_{ijk}\left[\left(\sqrt{w_{ijk}} - \sqrt{\hat{w}_{ijk}}\right)^2 + \left(\sqrt{h_{ijk}} - \sqrt{\hat{h}_{ijk}}\right)^2\right]$$

Loss : objectness

Pénalisation positive: écarts au carré entre l'objectness et IoU

$$E_{ijk}^{obj} = \hat{\mathbf{1}}_{ijk} \left(\mathrm{P}_{\mathrm{ijk}}^{\mathrm{obj}} - IoU\left(b_{ijk}; \hat{b}_{ijk}\right) \right)^2$$

Pénalisation négative: minimisation de l'objectness

$$\overline{E_{ijk}^{obj}} = \left(1 - \hat{\mathbf{1}}_{ijk}\right) \times \left(P_{ijk}^{obj} - 0\right)^2$$

Loss : objectness

Pénalisation positive: écarts au carré entre l'objectness et IoU

$$E_{ijk}^{obj} = \hat{\mathbf{1}}_{ijk} \left(\mathrm{P}_{\mathrm{ijk}}^{\mathrm{obj}} - IoU\left(b_{ijk}; \hat{b}_{ijk}\right) \right)^2$$

Pénalisation négative: minimisation de l'

$$\overline{E_{ijk}^{obj}} = \left(1 - \hat{\mathbf{1}}_{ijk}\right) \times$$

L'objectness quantifie la qualité de la boîte prédite

Loss : classification

Pénalisation de classification: une simple MSE

$$E_{ijk}^{clf} = \frac{\hat{\mathbf{1}}_{ijk}}{C} \sum_{c=1}^{C} \left(C_{ijk}^{c} - \hat{C}_{ijk}^{c} \right)^2$$

… ou n'importe quelle loss de classification

$$E_{ijk}^{clf} = \hat{\mathbf{1}}_{ijk} \times \mathcal{L}_{cross-entropy}\left(C_{ijk}; \hat{C}_{ijk} \right)$$

Loss

En sommant pour tout i, j et k, on obtient alors

$$\mathcal{L} = \sum_{i=1}^{W_{FM}} \sum_{j=1}^{H_{FM}} \sum_{k=1}^{P} \lambda_1 \left(E_{ijk}^{xy} + E_{ijk}^{hw} \right) + \lambda_2 E_{ijk}^{obj} + \lambda_3 \overline{E_{ijk}^{obj}} + \lambda_4 E_{ijk}^{clf}$$

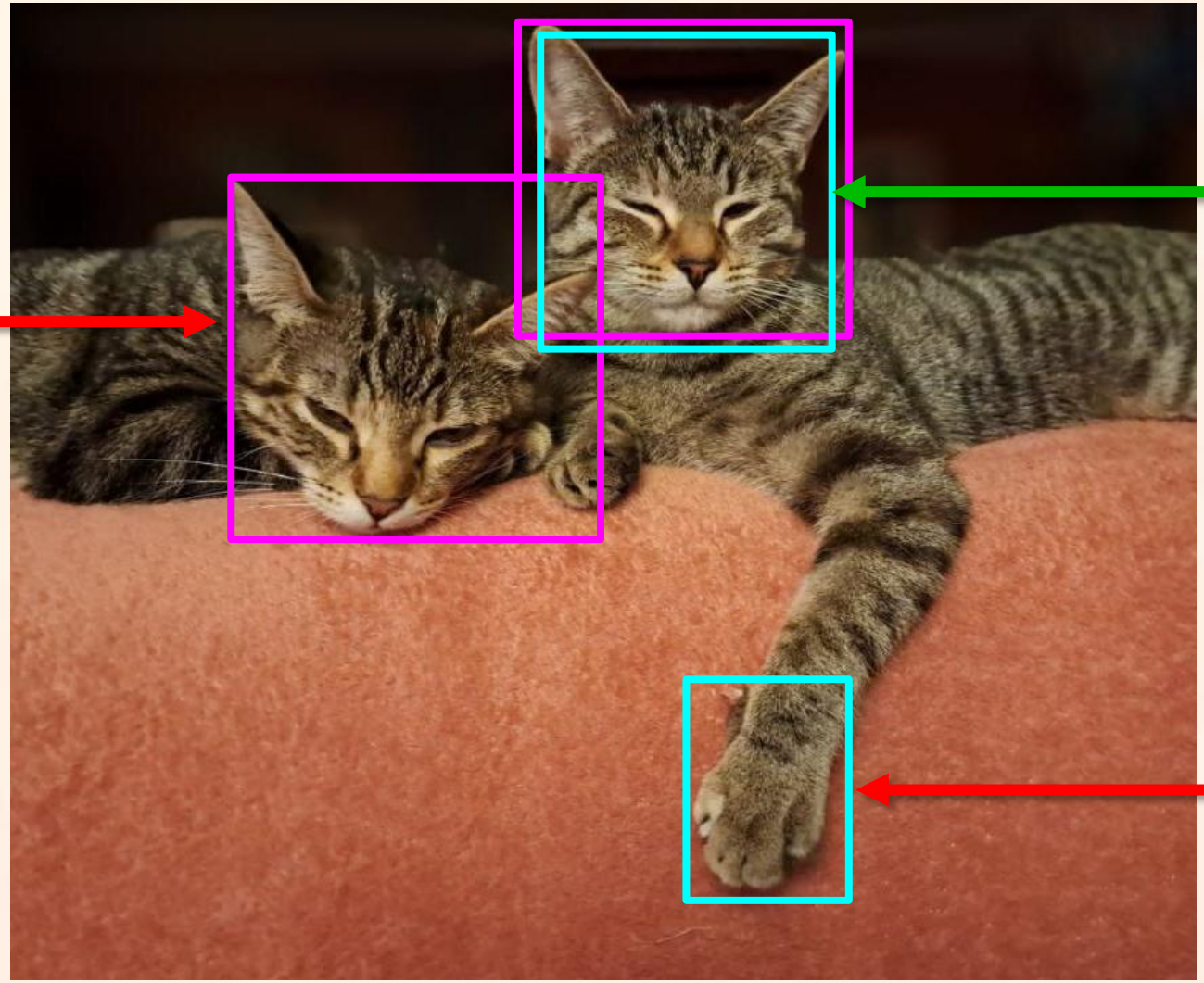
Questions?

