YOLO : Détection d'Objets en Temps Réel

Zaouche Djaouida

Cy-tech

November 26, 2024

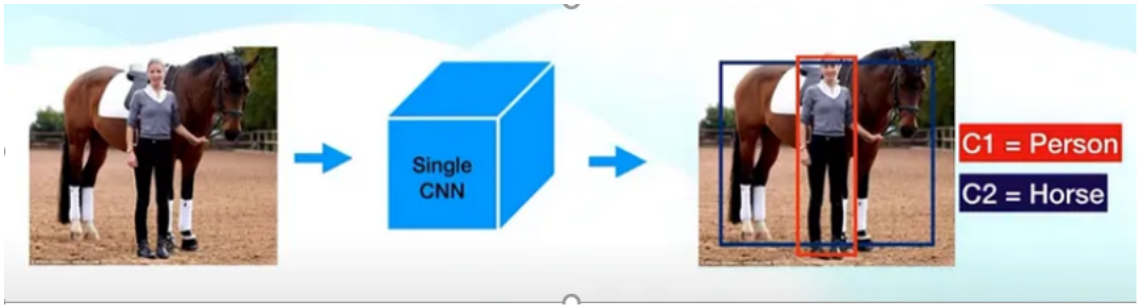
YOLO

- YOLO, acronyme de "You Only Look Once", est une méthode de détection d'objets en temps réel dans des images ou des vidéos.
- L'auteur principal qui a introduit YOLO est Joseph Redmon. Il a publié le premier article sur YOLO en 2016, intitulé : "You Only Look Once: Unified, Real-Time Object Detection".
- D'autres extensions de YOLO ont été proposé, YOLO4 par Alexey Bochkovskiy et d'autres et YOLO5 (développé par l'équipe Ultralytics).

Localisation et détection des objets

- YOLO est un modèle d'apprentissage profond qui utilise un seul réseau neuronal pour détecter des objets dans une image ou un flux vidéo. Il divise l'image en une grille de cellules et prédit les boîtes de délimitation, les scores de confiance et les probabilités de classe pour chaque cellule.

- Les boîtes de délimitation définissent l'emplacement et la taille des objets, tandis que les scores de confiance représentent la confiance en la présence d'un objet dans cette cellule. Les probabilités de classe indiquent la probabilité que l'objet appartienne à une classe spécifique, telle qu'une personne ou une voiture.



Avantages de YOLO (1)

- YOLO (You Only Look Once) est conçu pour être rapide et efficace, ce qui le rend idéal pour les applications en temps réel. Il peut détecter rapidement des objets dans des images ou des flux vidéo .

- Il est nettement plus rapide que les autres modèles de détection d'objets. Cette rapidité et cette efficacité sont essentielles pour des applications telles que la surveillance et la sécurité, où le contrôle en temps réel est primordial.

- YOLO est également très précis.Il est comparable à d'autres modèles de détection d'objets à la pointe de la technologie.

Avantages de YOLO (2)

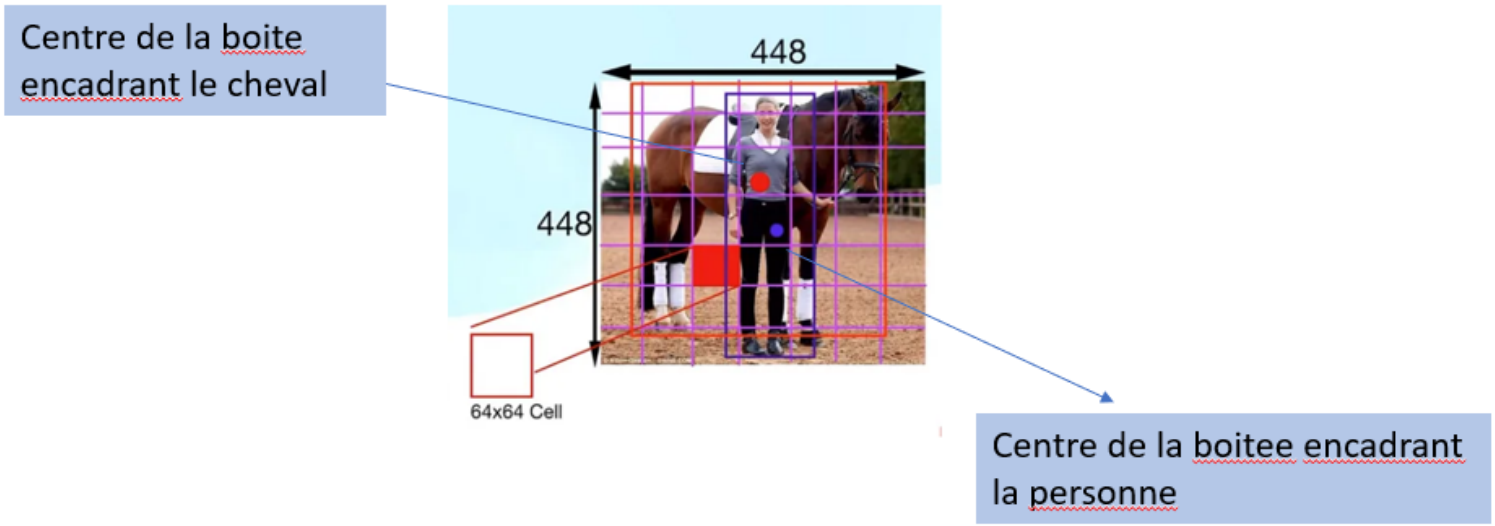
- YOLO est flexible et personnalisable, ce qui lui permet de s'adapter à diverses applications. Il peut détecter plusieurs objets simultanément et être entraîné sur des ensembles de données personnalisés, ce qui lui permet de reconnaître des objets ou des classes spécifiques.

