الكلية متعددة التخصصات-ورزازات
ⵜⴰⵢⵙⵉⵍⵜ ⵜⴰⵣⵜⵉⵎⵣⵉⵜ – ⵡⵓⵔⵣⴰⵣⴰⵜ
FACULTÉ POLYDISCIPLINAIRE DE OUARZAZATE

Master Ingénierie Mathématiques et Science de Données

Numéro d'ordre : Gr : 8/IMSD-2025

YOLO-World : Détection d'objets en temps réel à vocabulaire ouvert

Mini-projet, présenté par :

Ismail Benradouan
Ismail Benali
Youssef Safir

Sous la direction de

Dr. **A**issam **Hadri**
Dr. **A**youb **Elhayyani**
Dr. **M**ohamed **Benaddy**

Soutenu le : 13 septembre 2025

Année Universitaire : 2024-2025

Table des matières

1 Introduction

La détection d'objets constitue un défi fondamental et de longue date en vision par ordinateur, avec de nombreuses applications dans la compréhension d'images, la robotique et les véhicules autonomes. Des travaux remarquables ont permis des avancées significatives dans ce domaine grâce au développement des réseaux neuronaux profonds. Malgré leur succès, ces méthodes demeurent limitées, car elles ne gèrent la détection d'objets qu'avec un vocabulaire fixe — par exemple, 80 catégories dans l'ensemble de données COCO. Une fois les catégories d'objets définies et annotées, les détecteurs entraînés ne peuvent reconnaître que ces catégories spécifiques, ce qui limite leur capacité et leur applicabilité dans des scénarios ouverts.

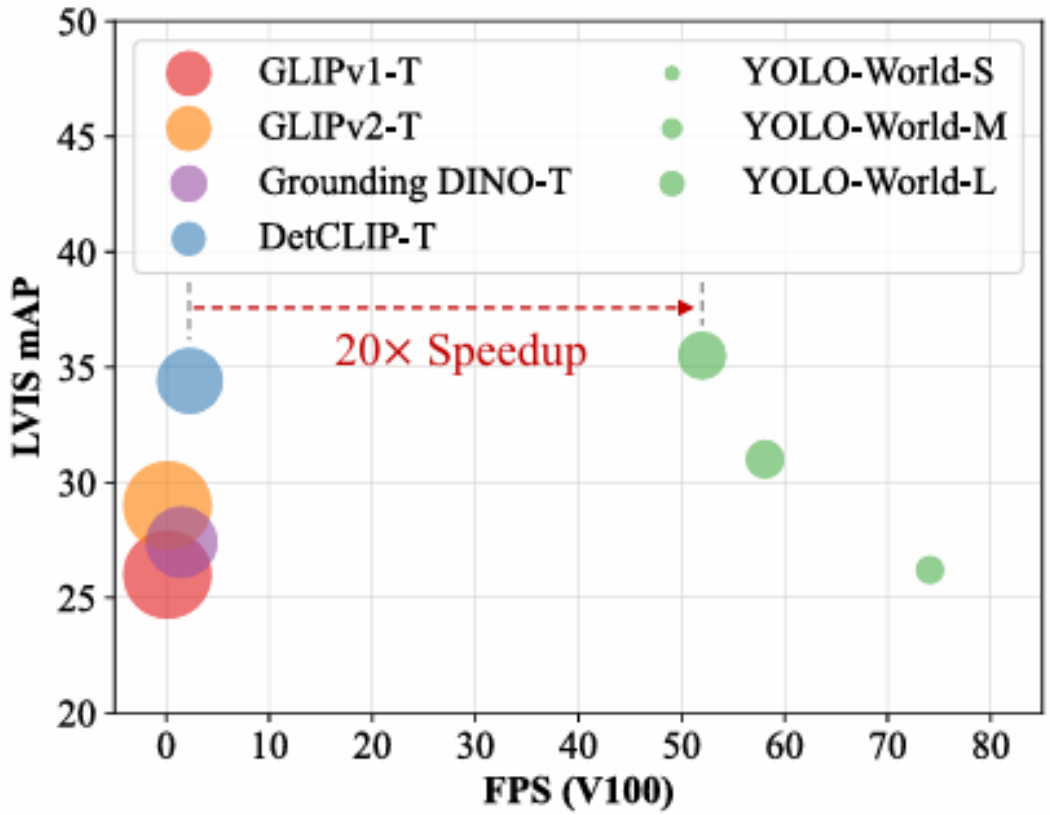

FIGURE 1 – Courbe de vitesse et de précision. Nous comparons YOLO-World aux méthodes à vocabulaire ouvert récentes en termes de vitesse et de précision. Tous les modèles sont évalués sur le minival LVIS et les vitesses d'inférence sont mesurées sur une carte NVIDIA V100 sans TensorRT. La taille du cercle représente la taille du modèle.

