



# LA DÉTECTION D'OBJETS À L'AIDE DE L'ALGORITHME YOLO

DEPARTMENT INFORMATIQUE  
4ÈME ANNÉE EN DATA SCIENCE

Haboubi Oussema  
Dahmeni Meysoun  
Zitouni Hajar

## SOMMAIRE

1	Introduction generale	4
2	Fondements de la Détection d'Objets	4
2.1	Vue d'Ensemble de la Détection d'Objets . . . . .	4
2.2	Machine learning vs deep learning in object detection . . . . .	5
2.3	Processus de la detection d'objet . . . . .	6
2.4	Localisation d'Objets . . . . .	7
2.5	Les algorithmes de détection d'objet . . . . .	9
2.5.1	R-CNN . . . . .	9
3	YOLO	11
3.1	C'est quoi YOLO ? . . . . .	11
3.2	Pourquoi YOLO est-il populaire pour la Détection d'Objets ? . . . . .	11
3.2.1	Vitesse . . . . .	11
3.2.2	Précision de la détection élevée . . . . .	11
3.2.3	Meilleure généralisation . . . . .	12
3.2.4	Open Source . . . . .	12
3.3	Architecture de YOLO . . . . .	12
3.4	Comment Fonctionne la Détection d'Objets YOLO ? . . . . .	13
3.4.1	Blocs résiduels . . . . .	13
3.4.2	Régression des boîtes englobantes . . . . .	14
3.4.3	Intersection Over Unions . . . . .	15
3.4.4	Non-Max Suppression . . . . .	16
3.5	Applications de YOLO . . . . .	16
3.6	Versions de YOLO . . . . .	17
3.7	YOLO v7 . . . . .	18
3.7.1	Principales caractéristiques de YOLOv7 . . . . .	18
3.7.2	Améliorations de performance . . . . .	18
4	Mise en oeuvre du projet	19
4.1	Introduction au projet . . . . .	19
4.2	Tâches du projet . . . . .	19
4.2.1	Recherche et collection de données : . . . . .	19
4.2.2	Préparation du jeu de données : . . . . .	19
4.2.3	Annotation des images : . . . . .	20
4.2.4	Entraînement du modèle machine learning et evaluation des résultats : . . . . .	20
4.2.5	Test et optimisation des résultats : . . . . .	20
4.2.6	Entraînement avec le modèle YOLO v8 : . . . . .	20
5	Conclusion générale:	21

## LIST OF FIGURES

Figure 1	Classification, Detection d'Object et Segmentation . . . . .	4
Figure 2	Déférence de détection d'objets basée sur Machine learning et Deep learning . . . . .	5
Figure 3	Classification D'images . . . . .	6
Figure 4	Détection d'objets avec cadre de délimitation . . . . .	6
Figure 5	Localisation de voiture . . . . .	7
Figure 6	Calculer IoU loss . . . . .	8
Figure 7	Mean IoU . . . . .	8
Figure 8	R-CNN . . . . .	9
Figure 9	Comparaison de la vitesse des algorithmes de détection d'objets YOLO Speed compared to other state-of-the-art object detectors	10
Figure 10	YOLO Architecture from the original paper . . . . .	11
Figure 11	Players Detection . . . . .	12
Figure 12	Residual blocks . . . . .	13
Figure 13	Vector representation for each bounding box. . . . .	14
Figure 14	Probability score of the grid containing an object . . . . .	14
Figure 15	Closer attention to the player on the bottom right . . . . .	15
Figure 16	Applying the grid selection process to the bottom left object . . . . .	15
Figure 17	2D Kidney detection by YOLOv3 . . . . .	16
Figure 18	Image from Tomato detection based on modified YOLOv3 . . . . .	17
Figure 19	Evolution of Yolo . . . . .	17
Figure 20	Yolo network architecture . . . . .	18
Figure 21	exemples des images entrainées . . . . .	19
Figure 22	Exemples images testées avec le modèle yolo V8 . . . . .	21
Figure 23		

---

\* Department of Biology, University of Examples, London, United Kingdom

<sup>1</sup> Department of Chemistry, University of Examples, London, United Kingdom

## 1 INTRODUCTION GENERALE

Au cours de la dernière décennie, l'apprentissage profond a attiré beaucoup plus d'attention et est devenu une technologie impérieuse dans le domaine de l'intelligence artificielle. La détection d'objets est considérée comme l'une des zones remarquables de l'apprentissage profond et de la vision par ordinateur. La détection d'objets a été déterminée comme ayant de nombreuses applications en vision par ordinateur, telles que le suivi d'objets, la recherche, la surveillance vidéo, la légende d'images, la segmentation d'images, l'imagerie médicale, et plusieurs autres applications également. Dans ce projet, nous allons comprendre toutes les notions fondamentales de la détection d'objets et de l'algorithme YOLO.

Dans ce rapport, nous expliquons d'abord les avantages de la détection d'objets, avant d'introduire YOLO, l'algorithme de détection d'objets de pointe.

Dans la deuxième partie, nous nous concentrerons davantage sur l'algorithme YOLO et sur son fonctionnement. Ensuite, nous présenterons quelques applications réelles utilisant YOLO.

## 2 FONDEMENTS DE LA DÉTECTION D'OBJETS

### 2.1 Vue d'Ensemble de la Détection d'Objets

**LA DÉTECTION D'OBJETS** : est une technologie de Deep Learning, où des éléments tels que des objets, des humains, des bâtiments et des voitures peuvent être détectés comme des objets dans des images et des vidéos.

**LA DÉTECTION D'OBJETS** vise simplement à reconnaître l'objet avec une boîte englobante (Bounding Box) dans l'image, tandis que dans la classification d'image, nous pouvons simplement catégoriser (classer) s'il s'agit d'un objet dans l'image ou non en termes de probabilité.(reference)

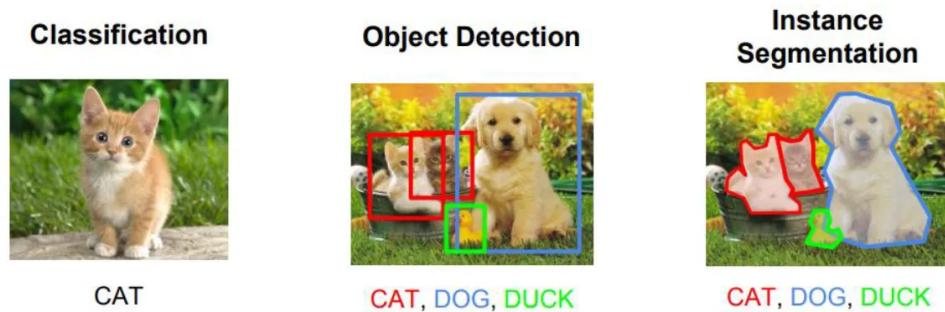


Figure 1: Classification, Detection d'Object et Segmentation

Dans la figure 1, vous pourrez voir que le chaton avec une boîte englobante et sans boîte englobante permet de distinguer la différence fondamentale entre la **classification d'image et la détection d'objets**.