Evaluation

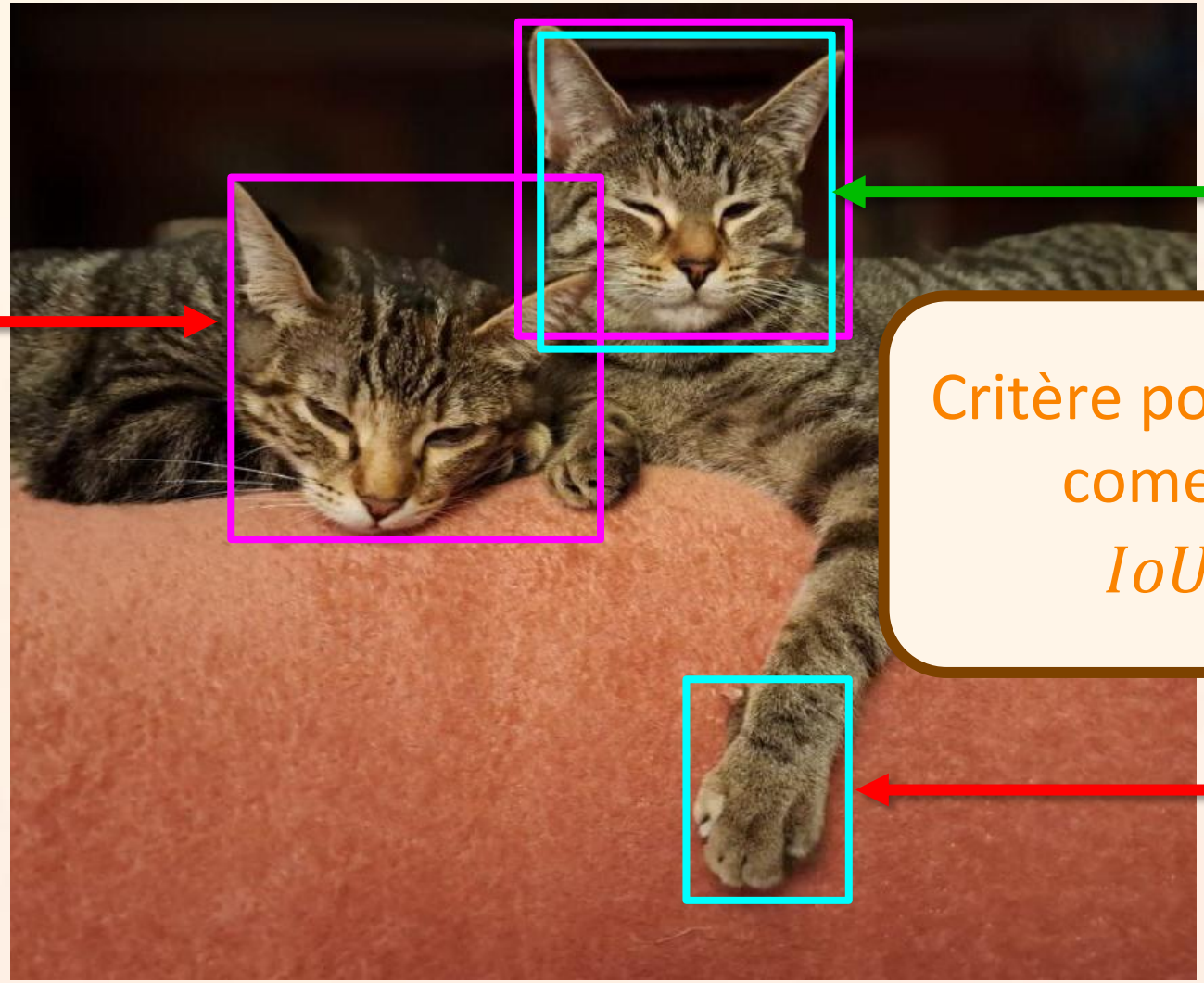
Une métrique un peu particulière

et discussion sur les hyperparamètres

VP, FP, FN et … VN?



VP, FP, FN et … VN?