2 Travaux antérieurs

2.1 Détection d'objets traditionnelle

La recherche classique sur la détection d'objets se concentre sur la détection à vocabulaire fixe (close-set), dans laquelle les détecteurs sont entraînés sur des ensembles de données avec des catégories prédéfinies, par exemple COCO et Objects365, et ne peuvent détecter que les objets dans cet ensemble limité.

Avant l'avènement des réseaux neuronaux profonds, les méthodes traditionnelles reposaient sur des caractéristiques manuelles et des classifieurs classiques. Elles utilisaient des descripteurs comme SIFT, HOG ou Haar cascades pour représenter les objets, suivis de classifieurs tels que SVM ou AdaBoost. Ces méthodes ont permis des progrès significatifs dans la reconnaissance et la localisation d'objets simples dans des images contrôlées, mais restent limitées par leur incapacité à généraliser sur des images complexes et diversifiées, et par leur dépendance à des features conçues manuellement.

Les méthodes traditionnelles peuvent être regroupées en trois familles principales :

— **Méthodes basées sur les régions** (region-based), comme Faster R-CNN, qui adoptent un cadre en deux étapes pour générer des propositions et classifier les régions d'intérêt.

— **Méthodes basées sur les pixels** (pixel-based), qui sont en général des détecteurs en une étape (one-stage) et effectuent classification et régression directement sur des ancres ou pixels prédéfinis.

— **Méthodes basées sur les requêtes** (query-based), comme DETR, qui utilisent des transformeurs pour formuler la détection comme un problème de prédiction de requêtes.

En termes de rapidité d'inférence, Redmon et al. ont introduit la famille **YOLO**, qui repose sur des architectures convolutionnelles simples pour la détection en temps réel. Depuis, plusieurs variantes de YOLO ont été proposées afin d'améliorer simultanément la vitesse et la précision, par exemple à travers des réseaux d'agrégation de chemins, des architectures partielles en stades croisés (CSP), ou encore des techniques de re-paramétrisation.

En comparaison à ces approches, **YOLO-World** vise à dépasser la contrainte du vocabulaire fixe pour détecter un large éventail d'objets avec une forte capacité de généralisation.

2.2 Détection d'objets à vocabulaire ouvert

La détection à vocabulaire ouvert vise à détecter des objets au-delà des catégories prédéfinies. Des travaux récents ont exploré les modèles vision-langage les plus répandus pour aborder ce problème, en distillant les connaissances du vocabulaire à partir des encodeurs linguistiques, comme BERT ou CLIP.

Cependant, ces méthodes présentent des limitations :

— Rareté et diversité limitée des données d'entraînement (ex : OV-COCO avec 48 catégories).

— Charge de calcul élevée et déploiement complexe sur périphériques.

Plusieurs méthodes ont reformulé l'apprentissage de la détection d'objets en pré-entraînement visuel-langage à l'échelle régionale, entraînant des détecteurs open-vocabulary à grande échelle. Néanmoins, elles peinent encore à détecter des objets dans des scénarios réels, principalement à cause de la lourdeur des modèles et de la complexité d'inférence.

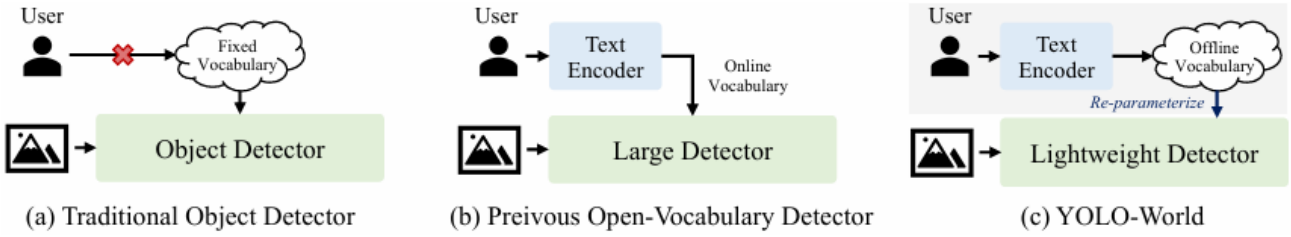

FIGURE 2 – Comparaison avec différents paradigmes de détection : (a) détecteurs traditionnels, (b) détecteurs à vocabulaire ouvert précédents, (c) YOLO-World.