## 2.2 Machine learning vs deep learning in object detection

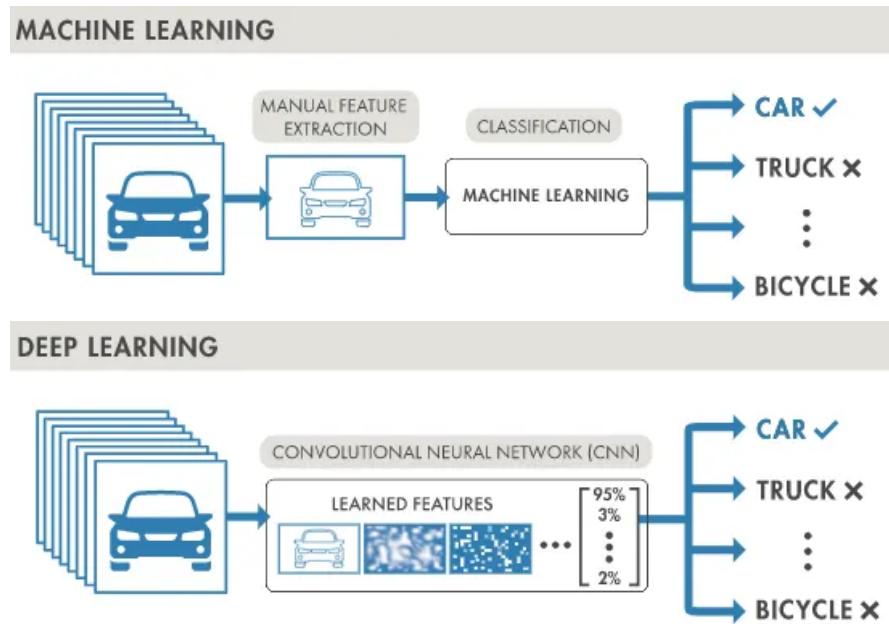


Figure 2: Différence de détection d'objets basée sur Machine learning et Deep learning

- **La méthode de détection d'objets basée sur l'apprentissage automatique doit extraire manuellement les caractéristiques en utilisant des techniques d'extraction de caractéristiques basées sur l'image telles que Histogram of oriented gradients(HOG), Speeded-up robust features(SURF), Local binary patterns (LBP), Haar wavelets, Color histograms etc.**
- **La méthode de détection d'objets basée sur l'apprentissage profond a la capacité d'extraire automatiquement les caractéristiques à l'aide d'algorithmes d'apprentissage profond tels que le réseau neuronal convolutionnel (Convolutional Neural Network - CNN), l'autoencodeur (Auto-Encoder), le Variational Autoencoder, etc. Ces algorithmes génèrent des caractéristiques à partir de l'image, telles que les contours, les formes, etc.**

### 2.3 Processus de la détection d'objet

Avant d'explorer la détection d'objets, nous devons comprendre la **classification d'images**. La classification d'images est accompagnée d'une abondance de matériel à étudier, et vous avez peut-être déjà mis en œuvre au moins une fois le processus via un tutoriel.

Lorsqu'une image est fournie en entrée au réseau neuronal convolutionnel (CNN), le problème de classer la classe correspondant à l'image est appelé la classification d'images. Comme illustré dans la figure ci-dessous, **des valeurs de probabilité pour toutes les classes cibles sont produites en sortie**.

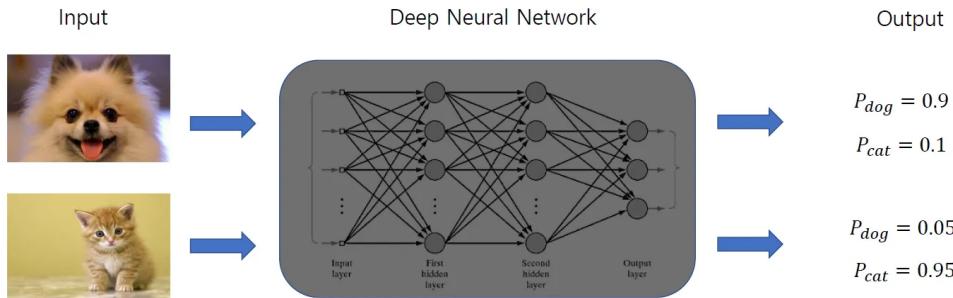


Figure 3: Classification D'images

On peut envisager la détection d'objets comme un problème dans lequel une tâche de classification d'images s'accompagne d'une tâche de **régression** qui prédit la position d'un objet à l'aide d'une boîte englobante. En d'autres termes, la détection d'objets combine la capacité de classer les objets présents dans une image avec celle de prédire leur emplacement précis en délimitant une boîte autour d'eux. Cette approche permet une compréhension plus détaillée et précise de la localisation des objets dans une scène visuelle.

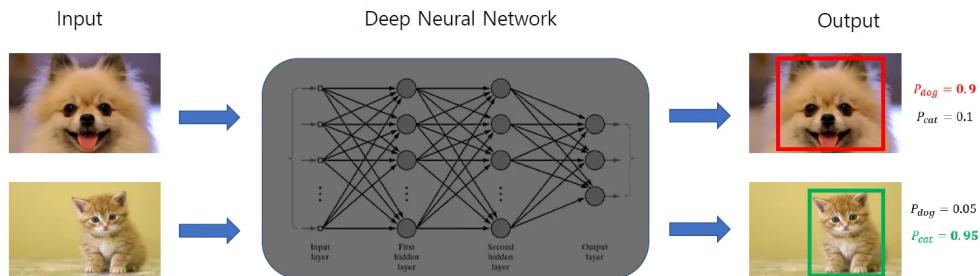


Figure 4: Détection d'objets avec cadre de délimitation

Le problème de la détection d'objets suppose que plusieurs classes d'objets peuvent coexister dans une seule image. On peut également visualiser cela comme deux types de problèmes :

- la classification multi-étiquettes (plusieurs classes dans une seule image)
- la boîte englobante (problème de régression), dans lequel nous devons prédire les valeurs de coordonnées de la boîte englobante en termes de  $x, y, w, h$ .

## 2.4 Localisation d'Objets

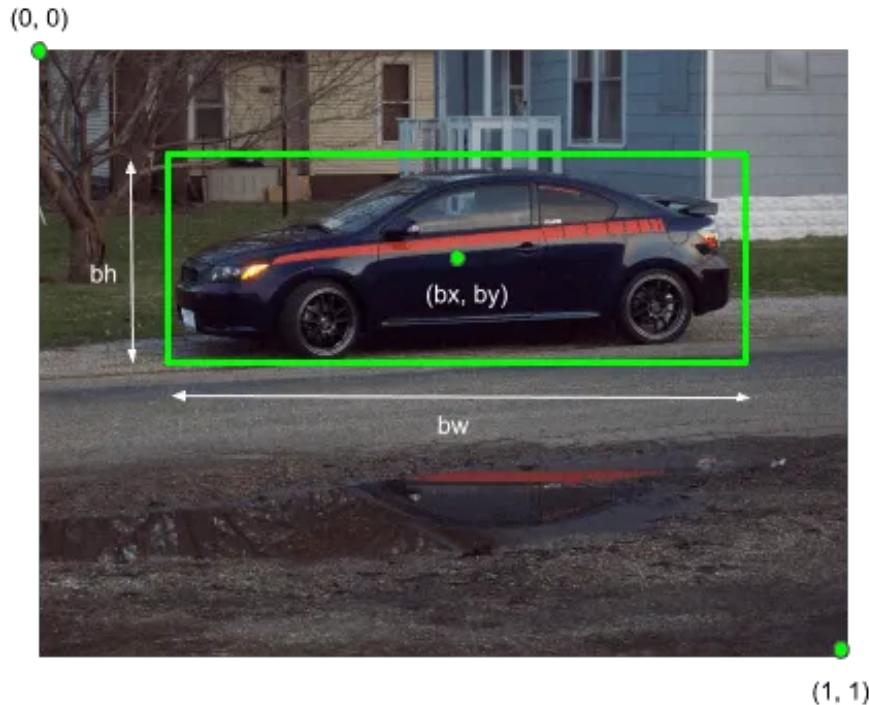


Figure 5: Localisation de voiture

La tâche de la **localisation d'objets** consiste à prédire l'objet dans une image ainsi que ses limites. La différence entre la localisation d'objets et la détection d'objets est subtile. En termes simples, la localisation d'objets vise à repérer l'objet principal (ou le plus visible) dans une image, tandis que la détection d'objets tente de trouver tous les objets et leurs limites.