- YOLO est constamment amélioré et mis à jour, de nouvelles versions et mises à jour étant régulièrement publiées. Cette amélioration continue lui permet de rester à la pointe de la technologie en matière de détection d'objets et de s'adapter aux nouveaux défis et aux nouvelles applications.

Division des images en grilles pour la détection d'objets

- YOLO divise l'image en une grille de cellules et prédit les boîtes de délimitation, les scores de confiance et les probabilités de classe pour chaque cellule.

- Une cellule de la grille où tombe le centre de d'un objet est responsable de la prédiction de cet objet.



Détection et utilisation des boîtes englobantes (1)

- Pour construire les données d'apprentissage, un objet d'une image donnée est associée à une boîte englobante. Celle-ci est définie à partir des coordonnées du coin supérieur gauche $(x_{\min}, y_{\min})$ et du coin inférieur droit $(x_{\max}, y_{\max})$. Cette boîte englobante est ensuite décrite par un vecteur $[x_c, y_c, w, h, c]$, qui est utilisé pour entraîner le modèle de détection :

  1. Le couple $(x_c,\ y_c)$ représente les coordonnées du centre de l'image.
  2. w et h donnent la largeur et la hauteur de l'image.
  3. c donne la classe de l'objet de la boite qui l'englobe.

Détection et utilisation des boîtes englobantes (2)

- Afin de mettre à l'échelle dans une plage comprise entre 0 et 1, x_c (resp. y_c) est divisé par la largeur (resp. la hauteur) de l'image.

- La largeur (resp. la hauteur) de la boîte englobante est normalisée en la divisant par la largeur (resp. hauteur) de l'image.