Cette figure illustre clairement que les détecteurs traditionnels ne reconnaissent que les catégories fixes, tandis que les méthodes récentes à vocabulaire ouvert utilisent de grands modèles coûteux pour encoder simultanément les images et les textes. YOLO-World démontre qu'il est possible d'obtenir des performances compétitives avec des détecteurs légers en adoptant une approche efficace de type *prompt-then-detect*, où les vocabulaires sont encodés hors ligne puis réutilisés comme poids du modèle, combinant ainsi efficacité et flexibilité pour des applications réelles.

3 Méthode

3.1 Formulation de pré-entraînement : Paires Région-Texte

Nous introduisons un schéma de pré-entraînement à grande échelle basé sur des paires région-texte. L'idée est d'unifier différentes sources de données — données de détection, données de grounding et données image-texte — sous la forme de couples région-texte. Cette approche permet de renforcer la correspondance entre les représentations visuelles et linguistiques et d'améliorer la capacité du détecteur à gérer un vocabulaire large et varié. Le modèle est ainsi entraîné à associer des régions visuelles à des descriptions textuelles correspondantes via un apprentissage contrastif.

Formellement, les annotations d'instances $\Omega = \{B_i, c_i\}_{i=1}^{N}$ sont reformulées comme des paires région-texte $\Omega = \{B_i, t_i\}_{i=1}^{N}$, où t_i correspond au texte associé à la région B_i (nom de catégorie, phrases nominales ou descriptions). Le modèle prend en entrée l'image I et un ensemble de textes T et prédit des boîtes $\{\hat{B}_k\}$ et des embeddings d'objets $\{e_k\}$ ($e_k \in \mathbb{R}^D$).

3.2 Architecture du modèle

Le détecteur **YOLO-World** repose sur l'architecture YOLO standard et utilise un encodeur de texte pré-entraîné CLIP pour encoder les entrées textuelles. L'encodeur d'image extrait les caractéristiques multi-échelles. Le module **RepVL-PAN** fusionne les caractéristiques image-texte pour renforcer la représentation visuo-sémantique.

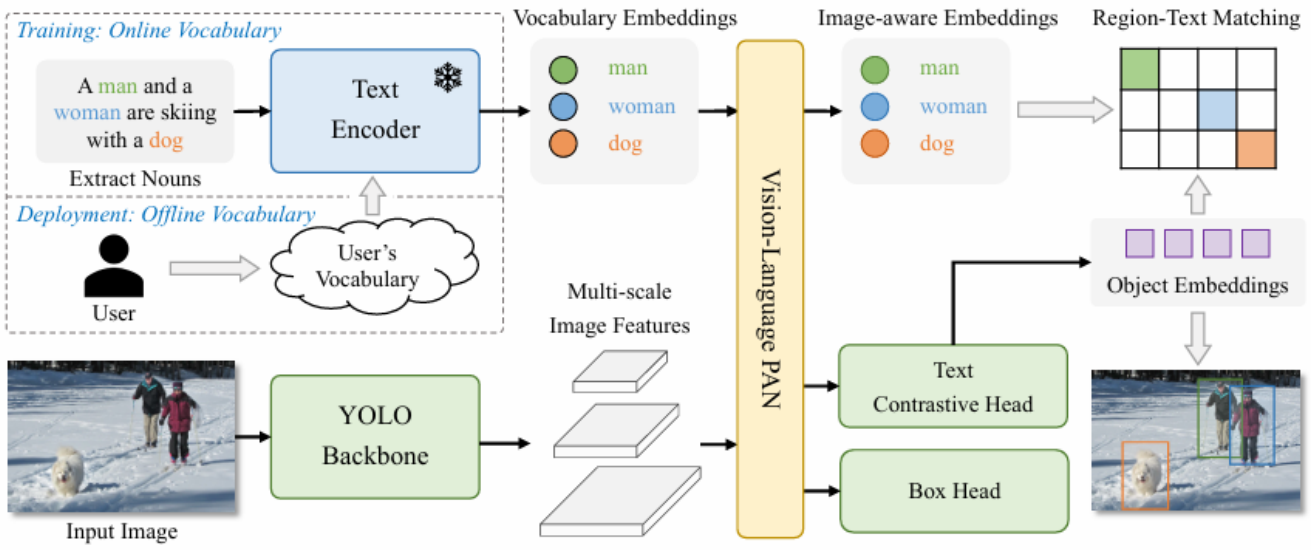

FIGURE 3 – Architecture globale de YOLO-World. Le texte est intégré en entrée et fusionné avec les caractéristiques visuelles via le RepVL-PAN.

YOLO Detector

YOLO-World est basé sur YOLOv8, avec backbone Darknet, un Path Aggregation Network (PAN) pour les pyramides multi-échelles et une tête pour la régression de boîtes et embeddings d'objets.