Un modèle de classification d'image ou de reconnaissance d'image détecte simplement la probabilité d'un objet dans une image. En revanche, la localisation d'objets consiste à identifier l'emplacement d'un objet dans l'image. Un algorithme de localisation d'objets produira les coordonnées de l'emplacement d'un objet par rapport à l'image. En vision par ordinateur, la façon la plus courante de localiser un objet dans une image est de représenter son emplacement à l'aide de boîtes englobantes.

Une boîte englobante peut être initialisée en utilisant les paramètres suivants :

- $bx, by$  : coordonnées du centre de la boîte englobante
- $bw$  : largeur de la boîte englobante par rapport à la largeur de l'image
- $bh$  : hauteur de la boîte englobante par rapport à la hauteur de l'image

En prédisant cela, ils calculent la Mean Intersection-Over-Union (Mean IoU) et prédisent la boîte englobante, ce qui permet de localiser l'objet dans l'image.

L'Intersection-over-Union (IoU), également appelé indice de Jaccard, est considéré comme l'une des métriques de performance les plus largement utilisées en détection d'objets.

L'IoU mesure le degré de chevauchement entre la boîte englobante prédite et la boîte englobante réelle de l'objet. C'est le rapport de la zone d'intersection entre les deux boîtes sur la zone de leur union. Un IoU élevé indique une meilleure précision dans la localisation de l'objet. La Mean IoU est souvent utilisée comme mesure agrégée de la performance de détection d'objets sur un ensemble de données.

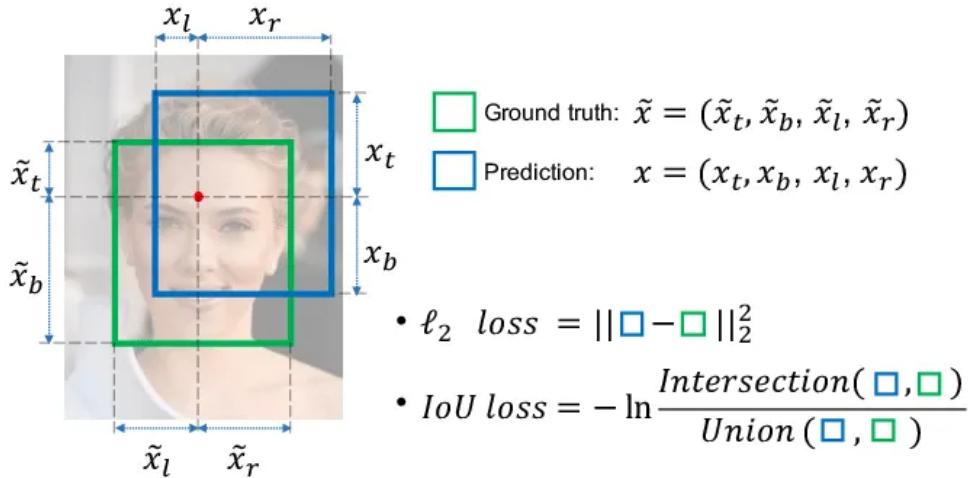


Figure 6: Calculer IoU loss

- **L'Intersection-over-Union (IoU):** est la zone de chevauchement entre la segmentation prédite et la vérité terrain, divisée par la zone d'union entre la segmentation prédite et la vérité terrain, comme indiqué sur l'image à gauche. Cette métrique varie de 0 à 1 (0 à 100%), 0 impliquant aucun chevauchement (informations non pertinentes) et 1 signifiant une segmentation parfaitement superposée (excellente correspondance).

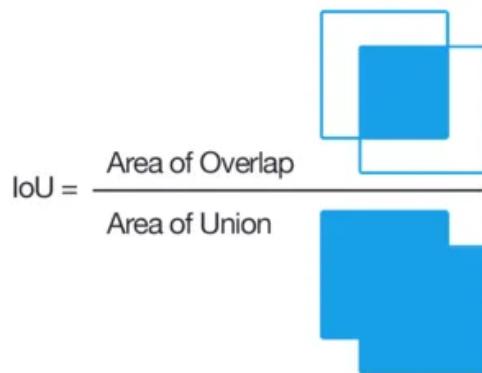


Figure 7: Mean IoU

- **Mean IoU :** Pour une segmentation binaire (deux classes) ou multiclasse, la Mean IoU d'une image est calculée en prenant l'IoU de chaque classe et en faisant la moyenne. Cela offre une mesure agrégée de la précision de la segmentation.

Maintenant, vous comprenez l'ensemble du jeu de la détection d'objets. Dans le prochain article, je vais expliquer une série d'algorithmes de détection d'objets qui sont à la mode sur le marché.

## 2.5 Les algorithmes de détection d'objet

### 2.5.1 R-CNN

R-CNN (Réseaux de Neurones Convolutifs basés sur les Régions) est une famille d'algorithmes de détection d'objets qui combinent des propositions de régions avec des réseaux de neurones convolutionnels (CNN). Les R-CNN sont des détecteurs en deux étapes, ce qui signifie qu'ils génèrent d'abord des propositions de régions, qui sont des boîtes englobantes potentielles contenant des objets, puis ils classifient et affinent ces propositions.

#### Etapes de R-CNN:

L'architecture R-CNN se compose de quatre étapes principales:

1. **Prétraitement de l'image :** L'image d'entrée est prétraitée pour normaliser son intensité, la redimensionner à une taille appropriée et l'augmenter avec des techniques telles que le retournement, le rognage et la variation des couleurs pour augmenter les données d'entraînement.
2. **Extraction de caractéristiques :** Un CNN est utilisé pour extraire des caractéristiques de l'image prétraitée. Les couches convolutionnelles du CNN apprennent à détecter des motifs et des formes dans l'image, tandis que les couches de mise en commun réduisent les dimensions spatiales tout en préservant les informations essentielles.
3. **Génération de propositions de régions :** Les propositions de régions sont générées à l'aide de techniques telles que la recherche sélective ou le réseau de propositions de régions (RPN). Ces propositions sont des boîtes englobantes potentielles qui pourraient contenir des objets.
4. **Classification et Affinement :** Les propositions de régions sont classifiées et affinées pour améliorer leur précision. La classification implique de déterminer la classe d'objet (par exemple, personne, voiture, chat) pour chaque proposition. L'affinement consiste à ajuster les boîtes englobantes pour mieux s'adapter aux limites de l'objet.

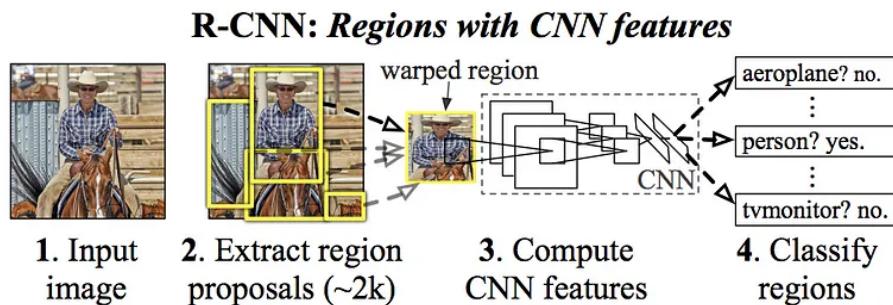


Figure 8: R-CNN

## Avantages de R-CNN:

R-CNN présente plusieurs avantages par rapport à d'autres algorithmes de détection d'objets :

- **Haute précision :** R-CNN atteint une précision de pointe sur divers bancs d'essai de détection d'objets.
- **Flexibilité :** R-CNN peut être adapté à différentes tâches de détection d'objets en utilisant différents algorithmes de proposition de régions et architectures de CNN.
- **Scalabilité :** R-CNN peut être entraîné sur de grands ensembles de données en utilisant des GPU.