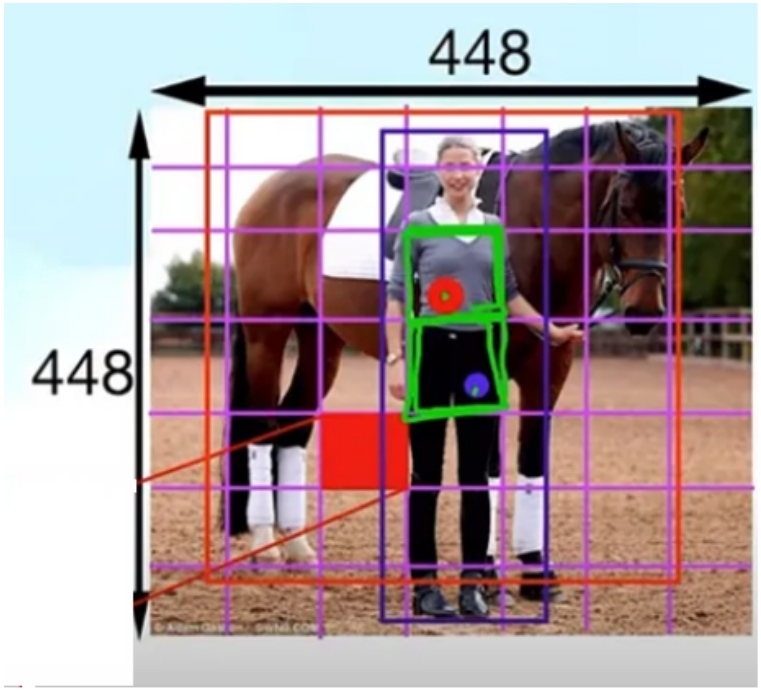
Conversion formula

- $center\_x = \frac{\frac{x_{min}+x_{max}}{2}}{width\ of\ the\ image}$
- $center\_y = \frac{\frac{y_{min}+y_{max}}{2}}{height\ of\ the\ image}$
- $w = \frac{x_{max}-x_{min}}{width\ of\ the\ image}$
- $h = \frac{y_{max}-y_{min}}{height\ of\ the\ image}$

Détection et utilisation des boîtes englobantes (3)

- Le processus de calcul des 5-uplets $(x_c,\ y_c,\ w,\ h,\ c)$ est répété pour tous les objets de chaque image des données d'apprentissage.

- Certaines cellules de la grille ne contiennent pas d'objet. Dans la figure ci-dessous, seules les cellules vertes sont responsables de détecter les deux objets de l'image.



Détection et utilisation des boîtes englobantes (4)

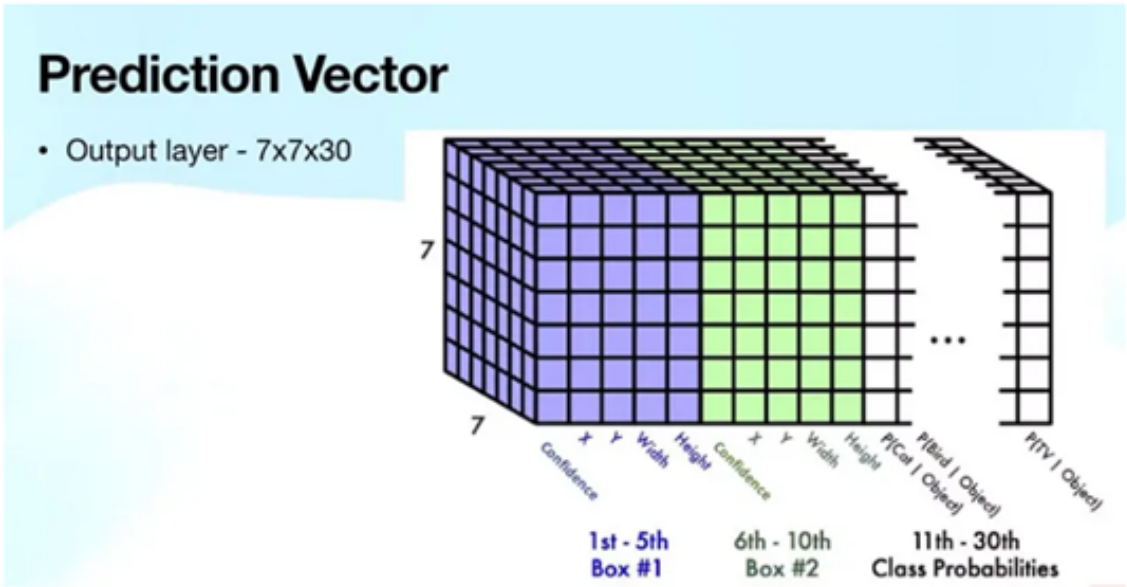
- Les données d'entraînement décrivent le contenu des cellules de la grille d'une image sous le format :

$$(\Delta\hat{x},\ \Delta\hat{y},\ \Delta\hat{w},\ \Delta\hat{h},\ \hat{c}),\ (\hat{p}_1,\ \hat{p}_2,\ \ldots,\ \hat{p}_{20}).$$

  Ici, $\Delta\hat{x}$ (resp. $\Delta\hat{y}$) correspond à l'abscisse (resp. l'ordonnée) réelle et normalisée du centre de l'objet. $\Delta\hat{w}$ (resp. $\Delta\hat{h}$) indique la largeur réelle (resp. hauteur) normalisée de la boîte englobante. La valeur $\hat{c}$ représente le score de confiance réel pour que la boite décrite par le 5-uplet contienne un objet.
- Dans le modèle de prédiction de YOLO, la sortie est représentée par $(x^*,\ y^*,\ w^*,\ h^*,\ c*)$ $(p_1^*,\ p_2^*\ \ldots,\ p_{20}^*)$. La signification des données est la même que pour le point précédent, sauf qu'il s'agit de données prédites.