Text Encoder

Le texte T est encodé via l'encodeur Transformer pré-entraîné CLIP pour obtenir $W = \text{TextEncoder}(T) \in \mathbb{R}^{C \times D}$, où C est le nombre de noms et D la dimension d'embedding. Lorsqu'une légende ou expression référentielle est utilisée, un algorithme n-gram extrait les phrases nominales pour l'encodeur.

Text Contrastive Head

Nous utilisons une tête découpée avec deux convolutions 3×3 pour régresser les boîtes $\{b_k\}_{k=1}^{K}$ et embeddings $\{e_k\}_{k=1}^{K}$. La similarité objet-texte $s_{k,j}$ est calculée par :

$$s_{k,j} = \alpha \cdot \text{L2-Norm}(e_k) \cdot \text{L2-Norm}(w_j)^\top + \beta$$

avec α, β apprenables.

3.3 Réseau d'agrégation de chemins vision-langage re-paramétrisable (RepVL-PAN)

Le **RepVL-PAN** suit les chemins top-down et bottom-up pour établir les pyramides de caractéristiques $\{P_3, P_4, P_5\}$ à partir des features $\{C_3, C_4, C_5\}$. Il inclut :

— **Text-guided CSPLayer (T-CSPLayer)** : injecte l'information linguistique dans les features image multi-échelles.

— **Image-Pooling Attention (I-Pooling Attention)** : améliore les embeddings texte avec des informations visuelles.

Durant l'inférence, les embeddings du vocabulaire hors ligne sont re-paramétrisés dans les couches du RepVL-PAN pour un déploiement efficace.