Vrai positif

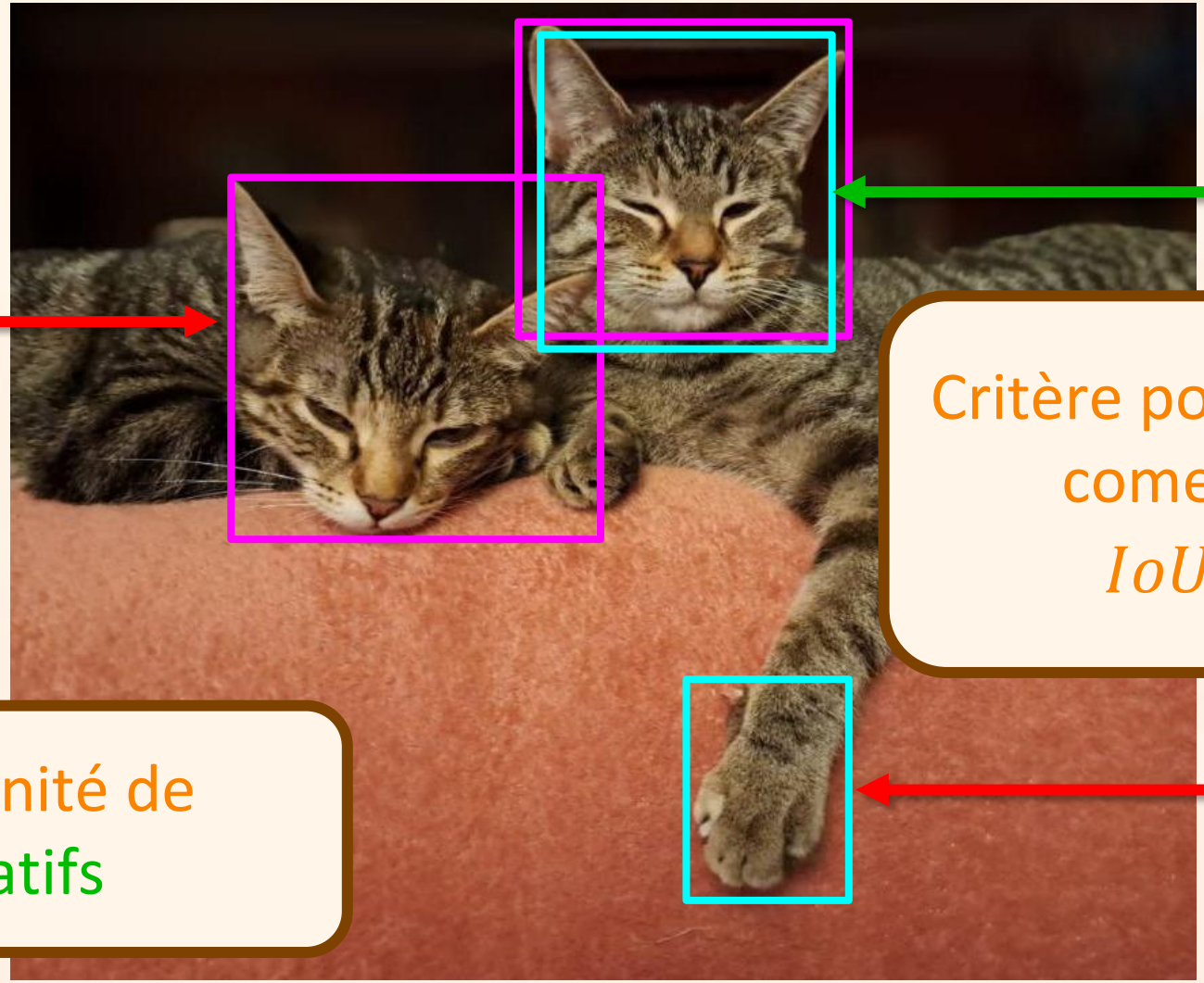
Faux négatif

Critère pour être considéré come vrai positif :

$$IoU\left(b;\hat{b}\right) > \beta$$

Faux positif

VP, FP, FN et … VN?



Vrai positif

Faux négatif

Faux positif

Critère pour être considéré come vrai positif :

$$IoU\left(b;\hat{b}\right) > \beta$$

Il y a une infinité de vrais négatifs

Précision et rappel

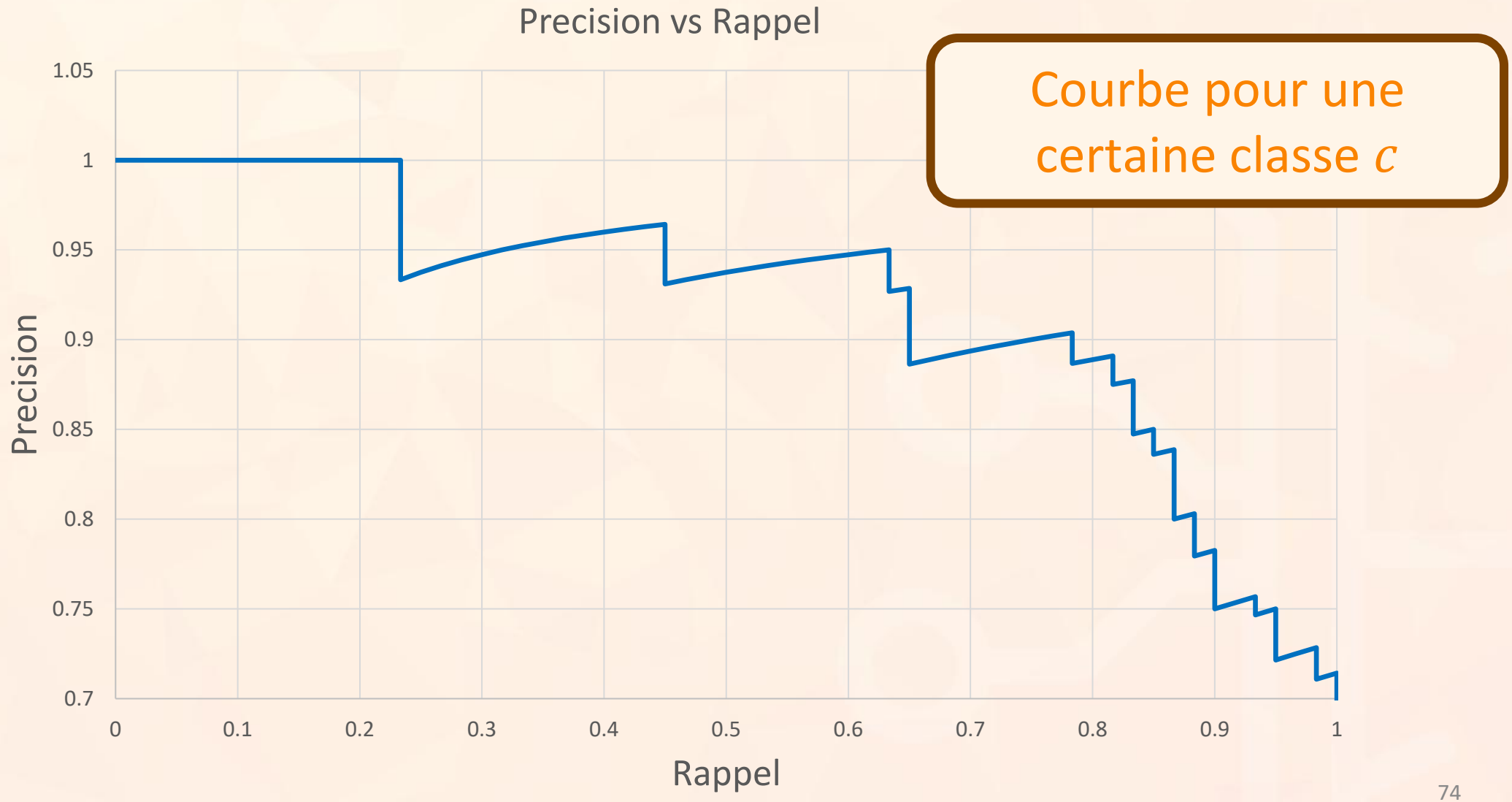
Précision : mesure de l'exactitude des détections

$$\text{precision} = \frac{VP}{VP + FP}$$

Rappel : mesure de l'exhaustivité des détections

$$\text{recall} = \frac{VP}{VP + FN} = \frac{VP}{N_{boîtes}}$$

Average Precision (AP)