## Applications de R-CNN

R-CNN a une large gamme d'applications, notamment :

- **Véhicules autonomes :** R-CNN peut détecter les piétons, les véhicules et les panneaux de signalisation pour les systèmes de conduite autonome.
- **Robotique :** R-CNN peut détecter des objets dans des environnements robotiques pour des tâches de manipulation et de navigation.
- **Imagerie médicale :** R-CNN peut détecter des tumeurs, des lésions et d'autres anomalies dans des images médicales.
- **Systèmes de surveillance :** R-CNN peut détecter des personnes, des véhicules et des activités suspectes dans des vidéos de surveillance.

## Faster R-CNN et Mask R-CNN

Faster R-CNN est une variante de R-CNN qui est significativement plus rapide que l'original R-CNN. Mask R-CNN est une variante de R-CNN qui peut également segmenter des objets en plus de les détecter.

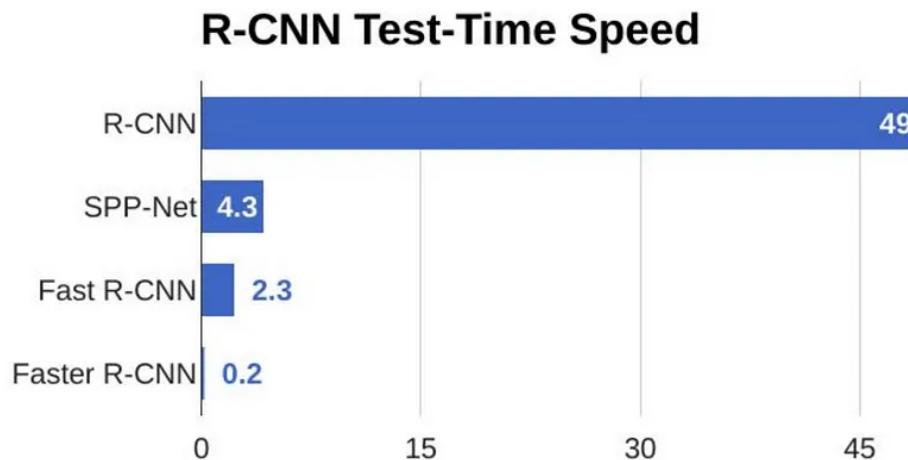


Figure 9: Comparaison de la vitesse des algorithmes de détection d'objets

### 3 YOLO

#### 3.1 C'est quoi YOLO ?

You Only Look Once (YOLO) is a state-of-the-art real-time object detection algorithm introduced in 2015 by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi in their renowned research paper titled "You Only Look Once: Unified, Real-Time Object Detection."

The authors approach the object detection problem as a regression problem rather than a classification task by spatially separating bounding boxes and associating probabilities with each of the detected objects using a single convolutional neural network (CNN).

YOLO est un algorithme à une seule étape, ce qui signifie qu'il prédit les boîtes englobantes et les classes des objets dans une image en une seule fois. Il divise l'image en une grille de cellules et prédit, pour chaque cellule, les boîtes englobantes et les classes des objets qui pourraient être présents dans la cellule.

YOLO est un algorithme efficace et rapide, ce qui le rend particulièrement adapté aux applications en temps réel.

#### 3.2 Pourquoi YOLO est-il populaire pour la Détection d'Objets ?

Certains des motifs pour lesquels YOLO domine la concurrence incluent :

##### 3.2.1 Vitesse

YOLO est extrêmement rapide car il n'utilise pas de pipelines complexes. Il peut traiter des images à 45 images par seconde (FPS). De plus, YOLO atteint plus du double de la précision moyenne (mAP) par rapport à d'autres systèmes en temps réel, ce qui en fait un excellent candidat pour le traitement en temps réel.

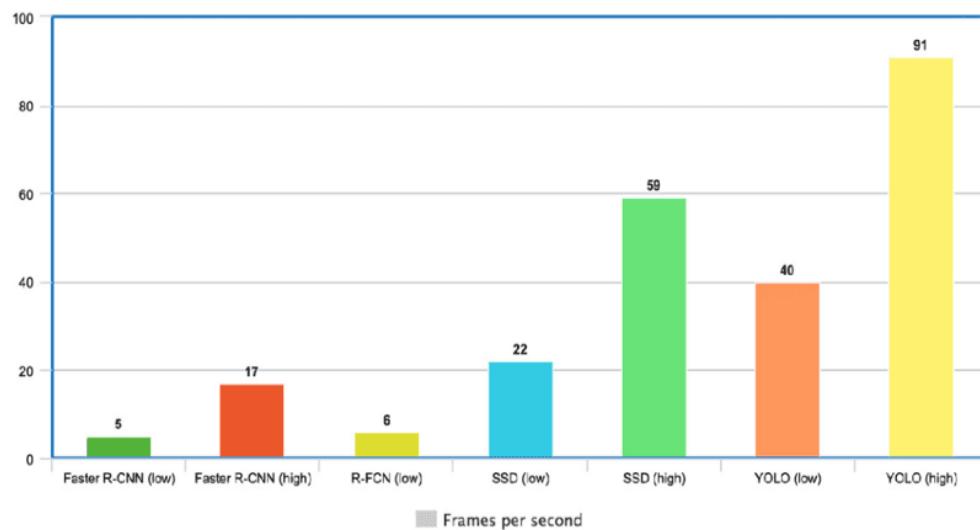


Figure 10: YOLO Speed compared to other state-of-the-art object detectors

##### 3.2.2 Précision de la détection élevée

YOLO dépasse de loin d'autres modèles de pointe en termes de précision, avec très peu d'erreurs de fond.

### 3.2.3 Meilleure généralisation

Cela est particulièrement vrai pour les nouvelles versions de YOLO, qui seront discutées plus tard dans l'article. Avec ces avancées, YOLO pousse un peu plus loin en offrant une meilleure généralisation pour de nouveaux domaines, ce qui le rend idéal pour des applications reposant sur une détection d'objets rapide et robuste.

### 3.2.4 Open Source

La mise en open source de YOLO a conduit la communauté à améliorer constamment le modèle. C'est l'une des raisons pour lesquelles YOLO a fait autant de progrès en si peu de temps.

## 3.3 Architecture de YOLO

L'architecture de YOLO est similaire à GoogleNet. Comme illustré ci-dessous, elle comporte au total 24 couches de convolution, quatre couches de max-pooling et deux couches entièrement connectées.