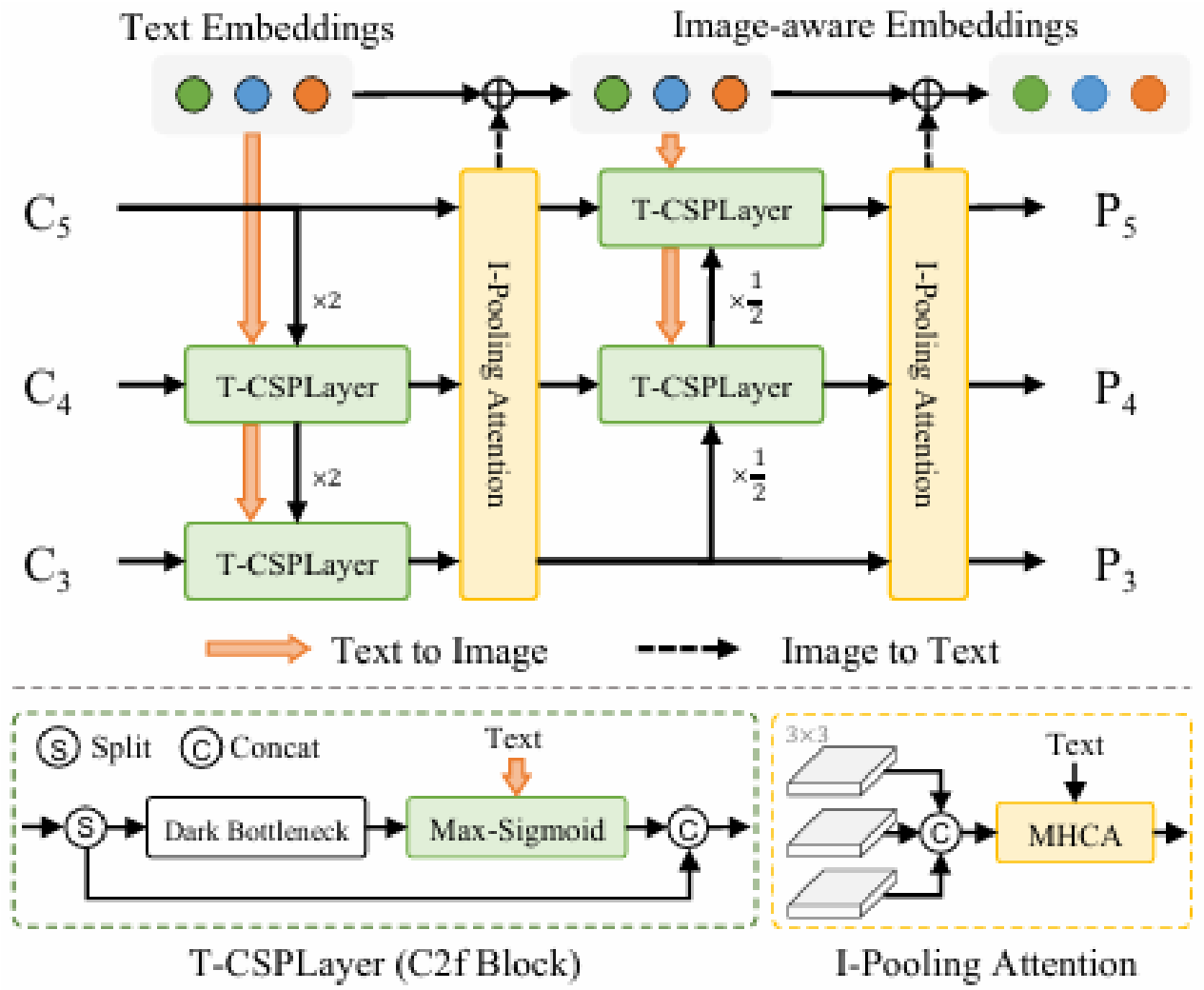

FIGURE 4 – Illustration du RepVL-PAN. Fusion multi-modale des caractéristiques image et texte via T-CSPLayer et I-Pooling Attention.

3.4 Schémas de pré-entraînement et paradigme *prompt-then-detect*

Apprentissage contrastif région-texte : Les annotations $\Omega = \{B_i, t_i\}_{i=1}^{N}$ sont utilisées pour calculer la loss contrastive entre objets et textes, combinée avec IoU loss et Distributed Focal Loss pour la régression des boîtes :

$$\mathcal{L}(I) = \mathcal{L}_{con} + \lambda_I(\mathcal{L}_{iou} + \mathcal{L}_{dfl})$$

Pseudo-labeling : Les paires région-texte sont générées automatiquement via : 1) extraction des phrases nominales, 2) pseudo-labeling avec un détecteur pré-entraîné, 3) filtrage par CLIP et NMS pour obtenir des annotations de haute qualité. Exemple : 246k images de CC3M avec 821k pseudo-annotations.

Prompt-then-detect : Pour l'inférence, les prompts utilisateurs sont encodés hors ligne en embeddings (offline vocabulary), évitant le recalcul à chaque image et permettant un déploiement rapide et flexible.

4 Expériences

Dans cette section, nous démontrons l'efficacité de YOLO-World via un pré-entraînement sur des datasets à grande échelle et une évaluation en zéro-shot sur les benchmarks LVIS et COCO. Nous étudions également les performances en fine-tuning pour la détection d'objets.

4.1 Détails d'implémentation

YOLO-World est développé sur les toolboxes MMYOLO et MMDetection. Trois variantes sont proposées selon la latence : petite (S), moyenne (M), et grande (L). L'encodeur texte CLIP est utilisé avec ses poids pré-entraînés. La vitesse d'inférence est mesurée sur une GPU NVIDIA V100 sans accélération supplémentaire.

5 Évaluation zéro-shot

Après la pré-formation, nous évaluons directement le YOLO-World proposé sur l'ensemble de données LVIS de manière zero-shot . L'ensemble de données LVIS contient 1 203 catégories d'objets, ce qui est bien plus que les catégories des ensembles de données de détection pré-entraînement et permet de mesurer les performances de détection d'un vocabulaire volumineux. Nous évaluons principalement sur LVIS minival et rapportons le Fixed AP pour comparaison. Le nombre maximal de prédictions est fixé à 1 000.

5.1 Principaux résultats sur la détection d'objets LVIS

Dans le Tableau 1, nous comparons le YOLO-World proposé avec les méthodes de pointe récentes sur le benchmark LVIS de manière zero-shot . Compte tenu de la charge de calcul et des paramètres du modèle, nous comparons principalement les méthodes basées sur des backbones plus légers, par exemple Swin-T. Il est remarquable que YOLO-World surpasse les méthodes de pointe précédentes en termes de performance zero-shot et de vitesse d'inférence. Comparé à GLIP, GLIPv2 et Grounding DINO, qui intègrent davantage de données comme Cap4M, YOLO-World pré-entraîné sur O365 et GoldG obtient de meilleures performances, même avec moins de paramètres. Comparé à DetCLIP, YOLO-World atteint des performances comparables (35,4 contre 34,4) tout en obtenant une augmentation de 20× de la vitesse d'inférence. Les résultats montrent également que les petits modèles, par exemple YOLO-World-S avec 13M paramètres, peuvent être utilisés pour la pré-formation vision-langage et obtenir de fortes capacités de vocabulaire ouvert.

TABLE 1 – Évaluation Zero-shot sur LVIS. Les FPS sont mesurés sur un GPU NVIDIA V100 sans TensorRT. Les paramètres et FPS de YOLO-World sont indiqués pour la version reparamétrée (sans parenthèses) et la version originale (entre parenthèses).