Courbe pour une certaine classe c

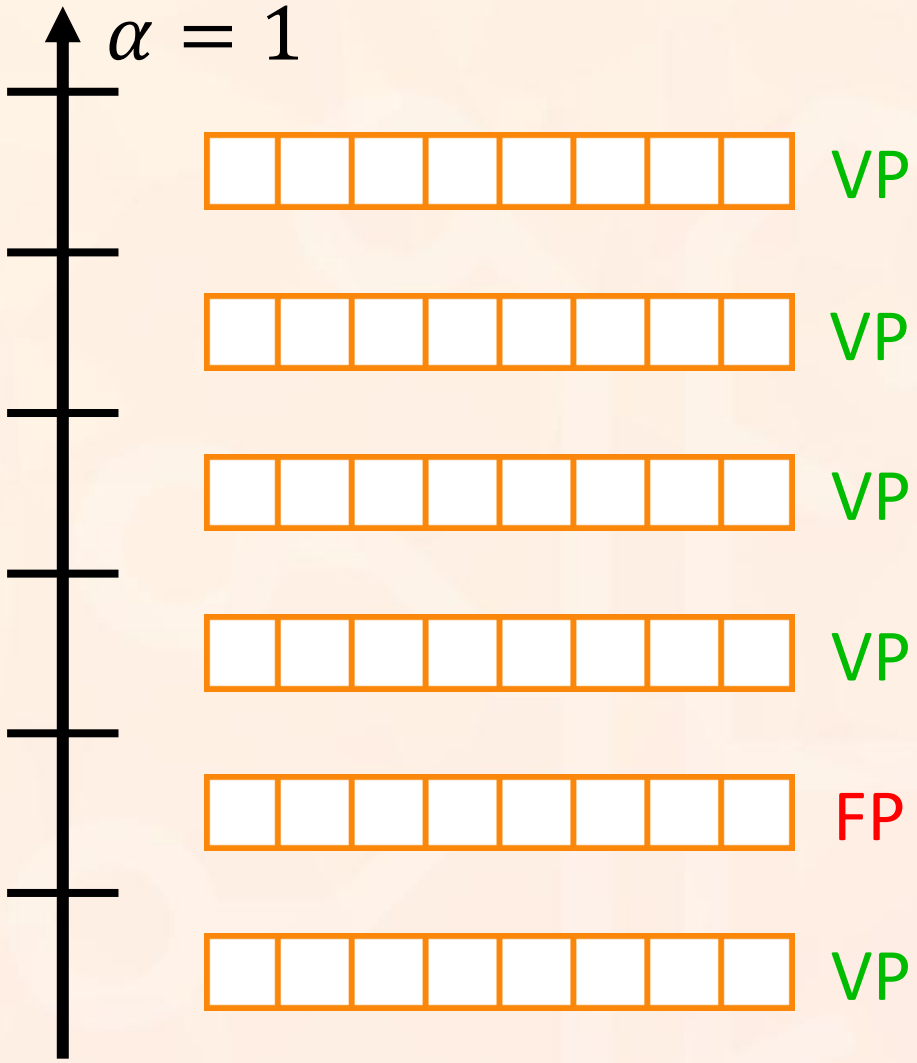
Average Precision (AP)

Pour tracer la courbe d'AP

- On ordonne les boîtes par score décroissant
- Pour $n \in [\![0; N_{boîtes}]\!]$, on calcule