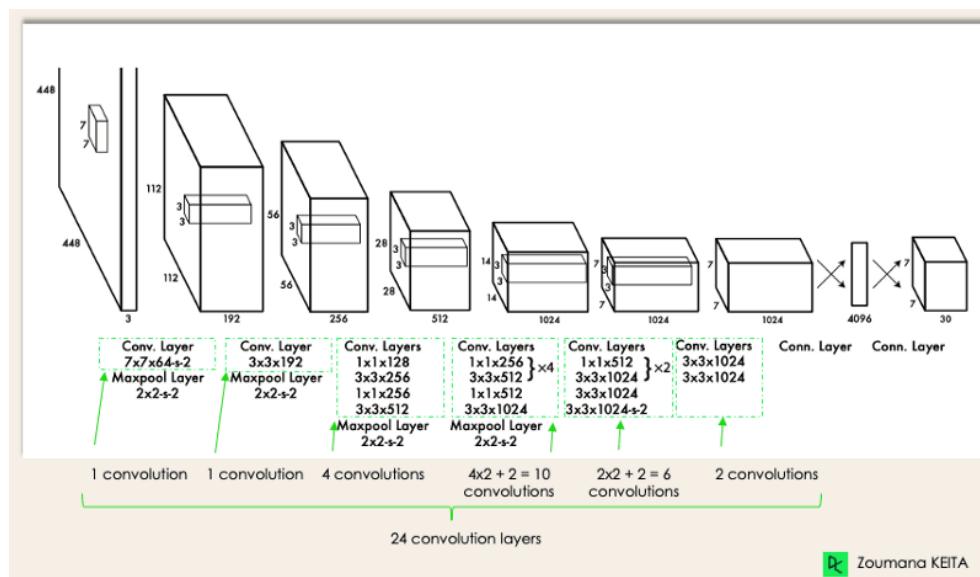


Figure 11: YOLO Architecture from the original paper

L'architecture fonctionne comme suit :

1. Redimensionne l'image d'entrée en 448x448 avant de passer par le réseau de convolution.
2. Une convolution 1x1 est d'abord appliquée pour réduire le nombre de canaux, suivie d'une convolution 3x3 pour générer une sortie cuboïdale potentielles qui pourraient contenir des objets.
3. La fonction d'activation utilisée est ReLU, sauf pour la dernière couche, qui utilise une fonction d'activation linéaire.
4. Certaines techniques supplémentaires, telles que la normalisation par lots et le dropout, régularisent respectivement le modèle et l'empêchent de surapprendre.

### 3.4 Comment Fonctionne la Détection d'Objets YOLO ?

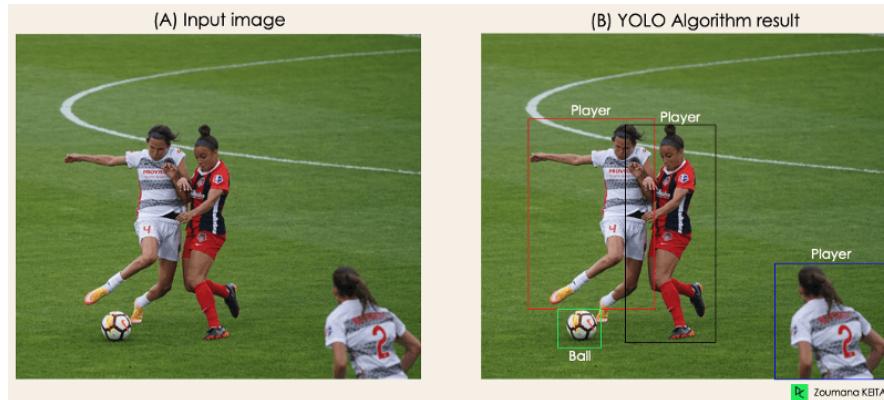


Figure 12: Players Detection

Maintenant que vous comprenez l'architecture, jetons un coup d'œil global sur la manière dont l'algorithme YOLO effectue la détection d'objets à l'aide d'un cas d'utilisation simple. "Imaginez que vous ayez construit une application YOLO qui détecte les joueurs et les ballons de soccer à partir d'une image donnée. Mais comment pouvez-vous expliquer ce processus à quelqu'un, surtout à des personnes non initiées ? → C'est tout l'objet de cette section. Vous comprendrez tout le processus de la manière dont YOLO effectue la détection d'objets ; comment obtenir l'image (B) à partir de l'image (A)" L'algorithme fonctionne selon les quatre approches suivantes :

1. Blocs résiduels
2. Régression des boîtes englobantes
3. Intersection Over Unions (IOU)
4. Non-Maximum Suppression (NMS)

Examinons de plus près chacun d'entre eux.

#### 3.4.1 Blocs résiduels

Cette première étape commence par diviser l'image originale (A) en cellules de grille NxN de forme égale, où N dans notre cas est 4, comme indiqué sur l'image à droite. Chaque cellule de la grille est responsable de la localisation et de la prédiction de la classe de l'objet qu'elle couvre, ainsi que de la valeur de probabilité/confiance.

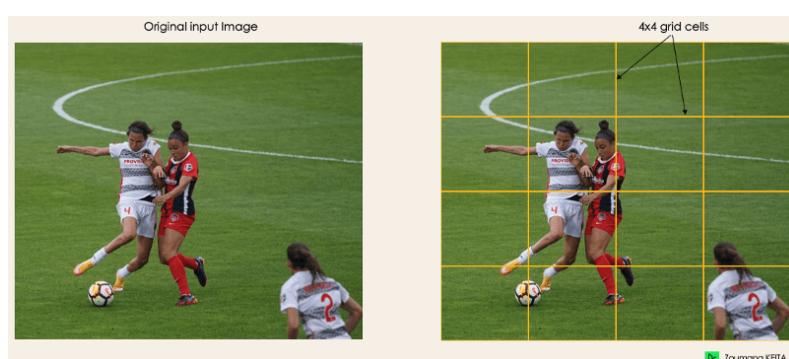


Figure 13: Residual blocks

### 3.4.2 Régression des boîtes englobantes

- La prochaine étape consiste à déterminer les boîtes englobantes qui correspondent aux rectangles mettant en évidence tous les objets dans l'image. Nous pouvons avoir autant de boîtes englobantes qu'il y a d'objets dans une image donnée.
- YOLO détermine les attributs de ces boîtes englobantes à l'aide d'un seul module de régression dans le format suivant, où Y est la représentation vectorielle finale pour chaque boîte englobante.

$$Y = [pc, bx, by, bh, bw, c_1, c_2]$$

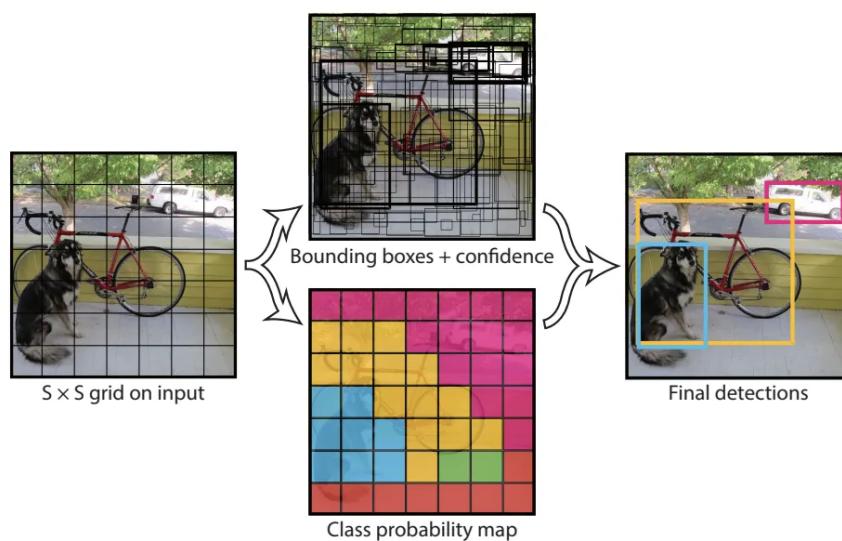


Figure 14: Vector representation for each bounding box.