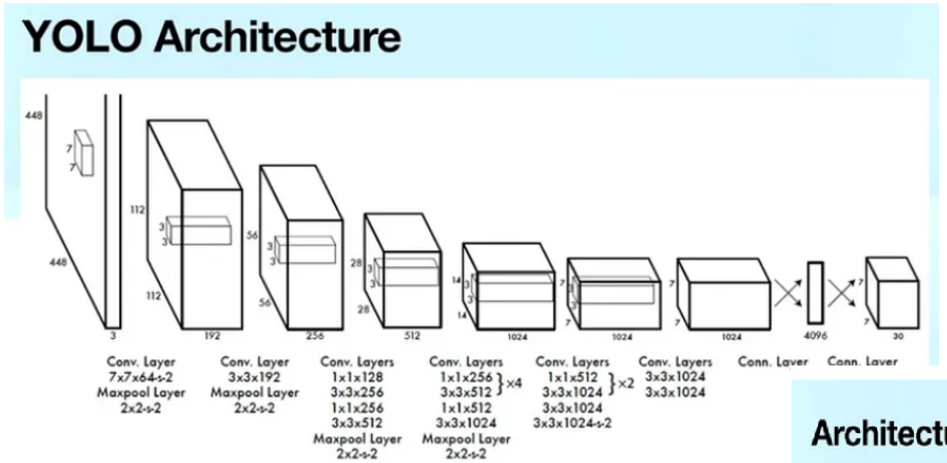
Prédictions des boîtes englobantes

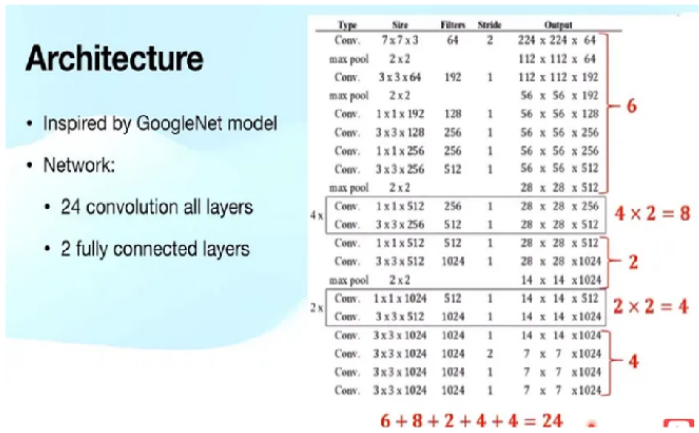
- Du point de vue du modèle, chaque cellule de la grille génère des prédictions pour deux boîtes englobantes, et pour chaque boîte englobante, le modèle fournit des valeurs $(x^*,\ y^*,\ w^*,\ h^*,\ c*)$ $(p_1^*,\ p_2^*\ \ldots,\ p_{20}^*)$. En outre, le modèle estime les probabilités de classe pour les 20 classes de l'ensemble de données, représentées par des vecteurs à 20 dimensions, où chaque élément se situe dans l'intervalle (0, 1). Par conséquent, le nombre total de paramètres associés à la prédiction de chaque cellule de la grille est de $(2\ *\ 5) + 20 = 30$.



Architecture de YOLO (1)

