Rapport - YOLO

Romain JAMINET

25 février 2019

Table des matières

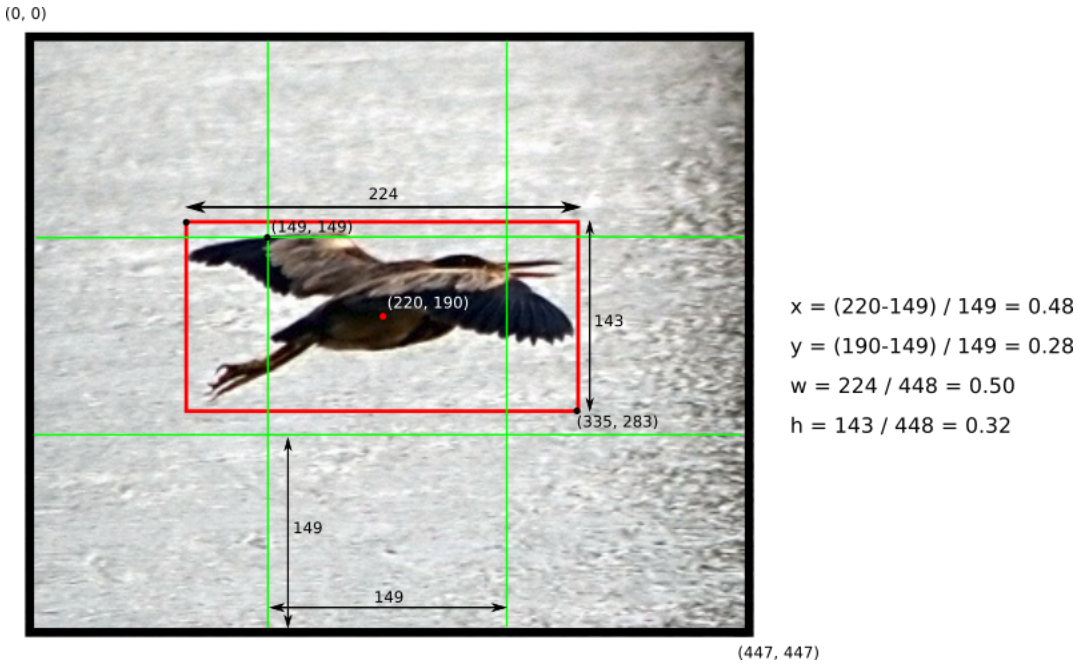
1 Qu'est-ce que "YOLO"

You Only Look Once est un algorithme de classification et détection d'objet en temps réel, il s'agit de l'un des meilleurs algorithme de reconnaissance d'images car le temps de la reconnaissance et de la classification d'une image complète ne mets que quelques milisecondes (22 ms avec une NVIDIA Titan X).

source : https ://pjreddie.com/darknet/yolo/

2 Comment cela fonctionne ?

Les systèmes de détection antérieurs utilisent des classificateurs ou des localisateurs pour effectuer leur détection. Ils appliquent un modèle à une image à différents endroits et avec des tailles différentes. Si il y a un score de ressemblance assez élevé à un endroit de l'image, alors l'algorithme considère qu'il y a une détection.
YOLO utilise quelque chose de totalement différent, ils utilisent un seul réseau de neurones pour une l'image entière. Le réseau de neurones divise l'image en région et prédit des "boites" qui englobent les objets et leur donne une probabilité.
L'image d'entré est divisé en une grille de cellules, pour chaque objet présent sur l'image, une cellule est responsable de sa détection, il s'agit de la cellule qui est au centre de l'objet. La boite entourant l'objet a 5 composants, x et y les coordonnées de la cellule centrale, w et h les dimensions de la boite, et un coefficient de confiance. Toutes ces données sont normalisés entre 0 et 1. S'il n'y a pas d'objet, alors le coefficient de confiance sera autour de 0. Il y a également plusieurs prédictions de classe pour une boite.
Exemple :



Le réseau de neuronne traitant l'image est "Darknet", il s'agit d'un réseau neuronal convolutif open-source écrit en C et utilise Cuda (de NVIDIA), il peut utiliser la puissance du GPU et du CPU. Le réseau de neuronnes comporte 24 couches convolutives suivies de 2 couches entièrement connectées. La sortie finale du réseau est un tenseur 7 x 7 x 3 contenant les prédictions.

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

source : https ://pjreddie.com/media/files/papers/YOLO9000.pdf

3 L'entrainement

Les développeurs ont décrit l'entrainement de la manière suivante :

1-Il y a eu un préentrainement des 20 premières couches en utilisant 1000 images de taille 224x224 durant une semaine.
2-Il y a eu le même entrainement mais d'une taille 448x448.
3-Puis enfin, un entrainement avec des images aléatoires, en changeant les dimensions, la saturation et le contraste.

source : https ://pjreddie.com/media/files/papers/yolo-1.pdf

4 Les limites

YOLO impose de fortes contraintes spaciales à cause du système de "boite", s'il y a plusieurs objets ayant un même centre, alors il n'y en aura qu'un seul qui sera détecté. Il y a également parfois un problème de localisation des objets.

source : https ://pjreddie.com/media/files/papers/yolo-1.pdf