$$P_n = \frac{VP_n}{VP_n + FP_n}$$

$$R_n = \frac{VP_n}{VP_n + FN_n} = \frac{VP_n}{N_{boîtes}}$$



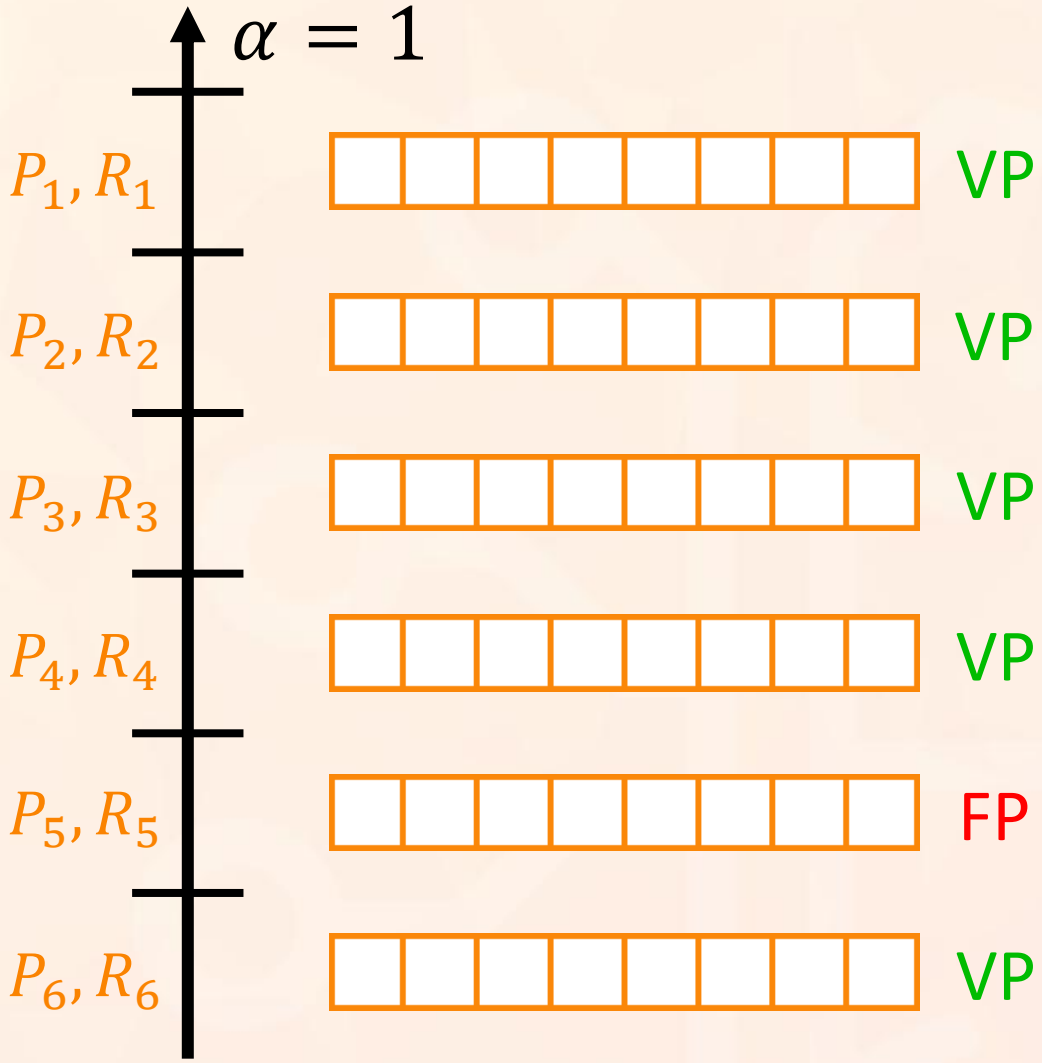
Average Precision (AP)

Pour tracer la courbe d'AP

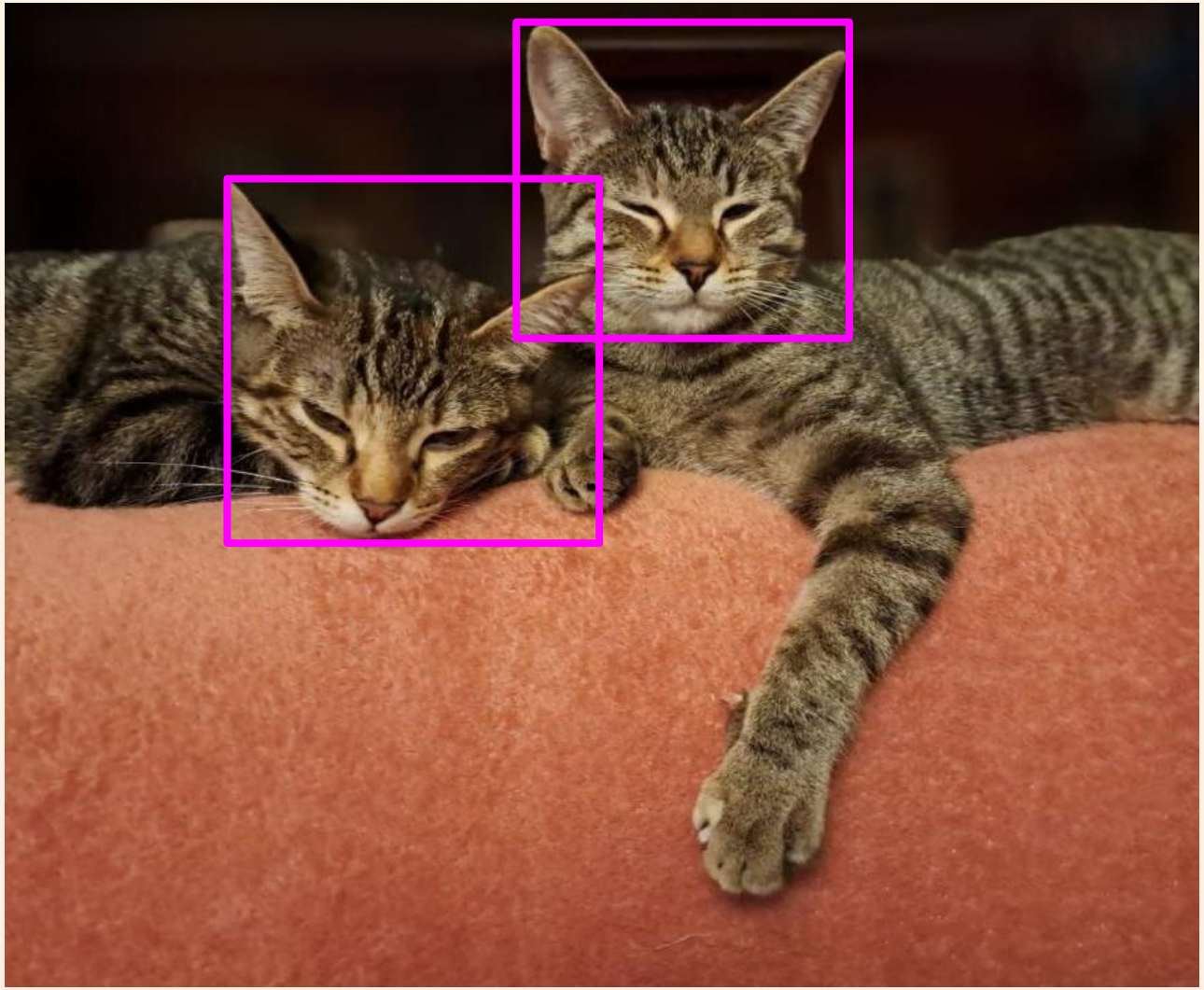
- On ordonne les boîtes par score décroissant
- Pour $n \in [\![0; N_{boîtes}]\!]$, on calcule

$$P_n = \frac{VP_n}{VP_n + FP_n}$$

$$R_n = \frac{VP_n}{VP_n + FN_n} = \frac{VP_n}{N_{boîtes}}$$



Average Precision (AP)