Ceci est particulièrement important pendant la phase d'entraînement du modèle.

- **pc** correspond au score de probabilité de la grille contenant un objet. Par exemple, toutes les grilles en rouge auront un score de probabilité supérieur à zéro. L'image à droite est la version simplifiée, car la probabilité de chaque cellule jaune est nulle (insignifiante).

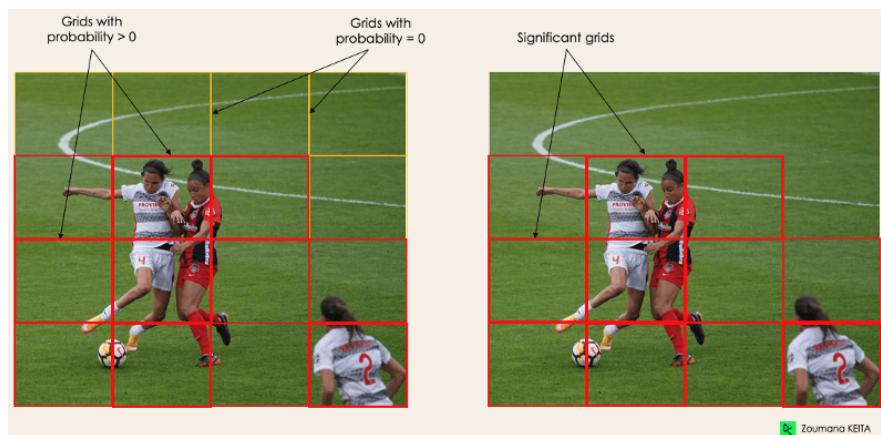


Figure 15: Probability score of the grid containing an object

- **bx, by** sont les coordonnées x et y du centre de la boîte englobante par rapport à la cellule de grille enveloppante.

- **bh**, **bw** correspondent à la hauteur et à la largeur de la boîte englobante par rapport à la cellule de grille enveloppante.
- **c1** et **c2** correspondent aux deux classes Player et Ball. Nous pouvons avoir autant de classes que votre cas d'utilisation l'exige. Pour comprendre, prêtons une attention particulière au joueur en bas à droite.

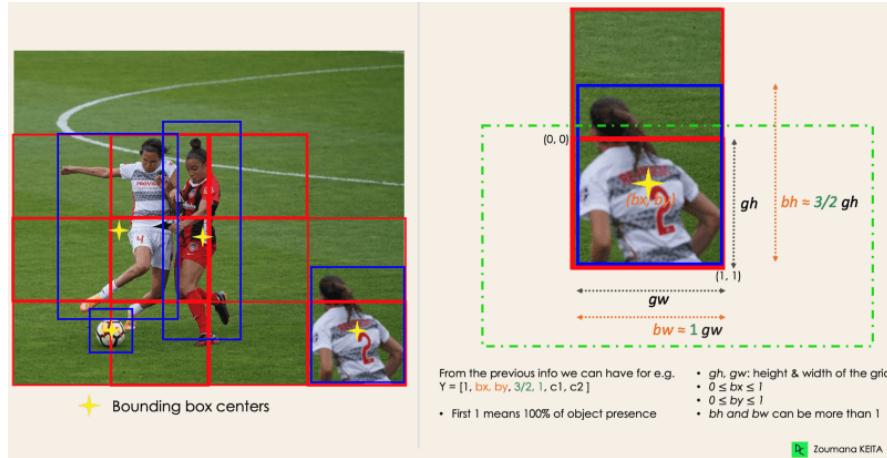


Figure 16: Closer attention to the player on the bottom right

### 3.4.3 Intersection Over Unions

La plupart du temps, un seul objet dans une image peut avoir plusieurs candidats de boîtes de grille pour la prédiction, même si tous ne sont pas pertinents. L'objectif de l'IOU (une valeur entre 0 et 1) est de rejeter de telles boîtes de grille pour ne conserver que celles qui sont pertinentes. Voici la logique qui sous-tend cela :

1. L'utilisateur définit son seuil de sélection IOU, qui peut être, par exemple, 0,5.
2. Ensuite, YOLO calcule l'IOU de chaque cellule de grille, qui est la zone d'intersection divisée par la zone d'union.
3. Enfin, il ignore la prédiction des cellules de grille ayant un IOU seuil et considère celles avec un IOU > seuil. Voici une illustration de l'application du processus de sélection de grille à l'objet en bas à gauche. Nous pouvons observer que l'objet avait initialement deux candidats de grille, puis seulement "Grid 2" a été sélectionné à la fin.

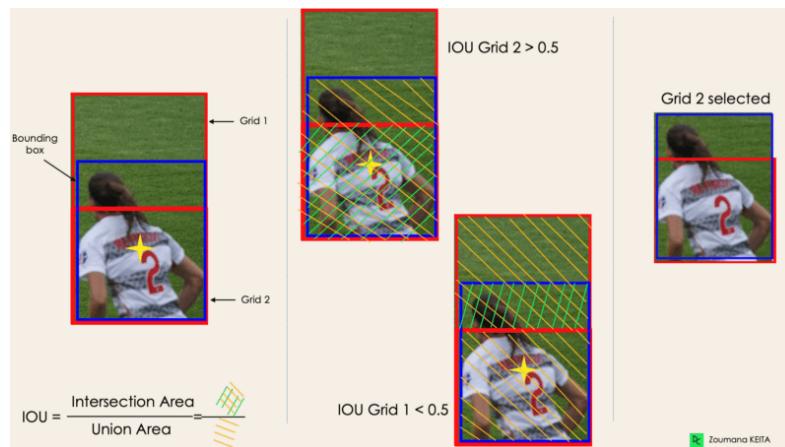


Figure 17: Applying the grid selection process to the bottom left object

### 3.4.4 Non-Max Suppression

Fixer un seuil pour l'IOU n'est pas toujours suffisant, car un objet peut avoir plusieurs boîtes avec un IOU au-delà du seuil, et laisser toutes ces boîtes pourrait inclure du bruit. C'est ici que nous pouvons utiliser le NMS pour ne conserver que les boîtes ayant le score de probabilité de détection le plus élevé.

## 3.5 Applications de YOLO

La détection d'objets YOLO a différentes applications dans notre vie quotidienne. Nous en couvrirons certaines dans les domaines suivants : santé, agriculture, surveillance de la sécurité et voitures autonomes.

- Santé:** Dans le domaine de la santé, plus précisément en chirurgie, la détection d'objets YOLO peut être d'une grande utilité. La localisation en temps réel d'organes peut être un défi en raison de la diversité biologique d'un patient à l'autre. YOLO peut être appliqué pour aider à la localisation précise des organes pendant une intervention chirurgicale, ce qui peut améliorer la précision et l'efficacité des procédures médicales. Cela peut être particulièrement important dans les chirurgies complexes où une localisation précise des organes est cruciale pour le succès de l'intervention.