Méthode	**Backbone**	**Params**	**Pre-trained Data**	**FPS**	**AP**	**AP_r / AP_c / AP_f**
MDETR	R-101	169M	GoldG	-	24.2	20.9 / 24.3 / 24.2
GLIP-T	Swin-T	232M	O365, GoldG	0.12	24.9	17.7 / 19.5 / 31.0
GLIP-T	Swin-T	232M	O365, GoldG, Cap4M	0.12	26.0	20.8 / 21.4 / 31.0
GLIPv2-T	Swin-T	232M	O365, GoldG	0.12	26.9	- / - / -
GLIPv2-T	Swin-T	232M	O365, GoldG, Cap4M	0.12	29.0	- / - / -
Grounding DINO-T	Swin-T	172M	O365, GoldG	1.5	25.6	14.4 / 19.6 / 32.2
Grounding DINO-T	Swin-T	172M	O365, GoldG, Cap4M	1.5	27.4	18.1 / 23.3 / 32.7
DetCLIP-T	Swin-T	155M	O365, GoldG	2.3	34.4	26.9 / 33.9 / 36.3
YOLO-World-S	YOLOv8-S	13M (77M)	O365, GoldG	74.1 (19.9)	26.2	19.1 / 23.6 / 29.8
YOLO-World-M	YOLOv8-M	29M (92M)	O365, GoldG	58.1 (18.5)	31.0	23.8 / 29.2 / 33.9
YOLO-World-L	YOLOv8-L	48M (110M)	O365, GoldG	52.0 (17.6)	35.0	27.1 / 32.8 / 38.3
YOLO-World-L	YOLOv8-L	48M (110M)	O365, GoldG, CC3M†	52.0 (17.6)	35.4	27.6 / 34.1 / 38.0

FIGURE 5 – Visualisation des résultats zero-shot de YOLO-World sur LVIS.

5.2 Expériences d'ablation

Nous fournissons des études d'ablation approfondies pour analyser YOLO-World sous deux aspects principaux, à savoir ., pré-formation et architecture. Sauf indication contraire, nous menons principalement des expériences d'ablation basées sur YOLO-World-L et pré-entraînons Objects365 avec une évaluation zéro-shot sur LVIS minival

Données de pré-entraînement

Dans le tableau 3 , nous évaluons les performances du pré-entraînement de YOLO-World à l'aide de différentes données. Par rapport à la base de données entraînée sur Objects365, l'ajout de GQA peut améliorer significativement les performances, avec un gain de 8,4 AP sur LVIS. Cette amélioration peut être attribuée aux informations textuelles plus riches fournies par le jeu de données GQA, qui peuvent améliorer la capacité du modèle à reconnaître des objets à vocabulaire étendu. L'ajout d'une partie des échantillons CC3M (8 pour cent des jeux de données complets) peut également apporter un gain de 0,5 AP, avec un gain de 1,3 AP sur les objets rares. Le tableau 3 démontre que l'ajout de données supplémentaires peut améliorer efficacement les capacités de détection dans les scénarios à vocabulaire étendu. De plus, à mesure que le volume de données augmente, les performances continuent de s'améliorer, soulignant l'intérêt d'exploiter des jeux de données plus volumineux et plus diversifiés pour l'entraînement.

Pre-trained Data	AP	AP_r	AP_c	AP_f
O365	23.5	16.2	21.1	27.0
O365, GQA	31.9	22.5	29.9	35.4
O365, GoldG	32.5	22.3	30.6	36.0
O365, GoldG, CC3M†	33.0	23.6	32.0	35.5

TABLE 2 – Ablation sur les données de pré-entraînement.

RepVL-PAN

Le tableau 4 illustre l'efficacité du RepVL-PAN proposé par YOLO-World, intégrant les couches CSPLayers guidées par texte ainsi que le mécanisme d'attention par pooling d'images, pour la détection à zéro coup sur LVIS. Deux configurations de pré-entraînement sont considérées : (1) pré-entraînement sur O365 et (2) pré-entraînement sur O365 combiné avec GQA. Comparé à O365, qui ne contient que des annotations de catégories, GQA fournit des textes enrichis, notamment sous forme de syntagmes nominaux.

Comme le montre le tableau 4, le RepVL-PAN proposé améliore le modèle de référence (YOLOv8-PAN [20]) de 1,1 AP sur LVIS, avec des gains particulièrement notables pour les catégories rares (AP r

), qui sont difficiles à détecter et à reconnaître. De plus, les améliorations sont plus significatives lorsque YOLO-World est pré-entraîné avec l'ensemble de données GQA, indiquant que le RepVL-PAN fonctionne mieux lorsqu'il est alimenté par des informations textuelles riches.