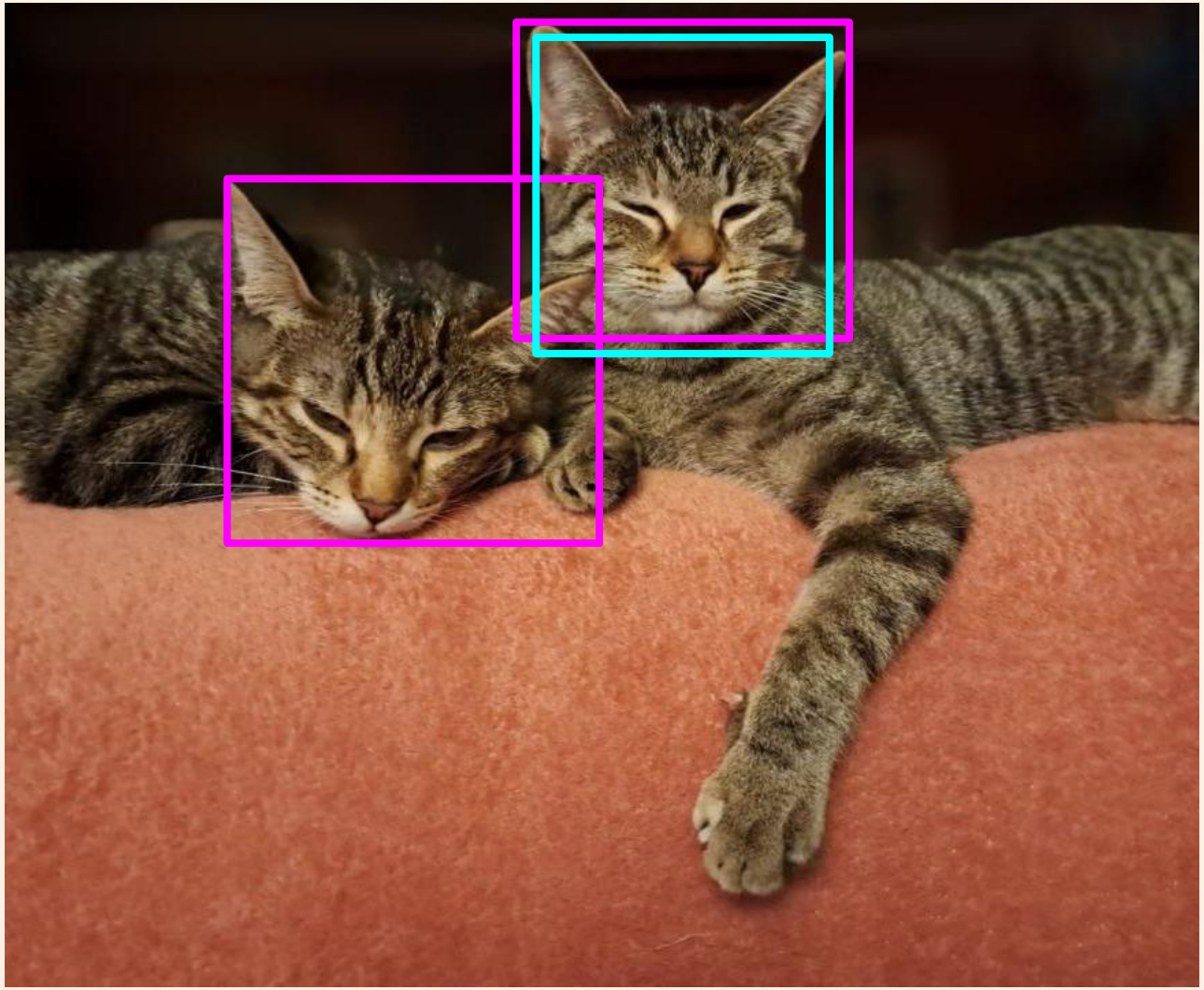


Pour $\alpha = 1$, on pose

$$P = \frac{VP}{VP + FP} = 1$$

$$R = \frac{VP}{VP + FN} = 0$$

Average Precision (AP)