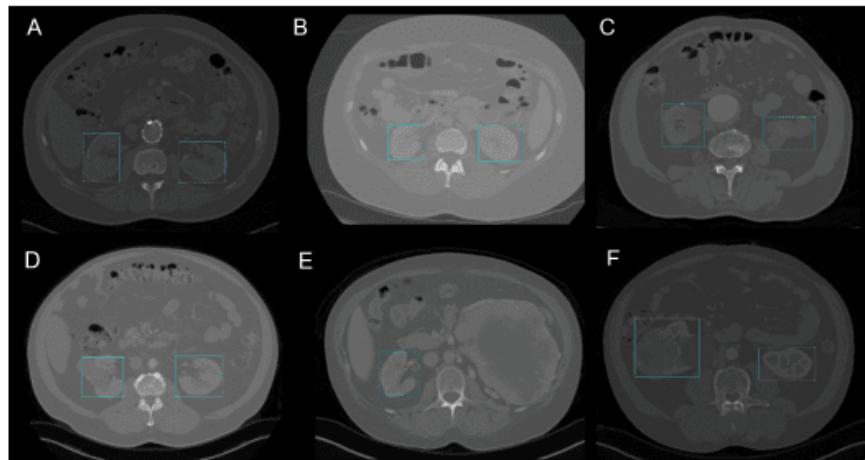


Figure 18: 2D Kidney detection by YOLOv3

- agriculture:** Dans le domaine de l'agriculture, l'intelligence artificielle et la robotique jouent un rôle majeur. Les robots de récolte, basés sur la vision, ont été introduits pour remplacer la cueillette manuelle de fruits et de légumes. L'un des meilleurs modèles dans ce domaine utilise YOLO. Dans le cadre de la détection de tomates basée sur le framework YOLOv3 modifié, les auteurs décrivent comment ils ont utilisé YOLO pour identifier les types de fruits et de légumes en vue d'une récolte efficace. Cela démontre comment la détection d'objets YOLO peut contribuer à l'automatisation des processus agricoles, améliorant ainsi l'efficacité et la productivité de la récolte.

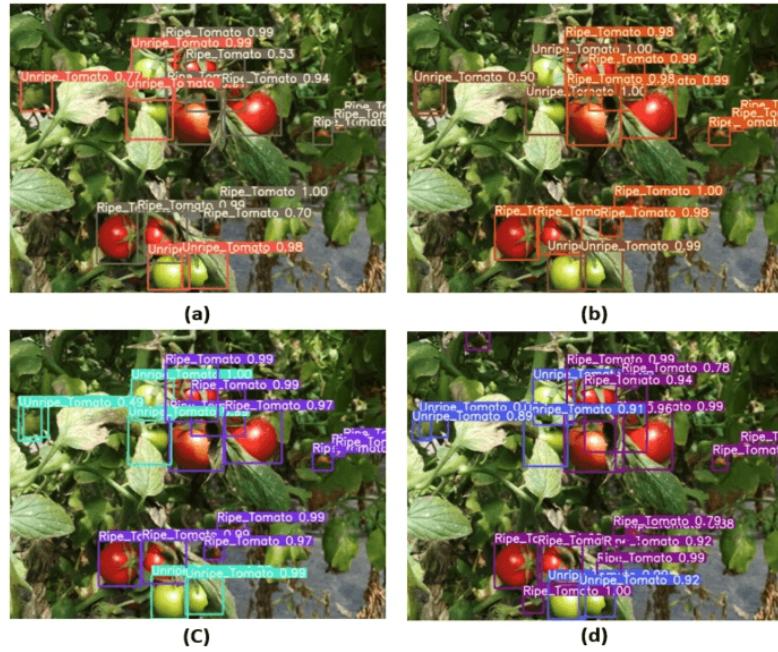


Figure 19: Image from Tomato detection based on modified YOLOv3

### 3.6 Versions de YOLO

**YOLO** (You Only Look Once) a évolué à travers différentes versions au fil du temps, chacune introduisant des améliorations significatives en termes de vitesse, de précision, et de facilité d'utilisation. Depuis la version initiale (**YOLOv1**) en 2015 jusqu'à la version actuelle en cours de développement (**YOLOv7**), des caractéristiques telles que la normalisation par lots, la prédiction multi-échelle, les réseaux de pyramides de caractéristiques, et des architectures plus efficaces ont été intégrées pour renforcer les performances de détection d'objets. Chaque itération vise à repousser les limites de la détection d'objets en temps réel, témoignant de l'engagement continu de la communauté à perfectionner YOLO en tant qu'algorithme d'avant-garde.

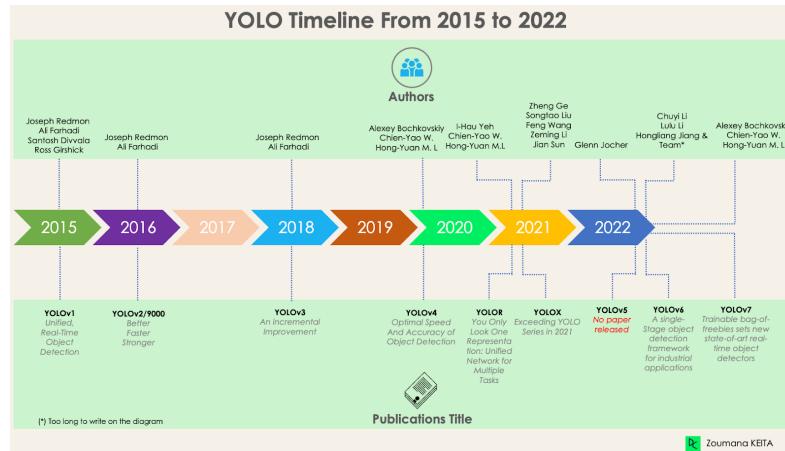


Figure 20: Evolution of Yolo

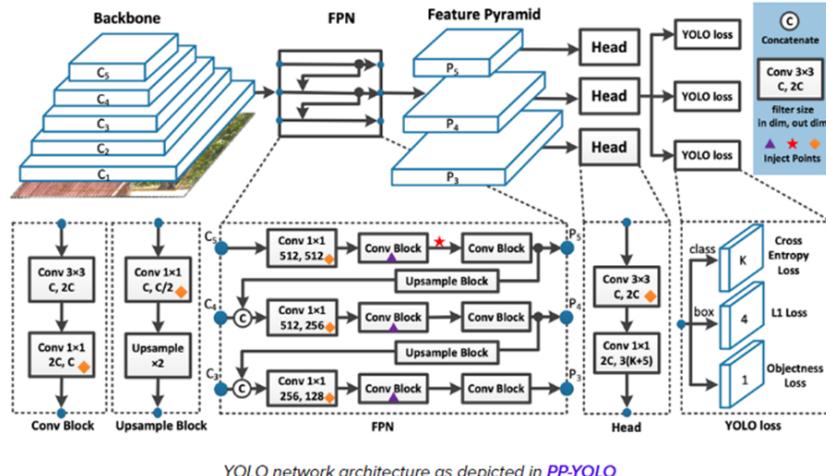
YOLO est un algorithme en évolution rapide, et de nouvelles versions sont constamment publiées. Chaque nouvelle version vise à améliorer les versions précédentes en termes de vitesse, de précision et de facilité d'utilisation.

### 3.7 YOLO v7

YOLOv7 est la version la plus récente et la plus avancée de l'algorithme de détection d'objets YOLO. Il a été publié en octobre 2022 par Ultralytics, les créateurs de YOLOv5. YOLOv7 capitalise sur les forces de YOLOv5, introduisant plusieurs nouvelles fonctionnalités et améliorations qui en font le modèle YOLO le plus puissant et précis à ce jour.

