

Reconnaissance d'espèces animales avec l'algorithme YOLO

Hayat ANKOUR
hayat.ankour.1@ens.etsmtl.ca



Le génie pour l'industrie

Génie de la Production Automatisée - École de Technologies Supérieures

Introduction

- L'identification d'espèce est un défi technologique complexe. De nos jours, ce sont des experts qui identifient et recensent les espèces. Cependant, cela prend beaucoup de temps, et les experts ne peuvent pas effectuer leur travail à grande échelle. De plus, c'est un travail visuel et manuel fastidieux et fatigant. Une des solutions est d'assister l'expert par l'utilisation d'algorithmes. Donnée une image, comment identifier l'espèce des animaux présents sur l'image?
- Les méthodes les plus utilisées sont les CNN avec prétraitement des images afin d'obtenir les régions d'intérêts.
- Le projet se découpe donc en deux parties :
 1. Comment déterminer efficacement les régions d'intérêt des différents animaux dans une image?
 2. Comment correctement classifier ces images ?

Définition

- **CNN (Réseau Neuronal Convolutif)** : Réseau artificiel composé de cluster couche de convolution permettant la classification de caractéristiques dans une image.

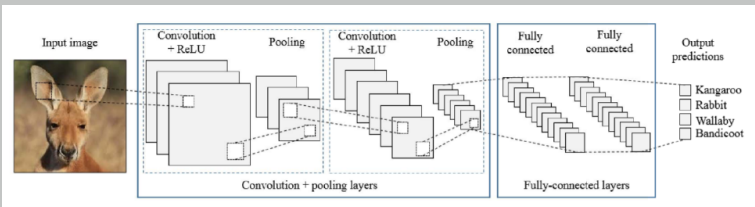


Figure 1: Illustration d'une architecture typique d'un réseau neuronal convolutif [1]

- **YOLO (You Only Look Once)** : Algorithme comportant une seule couche de CNN et qui permet de déterminer les bounding boxes des objets présents sur une image en même temps que leur classe avec le CNN. Le tout n'étant effectué qu'une fois.

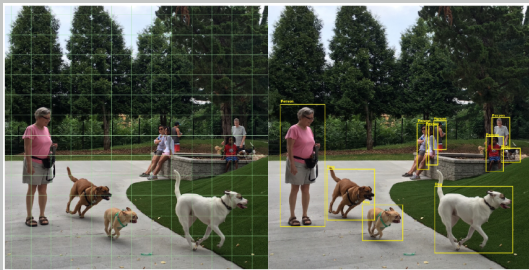


Figure 2: Division de l'image par l'algorithme YOLO et sa sortie

Méthodologie

- **Database** : la database utilisée est la Pascal VOC, qui contient 6 classes d'animaux (chat, chien, oiseau, vache, cheval et mouton).
- **Récupération des caractéristiques** : D'une part, on va appliquer la technique des histogrammes de gradients orientés afin de récupérer les caractéristiques présentes sur une image. Ensuite, on pourra extraire les caractéristiques des images à l'aide par exemple de la recherche sélective.

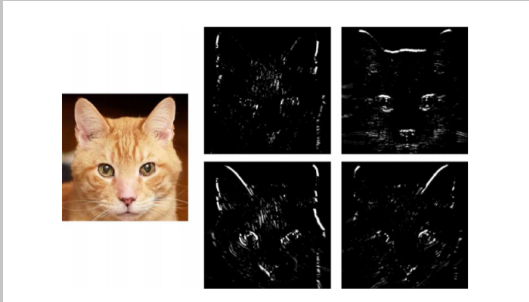


Figure 3: Division de l'image par l'algorithme YOLO et sa sortie [2]

- **Application du YOLO** :
 1. La première partie de l'algorithme récupère les caractéristiques et détermine les bounding boxes où un objet a le plus de probabilité d'y être.
 2. La seconde partie faisant appel au CNN classe les objets des bounding boxes.

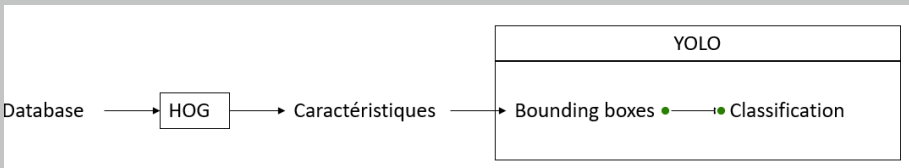


Figure 4: Récapitulatif des étapes nécessaires pour l'application de l'algorithme YOLO

Résultat

- Le résultat obtenu par l'algorithme YOLO sur la database Pascal VOC est le suivant [3]:

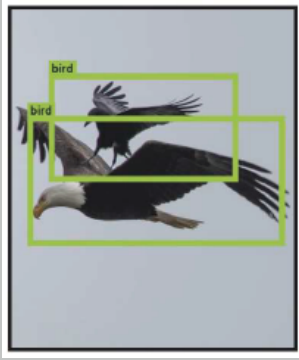


Figure 5: Résultat qualitatif obtenue par YOLO

- En termes de temps de calcul et de performance, YOLO est beaucoup plus rapide que les autres algorithmes existants tel que Fast R-CNN, un algorithme qui utilise des CNN comme base.

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	155
YOLO	2007+2012	63.4	45
Less Than Real-Time			
Fastest DPM [37]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Figure 6: Comparaison du temps de calcul et de la performance entre YOLO et d'autres algorithmes utilisant CNN

- En terme de précision, YOLO est légèrement en dessous. En effet, malgré sa vitesse de calcul, le taux d'erreur obtenue est supérieur au algorithme de CNN comme Fast R-CNN.

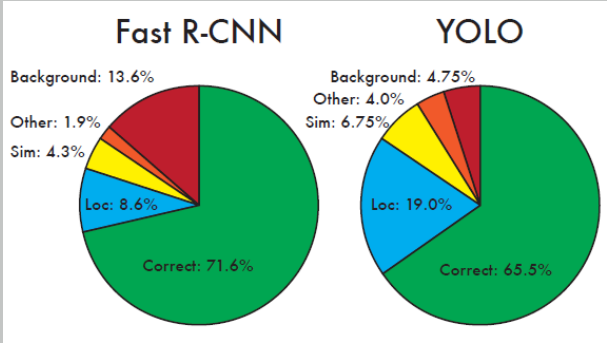


Figure 7: Comparaison de précision des résultats obtenues par YOLO et Fast R-CNN

Conclusions

- L'algorithme YOLO est très rapide mais il faut faire attention, à son taux d'erreur qui est légèrement supérieur aux algorithmes classiques de CNN. En effet, on ne peut pas tout avoir, la précision est donc légèrement négligée pour avoir une vitesse de calcul plus rapide.
- Pour la suite du projet, il faut que j'adapte le projet YOLO de Darknet que je vais utiliser à ma base de données. Il faut de plus que je l'adapte à l'algorithme SSD (un autre algorithme utilisant CNN) afin de faire une comparaison qualitative des performances de ces deux algorithmes.

Référence

- [1] Hung Nguyen, Sarah J MacLagan, Tu Dinh Nguyen, Thin Nguyen, Paul Flemons, Kylie Andrews, Euan G Ritchie, and Dinh Phung. Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. *In 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 40–49. IEEE, 2017.
- [2] Weiwei Zhang, Jian Sun, and Xiaoou Tang. From tiger to panda: animal head detection. *IEEE Transactions on Image Processing*, 20(6):1696–1708, 2011.
- [3] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.