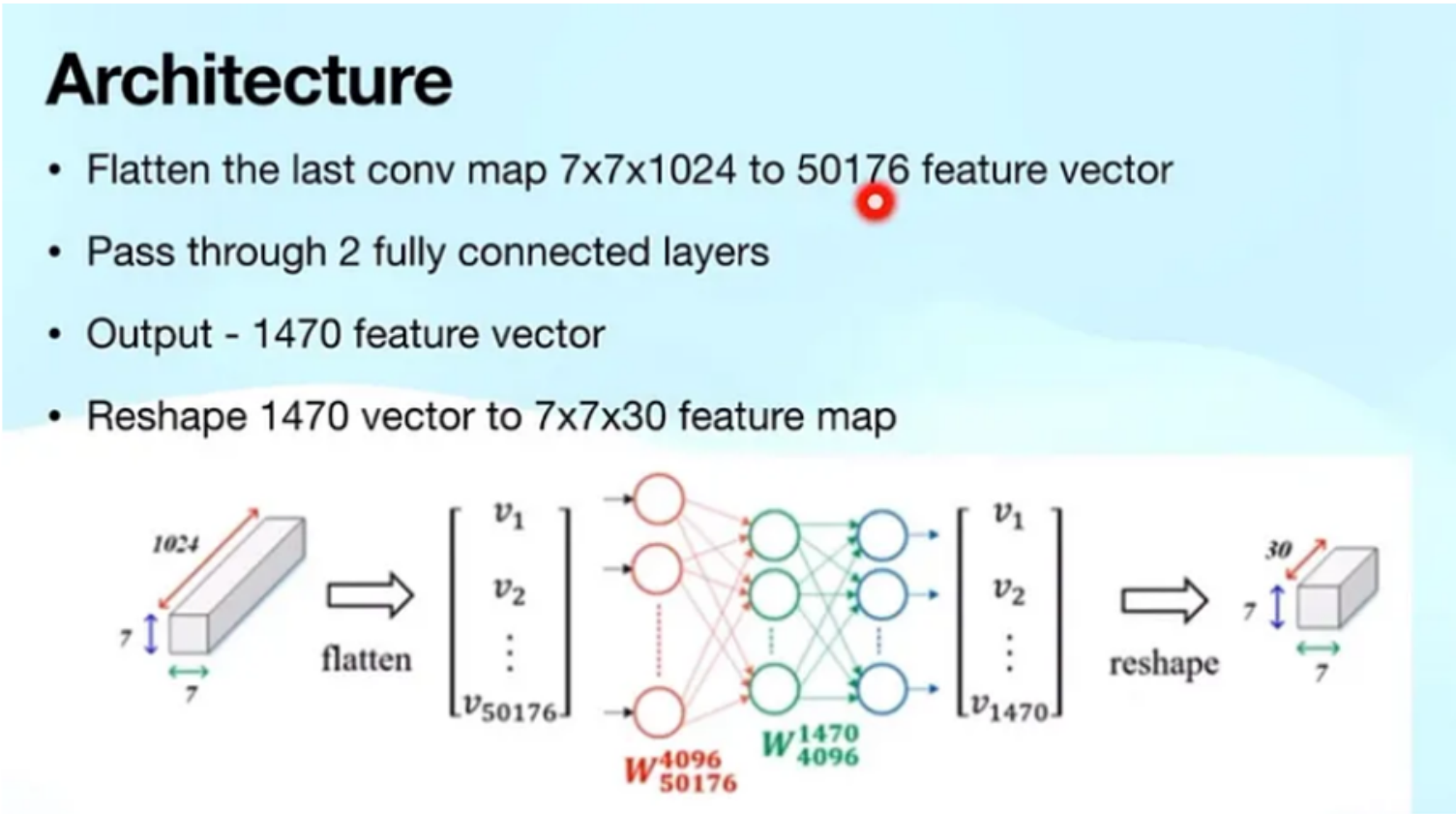
- YOLO est un réseau convolutionnel.





