

État de l'art : outils de détection d'objets par l'Intelligence Artificielle

Auteurs : Alexis MARIE, David MARCHÈS, Luca BANKOFSKI, Martial BROSTIN, Theo HELLER, Thomas MABILLE

1. INTRODUCTION

En 2023, l'OMS fait état de 1.19 millions de morts dû aux accidents à travers le monde.[1] Cela représente un peu plus de la moitié de la population de Paris la même année. L'objectif décrit par l'assemblée générale des Nations unies est de diviser le nombre de morts sur la route par deux d'ici 2030. Notre projet s'inscrit dans cette démarche en cherchant à détecter des situations dangereuses à l'aide de l'intelligence artificielle.

L'état de l'art portera sur l'étude des solutions d'intelligence artificielles existantes qui peuvent nous permettre de répondre à cette problématique de sécurité sur les routes.

Mots-clés : Yolo, CNN, RPN.

2. ETUDE DE L'EXISTANT

Yolo

Yolo est un outil populaire pour détecter des objets dans des images ou vidéos en une seule fois (détection et classification), open source, sans avoir besoin de plusieurs étapes ou analyses séparées. Il est basé sur une architecture de réseaux de neurones convolutifs (CNN pour *convolutional neural networks*). Ceux-ci sont adaptés pour le traitement d'images ou de vidéos.

Plusieurs versions de Yolo sont apparues chacune succédant à la précédente en apportant des améliorations sur l'architecture et introduisant de nouvelles fonctionnalités afin d'améliorer les performances.

Les algorithmes de détection d'objets en une seule fois comme Yolo sont efficaces sur le plan informatique, adaptés aux applications en temps réel et ne demandent pas beaucoup de ressources. Cependant, les algorithmes à deux coups sont plus gourmands en ressources mais plus adaptés que ceux à un coup (notamment pour les petits objets) si le paramètre précision est plus important que le paramètre détection en temps réel. [2][3]

Yolo nécessite un minimum de 8Go de GPU (notamment le GPU NVIDIA prenant en charge CUDA) pour des performances optimales, bien qu'il puisse fonctionner sur divers environnements voire certains périphériques qu'il comprenne du GPU ou du CPU.[4]

Le résultat que cet algorithme peut fournir est similaire à l'image ci-dessous :

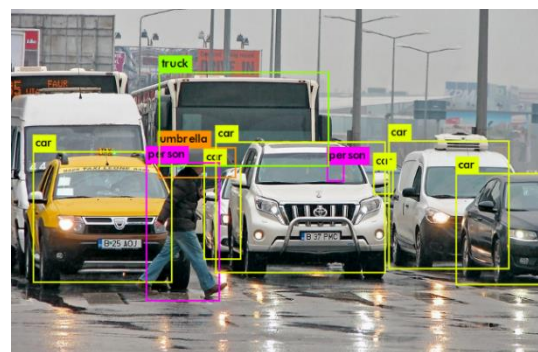


Figure 1 - Résultats des objets détectés par Yolo [5].

Mask R-CNN

Mask R-CNN est un modèle de deep learning utilisé pour la détection d'objets dans les images ou vidéos, et la segmentation d'instances. Il est open source et a la

particularité d'être capable de dessiner précisément les contours des objets détectés. Ce modèle de deep learning est composé de réseaux de neurones convolutifs, notamment d'une couche de *Region Proposal Network (RPN)* qui permet de générer les boîtes dans lesquelles les objets détectés sont placés sur l'image.

Il est également composé d'une couche *Region of Interest Align (RoIAlign)* qui permet d'obtenir les plus petits détails de chaque objet détecté. Il sert pour la couche suivante qui est la couche *Mask Head*, qui permet de segmenter les objets dans les images ou vidéos.