GQA	T → je	je → T	AP	AP_r	AP_c	AP_f
✗	✗	✗	22.4	14,5	20.1	26.0
✗	✓	✗	23.2	15.2	20,6	27.0
✗	✓	✓	23,5	16.2	21.1	27.0
✓	✗	✗	29,7	21.0	27.1	33,6
✓	✓	✓	**31,9**	**22,5**	**29,9**	**35,4**

FIGURE 6 – Ablations sur un réseau d'agrégation de voies vision-langage reparamétrable. Nous évaluons les performances zero-shot sur LVIS du réseau d'agrégation de voies vision-langage proposé. T → Moi et moi → T désigne respectivement les couches CSPLayers guidées par texte et l'attention de regroupement d'images.

Encodeurs texte

Dans le tableau 5, nous comparons les performances de différents encodeurs de texte, à savoir BERT et CLIP-base (ViT-base). Deux configurations sont considérées lors de la pré-formation : gelé ou réglé avec précision. Le taux d'apprentissage pour le réglage fin des encodeurs de texte est fixé à 0,01 fois le taux d'apprentissage de base.

Comme le montre le tableau 5, l'encodeur CLIP obtient de meilleurs résultats que BERT, avec un gain de 10,1 AP pour les catégories rares de LVIS, grâce à son pré-entraînement sur des paires image-texte, ce qui lui confère une meilleure capacité d'intégration centrée sur la vision. Le réglage fin de BERT pendant le pré-entraînement apporte des améliorations significatives de 3,7 AP, tandis que le réglage fin de CLIP entraîne une baisse notable des performances. Cette baisse s'explique par le fait que le réglage fin sur O365 peut dégrader la capacité de généralisation du CLIP pré-entraîné, limité à 365 catégories et dépourvu d'informations textuelles riches.

Text Encoder	Frozen ?	AP	AP_r	AP_c	AP_f
BERT-base	Frozen	14.6	3.4	10.7	20.0
BERT-base	Fine-tune	18.3	6.6	14.6	23.6
CLIP-base	Frozen	22.4	14.5	20.1	26.0
CLIP-base	Fine-tune	19.3	8.6	15.7	24.8

TABLE 3 – Ablation sur l'encodeur texte.

5.3 Fine-tuning de YOLO-World

Configuration expérimentale

Prise des poids pré-entraînés, fine-tuning sur 80 epochs avec AdamW, learning rate 0.0002. L'encodeur CLIP est fine-tuné avec un facteur 0.01.

COCO Object Detection

Méthode	Pre-train	AP	AP_{50}	FPS

TABLE 4 – Fine-tuning sur COCO. O, G, C : Objects365, GoldG, CC3M†.

LVIS Object Detection

Méthode	AP	AP_r	AP_c	AP_f
YOLOv8-L	26.9	10.2	25.4	35.8
YOLO-World-L	34.1	20.4	31.1	43.5

TABLE 5 – Fine-tuning sur LVIS. YOLO-World pré-entraîné sur O+G+C.

5.4 Open-Vocabulary Instance Segmentation

Dans cette section, nous effectuons un affinage supplémentaire de YOLO-World pour la segmentation d'objets dans un contexte à vocabulaire ouvert, appelé segmentation d'instances à vocabulaire ouvert (OVIS). Les méthodes précédentes ont exploré l'OVIS en utilisant un pseudo-étiquetage pour les objets nouveaux. En revanche, étant donné les solides capacités de transfert et de généralisation de YOLO-World, nous l'affinons directement sur un sous-ensemble de données avec des annotations de masques et évaluons ses performances de segmentation dans des scénarios à grand vocabulaire.

Plus précisément, nous évaluons la segmentation d'instances à vocabulaire ouvert selon deux configurations :

COCO vers LVIS : YOLO-World est affiné sur le jeu de données COCO (80 catégories) avec des annotations de masques, ce qui oblige le modèle à transférer les connaissances de 80 catégories vers 1203 catégories ($80 \rightarrow 1203$).

LVIS-base vers LVIS : YOLO-World est affiné sur LVIS-base (866 catégories, incluant les catégories communes et fréquentes) avec des annotations de masques, ce qui oblige le modèle à transférer les connaissances de 866 catégories vers 1203 catégories ($866 \rightarrow 1203$).

Les modèles affinés sont évalués sur le jeu de validation standard LVIS val2017 avec 1203 catégories, incluant 337 catégories rares qui ne sont pas vues pendant l'affinage, fournissant ainsi une mesure des performances en vocabulaire ouvert

Model	Fine-tune Data	Modules	AP	AP_r	AP_c	AP_f
YOLO-World-L	LVIS-base	Seg Head	19.1	14.2	17.2	23.5
YOLO-World-L	LVIS-base	All	28.7	15.0	28.3	35.2

TABLE 6 – Open-Vocabulary Instance Segmentation. Seg Head : tête de segmentation. All : tous les modules fine-tunés.