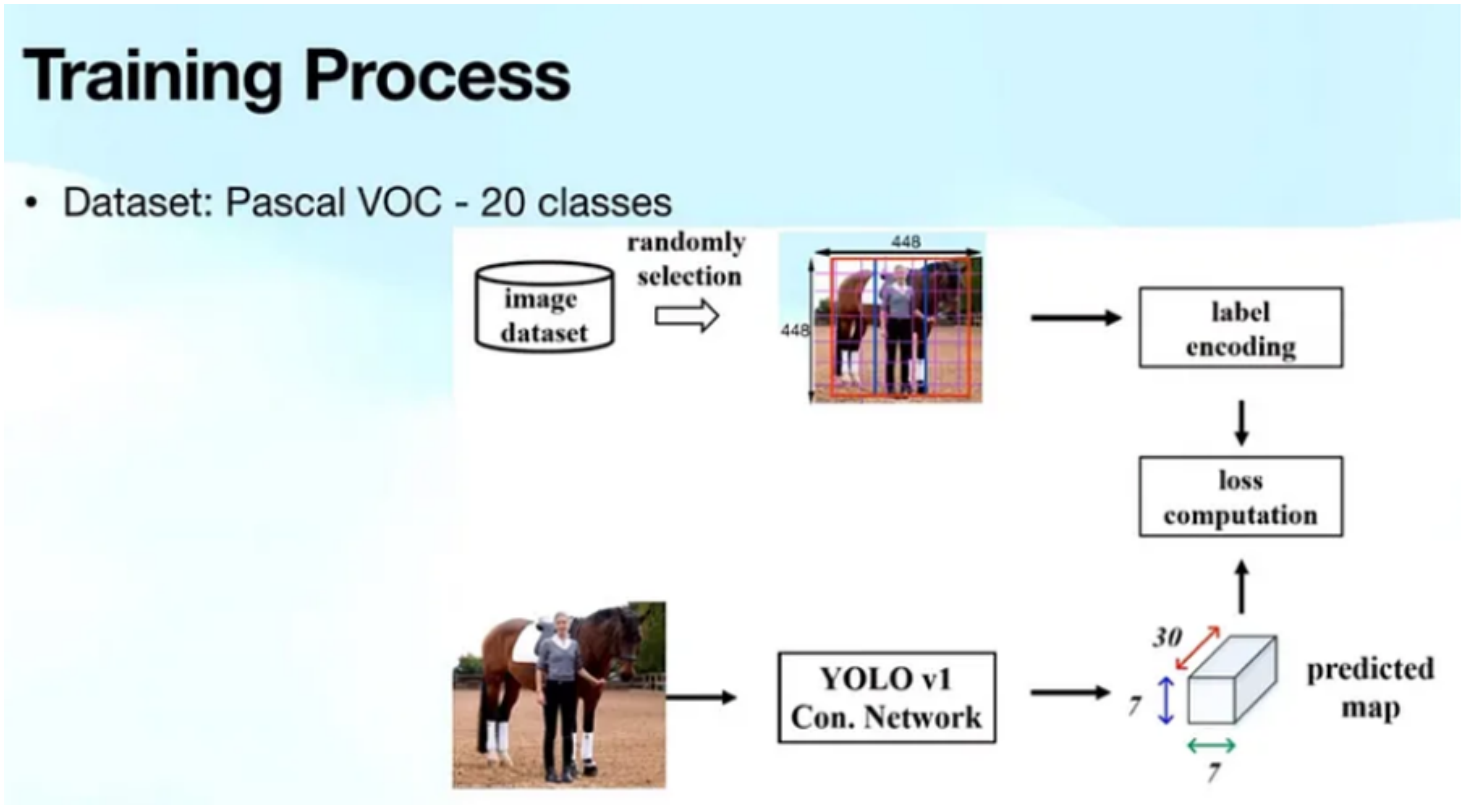
Architecture de YOLO (2)

- Les dernières couche du réseau convolutionnel.



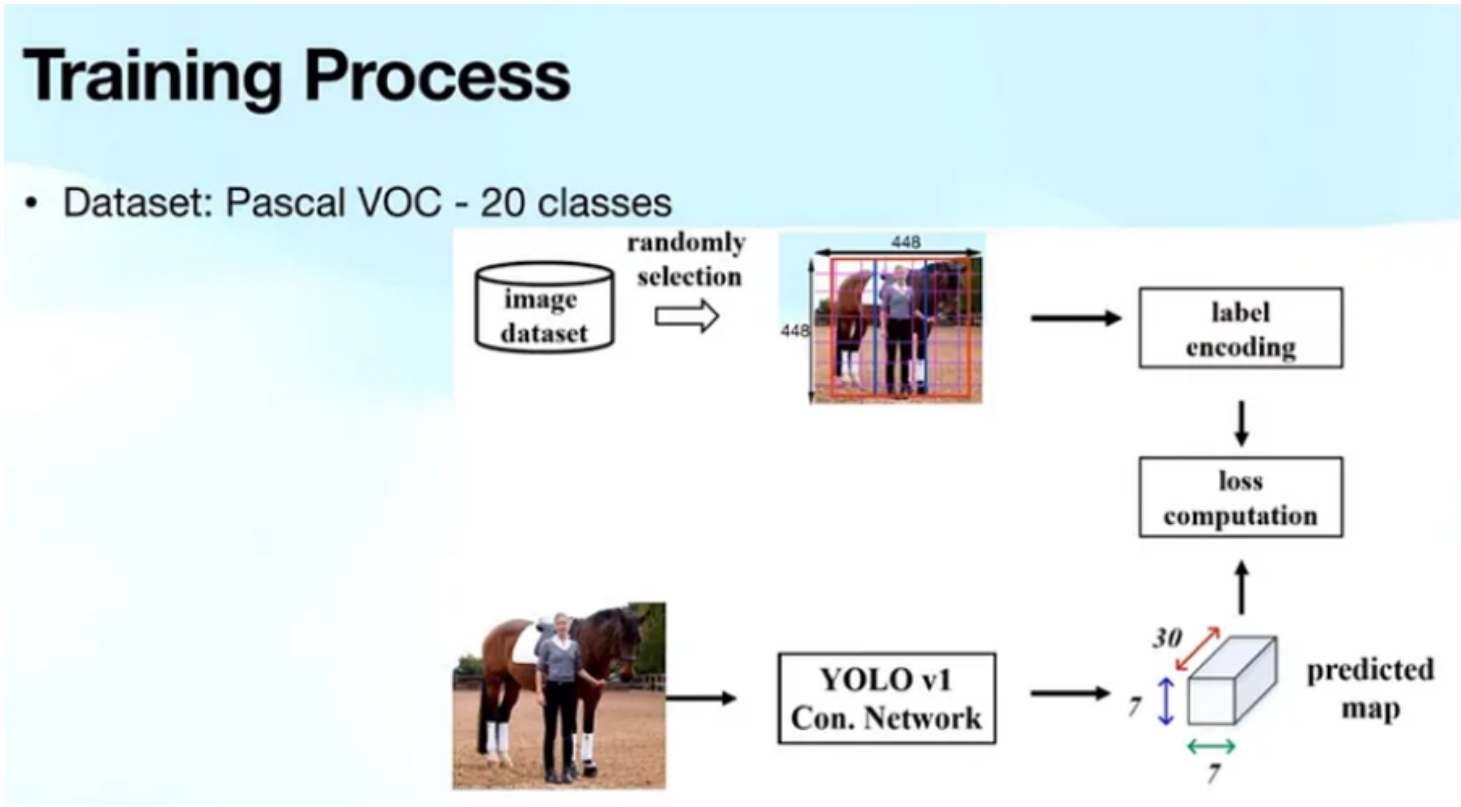
Phase d'entraînement (1)