Le résultat que ce modèle peut produire est similaire à l'image ci-dessous. Il est cependant très coûteux en ressources en raison des différents travaux conséquents qu'il réalise.[6]

Voici ce que l'algorithme est capable de représenter :

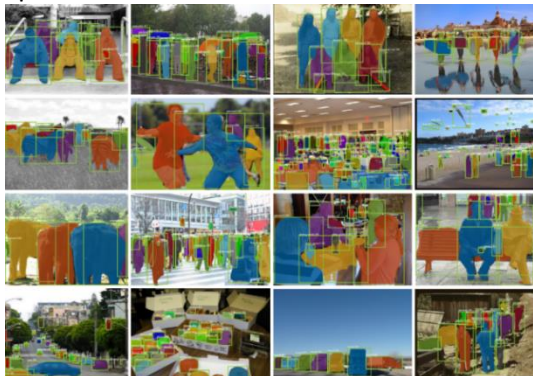


Figure 2 - Résultat des objets détectés par Mask R-CNN [6].

Il fait partie de la famille que tous les autres R-CNN : R-CNN, Fast-RCNN, Faster R-CNN. Cette famille d'algorithmes est performante mais ce sont des modèles de détection en plus d'un coup. Cela affecte donc leur vitesse. Il est par conséquent plus adapté selon si l'importance est la précision (détection de cancer ...) mais sera moins adapté dans le contexte de la détection dans le cadre de la voiture autonome. [7][8]

SSD (Single Shot Detector)

SSD est lui aussi un algorithme de détection d'objets en vision par ordinateur mais il n'est pas open source contrairement aux autres

algorithmes évoqués précédemment. Comme son nom l'indique, il détecte des objets en un seul coup, comme Yolo. L'algorithme utilise des réseaux de neurones convolutifs.

SSD reste plus lent que Yolo. En effet, il est recommandé d'utiliser Yolo plutôt que SSD si l'exactitude n'est pas cruciale mais que nous souhaitons être très rapide car SSD peut perdre énormément de vitesse si le modèle est grand.[9]

Voici un exemple de ce que l'algorithme pourra rendre lors de la détection des objets :



Figure 3 - Résultat de la détection de bâtiments avec SSD [10].

MobileNet

MobileNet est un algorithme de reconnaissance d'image en temps réel et embarqué, open source. Il a l'avantage d'être extrêmement léger, rapide, d'une précision redoutable, il n'est pas du tout gourmand en ressources et est conçu pour l'embarqué et smartphones.

Il est basé sur une couche de réseaux de neurones convolutifs.[11]

Voici une image du résultat de l'algorithme :

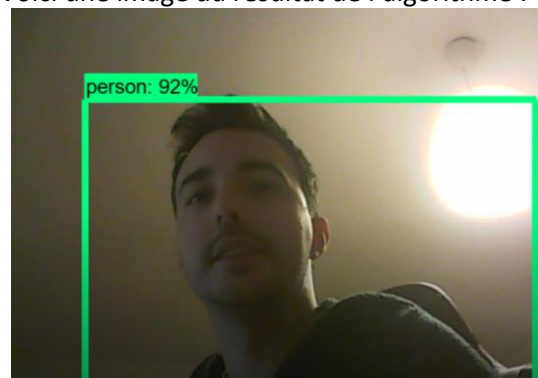


Figure 4 - Résultat de la détection avec MobileNet. [11]

MobileNet-SSD

MobileNet-SSD est la combinaison des algorithmes MobileNet et SSD. C'est un algorithme de détection d'objets en temps réel open source. Dans l'association de ces deux algorithmes, chacun a un rôle particulier : SSD permet de faire de la détection d'objets sur l'image, mais il ne les classe pas, il les détecte et les met dans des boîtes, et c'est MobileNet qui s'occupe de classer les objets détectés par SSD.