#### 3.7.1 Principales caractéristiques de YOLOv7

- Connexions partielles en étape croisée (CSP) :** CSP est une architecture novatrice qui améliore le flux d'informations et réduit la complexité computationnelle dans le réseau.
- Réseau d'agrégation de chemins (PANet) :** PANet est un module de fusion de fonctionnalités qui permet à YOLOv7 de combiner efficacement des caractéristiques de plusieurs étapes du réseau, améliorant les performances de détection d'objets à différentes échelles.
- Bloc Squeeze-and-Excitation (SE) :** Des blocs SE sont insérés dans le réseau pour améliorer la focalisation sur des caractéristiques informatives et réduire l'impact de caractéristiques moins pertinentes.
- Activation Mish :** Mish est une fonction d'activation qui présente de meilleures performances par rapport à la fonction d'activation ReLU largement utilisée, en particulier pour la détection d'objets de petite taille.
- Pyramide d'images :** YOLOv7 utilise plusieurs résolutions d'image pour traiter une gamme plus large de tailles d'objets, améliorant sa capacité à détecter des objets de dimensions variées.



YOLO network architecture as depicted in PP-YOLO

Figure 21: Yolo network architecture

#### 3.7.2 Améliorations de performance

YOLOv7 réalise des améliorations significatives tant en termes de vitesse que de précision par rapport à son précurseur, YOLOv5. Le modèle est capable de détecter des objets avec une précision moyenne (mAP) de 52,4% sur l'ensemble de données COCO, dépassant le record précédent de 51,1 % établi par YOLOv5. De plus, YOLOv7 atteint cette performance tout en maintenant des capacités de détection d'objets en temps réel.

## 4 MISE EN OEUVRE DU PROJET

### 4.1 Introduction au projet

Notre objectif principal est de mettre au point un modèle permettant au robot de sécurité externe de surveiller l'état des portes et des fenêtres, une tâche cruciale pour garantir la sécurité des environnements résidentiels et commerciaux. Ce projet allie la technologie de pointe à la sécurité, offrant ainsi une occasion exceptionnelle d'acquérir des compétences pratiques dans le domaine de l'apprentissage automatique.



Figure 22: exemples des images entraînées

### 4.2 Tâches du projet

#### 4.2.1 Recherche et collection de données :

- La tâche consiste à rassembler 501 images de portes et de fenêtres de garages prises de jour et de nuit pour développer un modèle de détection d'objets spécifique.
- Utilisation de moteurs de recherche d'images, de programmes Python (comme bing-image-downloader), et de logiciels de montage vidéo (comme shotcut) pour collecter des images diversifiées représentant différentes orientations, tailles, couleurs, textures et angles de prises de vue.

#### 4.2.2 Préparation du jeu de données :

- **Filtrage des données :** Prétraitement des images, incluant le redimensionnement, le recadrage, l'ajustement des niveaux de luminosité, et la normalisation.
- **Traitements des images :**
  - Utilisation de Photoshop pour appliquer différentes transformations, augmentant ainsi la diversité de l'ensemble de données.
  - Application d'algorithmes de traitement d'image en Python, notamment avec la bibliothèque PIL.
  - Construction du jeu de données final en regroupant toutes les images traitées dans un format standard.

#### **4.2.3 Annotation des images :**

- Utilisation de la plateforme Roboflow pour annoter précisément les 501 images, marquant chaque porte et fenêtre pour l'entraînement d'un modèle de détection.
- Division des images en ensembles d'entraînement, de validation et de test.

#### **4.2.4 Entraînement du modèle machine learning et évaluation des résultats :**

- Entraînement :
  - Utilisation du modèle YOLO v7 avec 50 epochs :
  - Installation des dépendances, importation du jeu de données et entraînement du modèle .
- Évaluation :
  - Analyse des courbes de performance y compris F<sub>1</sub>, courbe précision-rappel, et matrice de confusion.
  - Évaluation sur l'ensemble des images de test, avec affichage des résultats de détection.

#### **4.2.5 Test et optimisation des résultats :**

- Rectification et réordonnancement du jeu de données :
  - Suppression des données non utiles, comme les images de garages encombrés, et redistribution des ensembles d'entraînement, de validation et de test.
  - Création d'une nouvelle version du jeu de données avec 496 images.
- Répétition des entraînements avec ajustements des paramètres :
  - Plusieurs entraînements avec différentes epochs (75, 85, 95) pour améliorer la précision du modèle.
  - Création d'une nouvelle version du jeu de données avec 496 images.

#### **4.2.6 Entraînement avec le modèle YOLO v8 :**

Test du modèle YOLO v8, mais les résultats ne surpassent pas ceux du dernier entraînement avec YOLO v7 en termes de précision.

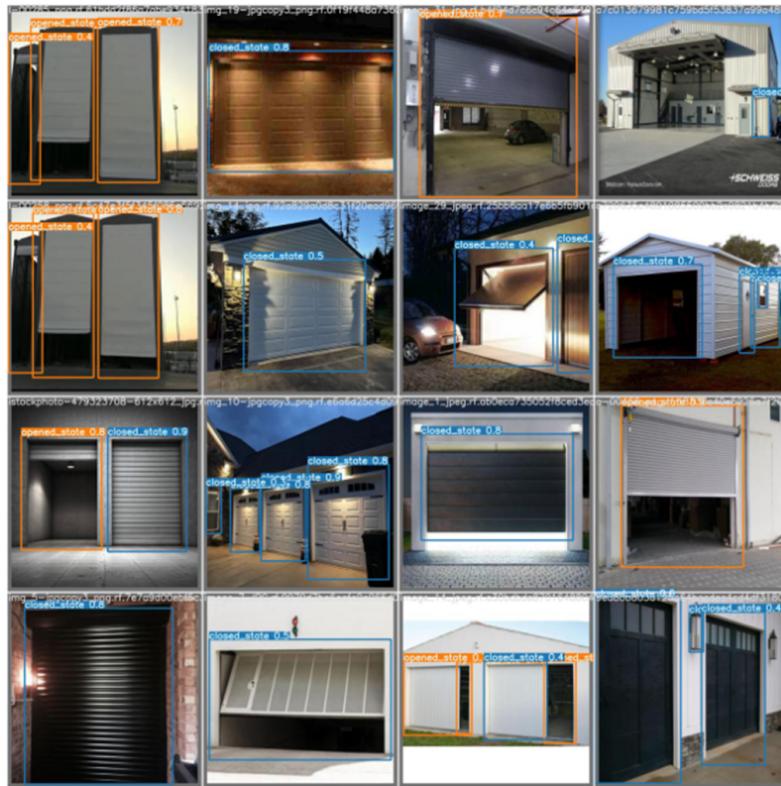


Figure 23: Exemples images testées avec le modèle yolo V8

## 5 CONCLUSION GÉNÉRALE:

En conclusion, ce rapport a exploré en profondeur les fondements de la détection d'objets, en mettant particulièrement l'accent sur l'algorithme YOLO (You Only Look Once). Nous avons examiné les concepts clés tels que la différence entre l'apprentissage automatique et l'apprentissage profond dans la détection d'objets, le processus de détection d'objets, la localisation des objets, ainsi que les algorithmes de détection d'objets tels que R-CNN, Faster R-CNN et YOLO.

La mise en œuvre pratique du projet a été abordée, avec des étapes détaillées, de la recherche et la collecte de données à l'évaluation et l'ajustement du modèle YOLO v7. Ce projet s'est concentré sur la surveillance des portes et des fenêtres pour renforcer la sécurité, démontrant l'application concrète de la détection d'objets dans des scénarios du monde réel.

L'évolution constante de l'algorithme YOLO, illustrée par les différentes versions, témoigne de son adaptabilité et de son engagement envers l'amélioration continue. En fin de compte, la détection d'objets, en particulier avec des approches telles que YOLO, continue de jouer un rôle crucial dans divers domaines, allant de la chirurgie à l'agriculture, démontrant son impact significatif dans notre quotidien.

## REFERENCES