- La première version de YOLO (YOLO v1) a d'abord été entrainée sur l'ensemble de données ImageNet (224x224) afin d'extraire des caractéristiques de base, avant d'être ajustée sur l'ensemble de données Pascal VOC (448x448).



Phase d'entraînement (2)

- Deux types de cellules de la grille existent : la première contenant un objet et la seconde n'en contenant aucun. La fonction de perte est la somme des deux types de pertes.



Fonction de perte (1)

Loss Function

- Loss L is the sum of losses over all grid cells SxS.
- Put more importance on grid cells that contain objects
- Decrease the importance of grid cells having no objects

$$L = \sum_{i=1}^{S^2} 1_i^{obj} \times L_{i,obj} + \lambda_{no\_obj} \sum_{i=1}^{S^2} 1_i^{no\_obj} \times L_{i,no\_obj}$$

$1_i^{obj} = 1$ if i^{th} grid is **object anchor**

$1_i^{no\_obj} = 1$ if i^{th} grid is **no-object anchor**

Fonction de perte (2)

Loss for object cells

- Loss = Objectness loss + classification loss + Box Regression loss
- Put more weightage on box parameters

$$L_{i,obj} = \lambda_{coord} \times L_{i,obj}^{box} + L_{i,obj}^{conf} + L_{i,obj}^{cls}$$

$$= 5$$

Bounding box loss

- Sum of squared errors on predicted box parameters and ground truth labels

$$L_{i,obj}^{box} = \left(\Delta x_i^* - \Delta \hat{x}_i\right)^2 + \left(\Delta y_i^* - \Delta \hat{y}_i\right)^2 + \left(\sqrt{\Delta w_i^*} - \sqrt{\Delta \hat{w}_i}\right)^2 + \left(\sqrt{\Delta h_i^*} - \sqrt{\Delta h_i}\right)^2$$

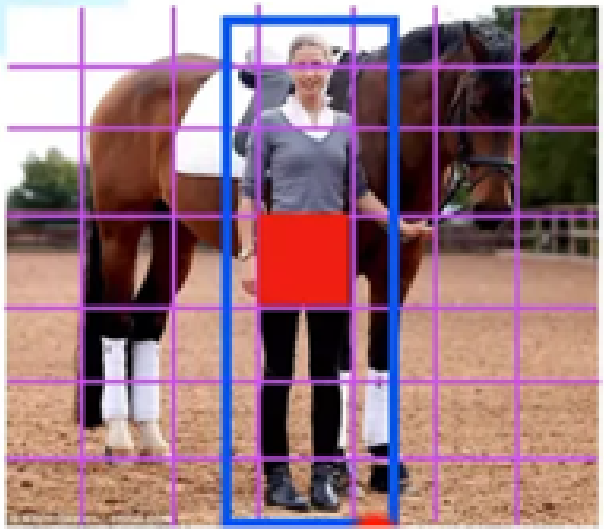
- $(\Delta \hat{x}_i, \Delta \hat{y}_i, \Delta \hat{w}_i, \hat{h}_i)$: ground-truth box
- $(\Delta x_i^*, \Delta y_i^*, \Delta w_i^*, \Delta h_i^*)$: **responsible** predicted box that has the largest IoU with ground-truth box