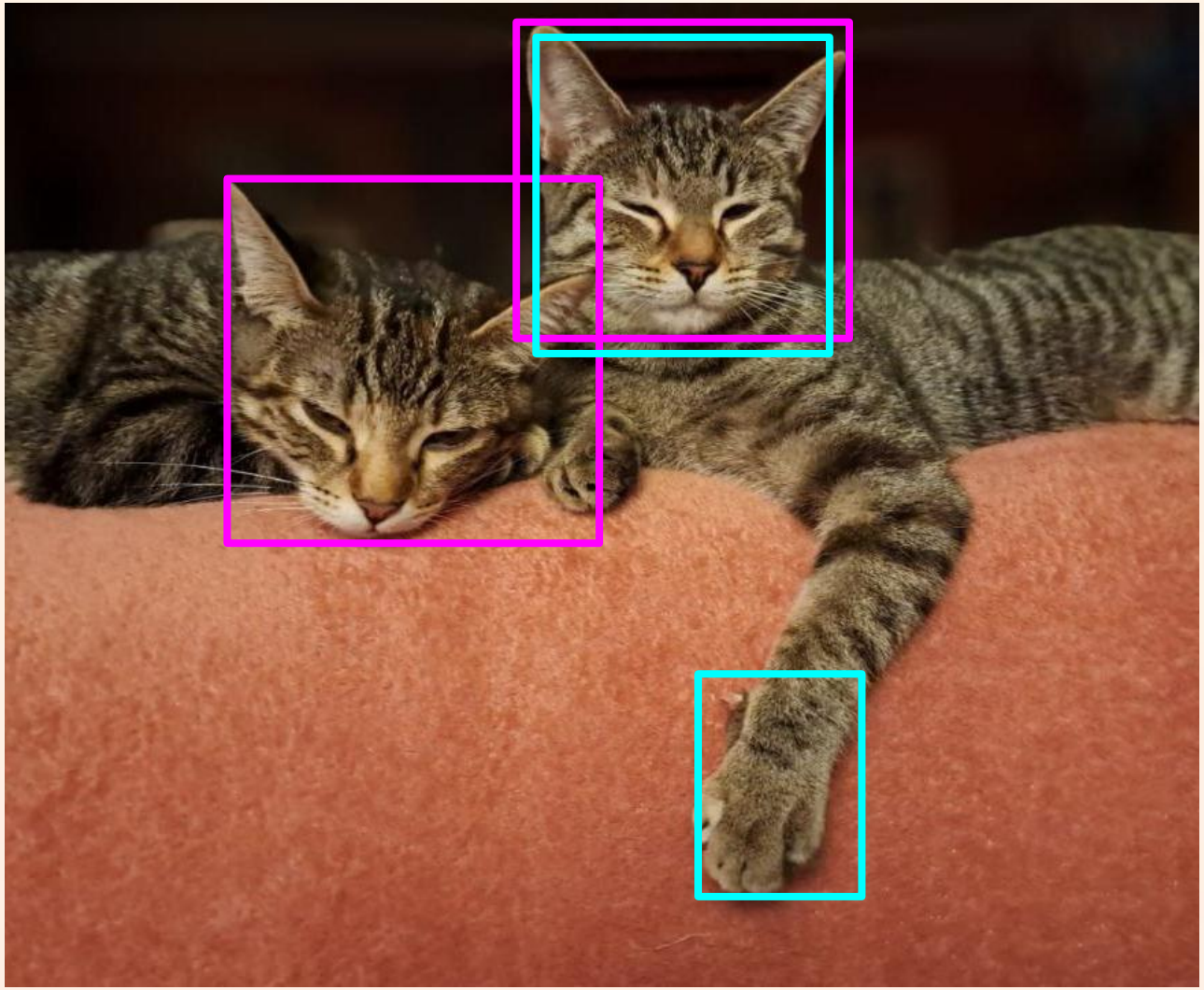


Si on ajoute une boîte légitime, alors $VP \nearrow$ et $FN \searrow$

$$P = \frac{VP}{VP + FP} \nearrow$$

$$R = \frac{VP}{VP + FN} \nearrow$$

Average Precision (AP)