De ce fait l'algorithme est très puissant et utile.[12]

Voici un exemple du rendu de l'algorithme :

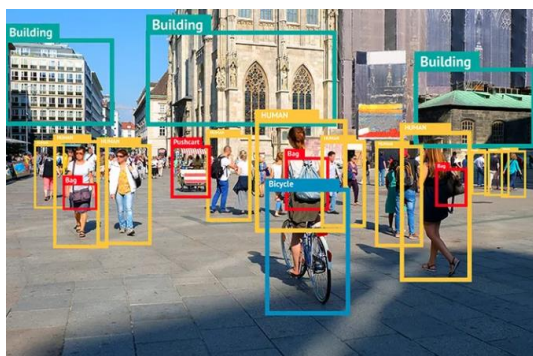


Figure 5 - Résultat de détection d'objets avec MobileNet-SSD [13].

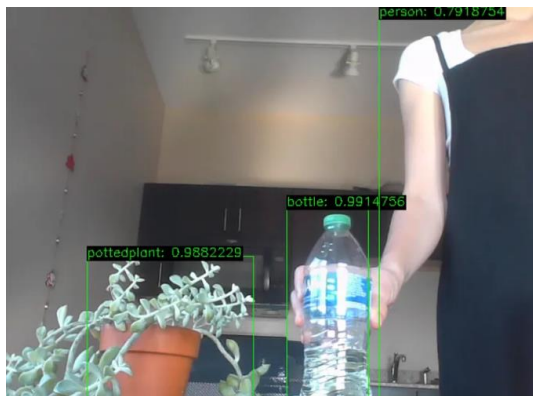


Figure 6 - Résultat de détection d'objets avec MobileNet-SSD. [13]

3. REFERENCES

[1] Organisation mondiale de la santé. 2023. « Road traffic injuries », OMS. Retrieved January, 2024, from <https://www.who.int/fr/news-room/fact-sheets/detail/road-traffic-injuries>

[2] Drew Banques : « Démystifier YOLO »
<https://www.datalabelify.com/fr/algorithmes-yolo-pour-la-detection-dobjet>

[3] Ronit Taleti dans *Towards Data Science*: « YOLO, a deep learning Convolutional neural network »
<https://towardsdatascience.com/yolo-you-only-look-once-f05c054a06b4>

[4] Glenn Jaucher : Ultralytics FAQ
<https://docs.ultralytics.com/help/FAQ/?h=hardware+requirements#1-what-are-the-hardware-requirements-for-running-ultralytics-yolo>

[5] Publication sur *blent.ia* : « YOLO : détection des images avec TensorFlow »
<https://blent.ai/blog/a/detection-images-yolo-tensorflow>

[6] Petru Potrimba : « What is Mask R-CNN »
<https://blog.roboflow.com/mask-rcnn/>

[7] Publication sur *saagie.com* : « Qu'est-ce que la détection d'objets ? »
<https://www.saagie.com/fr/blog/quest-ce-que-la-detection-dobjet/>

[8] Claude Fendzi : « YOLOV8 vs Faster R-CNN »
https://www.linkedin.com/posts/claude-fendzi-662a1350_yolov8-vs-faster-r-cnn-en-vision-par-ordinateur-activity-7101975079030247424-q3rj/?originalSubdomain=fr

[9] Posté sur *technostacks.com* : « YOLO vs SSD : Which is Best Object Detection Method »
<https://technostacks.com/blog/yolo-vs-ssd/>

[10] Posté par le site *developers.arcgis.com* : « How single-shot detector (SSD) works ? »
<https://developers.arcgis.com/python/guide/how-ssd-works/>

[11] Lambert R. : « MobileNet, une reconnaissance d'images temps réel et embarquée surpuissante »
<https://penseeartificielle.fr/mobilenet-reconnaissance-images-temps-reel-embarquee/>

[12] *Lambert R.* : « Focus : MobileNet-SSD, pour identifier les objets avec une caméra de smartphone »

<https://penseeartificielle.fr/mobilenet-ssd-identifier-objets-camera-smartphone/>

[13] *Tauseef Ahmad* dans *MLearning.ai* :
« Object Detection using mobilenet SSD »

<https://medium.com/mlearning-ai/object-detection-using-mobilenet-ssd-e75b177567ee>