Fonction de perte (3)

Objectness Confidence Loss

- Squared error between the predicted confidence and encoded label confidence

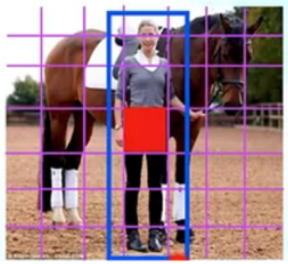
$$L_{i,obj}^{conf} = (c_i^* - \hat{c}_i)^2 = (0.9 - 1.0)^2$$



Fonction de perte (4)

Classification Loss

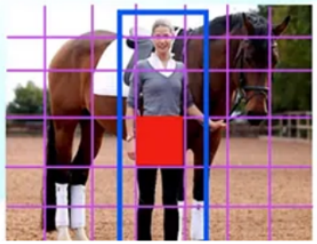
- Sum of squared errors over all class probabilities



$$L_{i,obj}^{cls} = \sum_{c=1}^{20} \left(p_{i,c} - \hat{p}_{i,c}\right)^2$$

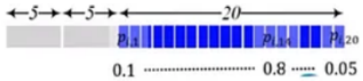
Classification Loss

- Sum of squared errors over all class probabilities



$$L_{i,obj}^{cls} = (0.1 - 0.0)^2 + \cdots + (0.8 - 1.0)^2 + \cdots + (0.05 - 0.0)^2$$

$\hat{p}_{i,14}$



$$\hat{p}_{i,14=\text{person}} = 1.0$$

Fonction de perte (5)

Total Loss

$$L = \lambda_{coord} \times \sum_{i=1}^{S^2} 1_i^{obj} \times \begin{pmatrix} (\Delta x_i^* - \Delta \hat{x}_i)^2 + (\Delta y_i^* - \Delta \hat{y}_i)^2 + \\ \left(\sqrt{\Delta w_i^*} - \sqrt{\Delta \hat{w}_i}\right)^2 + \left(\sqrt{\Delta h_i^*} - \sqrt{\Delta h_i}\right)^2 \end{pmatrix}$$

$$+ \sum_{i=1}^{S^2} 1_i^{obj} \times (c_i^* - \hat{c}_i)^2 + \sum_{i=1}^{S^2} 1_i^{obj} \times \sum_{c=1}^{20} (p_{i,c} - \hat{p}_{i,c})^2$$

$$+ \lambda_{no\_obj} \sum_{i=1}^{S^2} 1_i^{no\_obj} \times \sum_{j=1}^{B} (c_{i,j} - \hat{c}_{i,j})^2$$

Conclusion

- Nombre fixe de prédictions : YOLOv1 prédit un nombre fixe de boîtes englobantes par cellule de la grille (dans notre cas, 49 prédictions au total sont possibles car nous avons 49 cellules de la grille), ce qui peut ne pas être optimal pour des images dont le nombre d'objets varie.
- localisation : YOLOv1 a eu du mal à déterminer avec précision où se trouvaient les objets dans les images et quelle était leur taille. Il dessinait souvent des boîtes qui ne s'adaptaient pas bien aux objets, surtout si ceux-ci étaient petits ou proches les uns des autres. Cela s'explique par le fait que YOLOv1 ne pouvait pas gérer efficacement les différentes tailles d'objets et que ses images perdaient certains détails fins, ce qui rendait ses estimations moins précises.
- Limité dans la détection des petits objets : YOLOv1 a eu du mal à détecter et à localiser avec précision les petits objets dans les images, car son approche à échelle unique n'était pas adaptée à la gestion d'objets de tailles différentes.

Références Bibliographiques

1. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, https://arxiv.org/abs/1506.02640.

2. Blog, "Concept of YOLOv1:The Evolution of Real-Time Object Detection", https://medium.com/@sachinsoni600517/concept-of-yolov1-the-evolution-of-real-time-object-detection-d773770ef773.