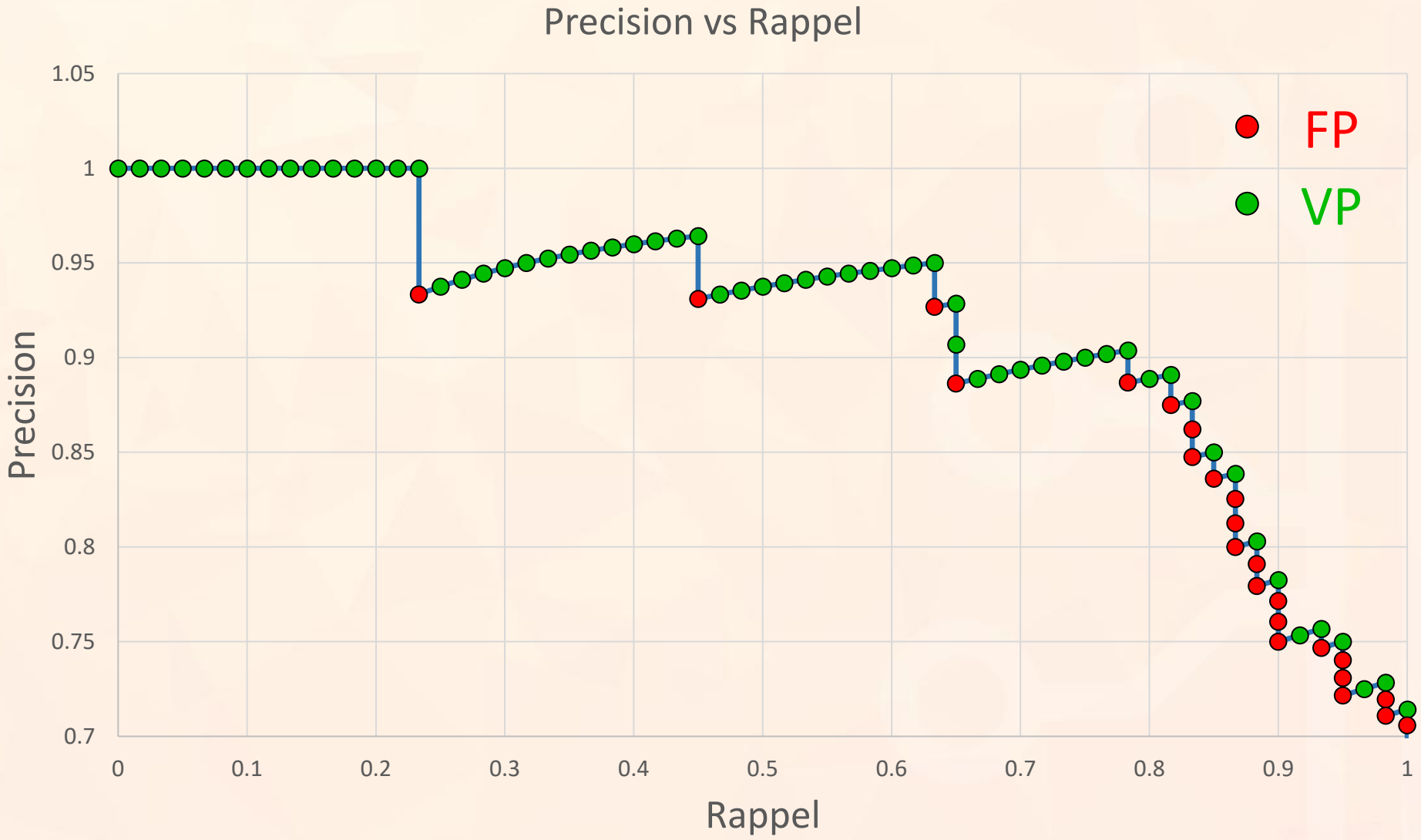


Si on ajoute une mauvaise boîte, alors $FP \nearrow$

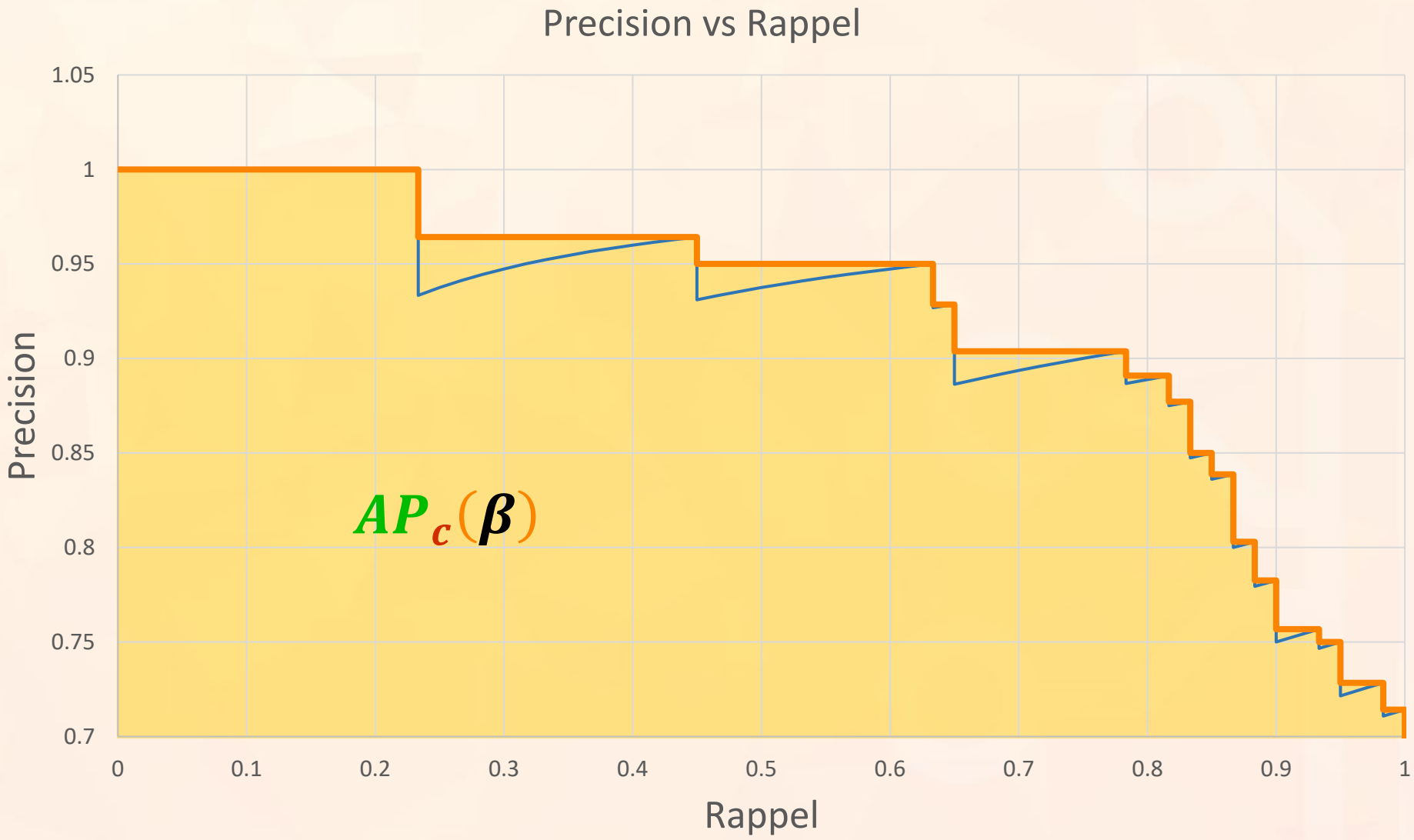
$$P = \frac{VP}{VP + FP} \searrow$$

$$R = \frac{VP}{VP + FN} \rightarrow$$

Average Precision (AP)



Average Precision (AP)



Mean Average Precision (mAP)

La métrique finale est obtenue en faisant la moyenne de l'AP de chaque classe :

$$\boldsymbol{mAP}(\beta) = \frac{1}{C}\sum_{c=1}^{C} \boldsymbol{AP}_c(\beta)$$

$$\boldsymbol{mAP} = \frac{1}{\#B}\sum_{\beta \in B} \boldsymbol{mAP}(\beta) = \frac{1}{C \times \#B}\sum_{\beta \in B}\sum_{c=1}^{C} \boldsymbol{AP}_c(\beta)$$

$$B_{coco} = \{\beta = 0.5 + 0.05k, \forall k \geq 0 \,/\, \beta < 1\}$$

Mean Average Precision (mAP)

La métrique finale est obtenue en faisant la moyenne de l'AP de chaque classe :

Ordre de grandeur
$\boldsymbol{mAP} \simeq 0.4 \Rightarrow$ modèle pas mal
$\boldsymbol{mAP} \simeq 0.5 \Rightarrow$ très bon modèle

$$\boldsymbol{mAP} = \frac{1}{\#B} \sum_{\beta \in B} \boldsymbol{mAP}(\beta) = \frac{1}{C \times \#B} \sum_{\beta \in B} \sum_{c=1}^{C} \boldsymbol{AP}_c(\beta)$$

$$B_{coco} = \{\beta = 0.5 + 0.05k, \forall k \geq 0 \,/\, \beta < 1\}$$

Questions?

Bonus

CNN pur

Conséquences du CNN pur