6 Visualisation des résultats

Nous présentons les résultats de visualisation de YOLO-World-L pré-entraîné sous trois paramètres :

1. inférence zéro-shot sur les catégories LVIS ;
2. utilisation de prompts personnalisés avec des catégories fines et des attributs ;
3. détection par référence (referring detection).

FIGURE 7 – Visualization Results on Referring Object Detection. We explore the capability of the pre-trained YOLO-World to detect objects with descriptive noun phrases. Images are obtained from COCO val2017.

Ces visualisations montrent également que YOLO-World possède une forte capacité de généralisation dans les scénarios en vocabulaire ouvert ainsi qu'une capacité de détection par référence.

6.1 Inférence zéro-shot sur LVIS

La Figure 8 présente les résultats de visualisation basés sur les catégories LVIS, générés par YOLO-World-L pré-entraîné en zéro-shot. Le modèle pré-entraîné montre une forte capacité de

transfert zéro-shot et est capable de détecter un grand nombre d'objets dans l'image.

FIGURE 8 – Visualization Results on Zero-shot Inference on LVIS. We adopt the pre-trained YOLO-World-L and infer with the LVIS vocabulary (containing 1203 categories) on the COCO val2017 dataset.

7 Résultats et discussions

La Figure 9 présente les résultats expérimentaux de l'extension de YOLO-World pour la segmentation d'instances en vocabulaire ouvert (OVIS). Deux stratégies de fine-tuning sont considérées : (1) fine-tuning uniquement de la tête de segmentation et (2) fine-tuning de tous les modules. Avec la stratégie (1), YOLO-World conserve les capacités zéro-shot acquises lors de la pré-formation, ce qui lui permet de généraliser aux catégories non vues sans ajustements supplémentaires. La stratégie (2) permet à YOLO-World de mieux s'adapter au jeu de données LVIS, mais peut entraîner une dégradation des capacités zéro-shot.

La Figure 9 compare les performances de YOLO-World selon différents jeux de données de fine-tuning (COCO ou LVIS-base) et stratégies (tête de segmentation uniquement ou tous les modules). Tout d'abord, le fine-tuning sur LVIS-base permet d'obtenir de meilleures performances que sur COCO. Cependant, le rapport entre AP et AP_r (AP_r/AP) reste presque inchangé, par exemple 76,5 % pour COCO et 74,3 % pour LVIS-base. Étant donné que le détecteur est gelé, nous attribuons cet écart de performance au fait que LVIS fournit des annotations de segmentation plus détaillées et plus denses, favorisant l'apprentissage de la tête de segmentation.

Lorsque tous les modules sont fine-tunés, YOLO-World obtient des améliorations remarquables sur LVIS, par exemple YOLO-World-L gagne 9,6 AP. Cependant, ce fine-tuning complet peut dégrader les performances en vocabulaire ouvert, entraînant une baisse de 0,6 AP_r pour YOLO-World-L.

Model	Fine-tune Data	Fine-tune Modules	AP	AP_r	AP_c	AP_f	AP^b	$AP_r{}^b$
YOLO-World-M	**COCO**	*Seg Head*	12.3	9.1	10.9	14.6	22.3	16.2
YOLO-World-L	**COCO**	*Seg Head*	16.2	12.4	15.0	19.2	25.3	**18.0**
YOLO-World-M	**LVIS-base**	*Seg Head*	16.7	12.6	14.6	20.8	22.3	16.2
YOLO-World-L	**LVIS-base**	*Seg Head*	19.1	14.2	17.2	23.5	25.3	**18.0**
YOLO-World-M	**LVIS-base**	*All*	25.9	13.4	24.9	32.6	32.6	15.8
YOLO-World-L	**LVIS-base**	*All*	**28.7**	**15.0**	**28.3**	**35.2**	**36.2**	17.4

FIGURE 9 – Open-Vocabulary Instance Segmentation. We evaluate YOLO-World for open-vocabulary instance segmentation under the two settings. We fine-tune the segmentation head or all modules of YOLO-World and report Mask AP for comparison. AP_b denotes the box AP.

8 Conclusion

Nous présentons YOLO-World, un détecteur à vocabulaire ouvert en temps réel de pointe, visant à améliorer l'efficacité et les capacités à vocabulaire ouvert dans des applications réelles. Dans cet article, nous avons remodelé les architectures YOLO prévalentes en une architecture YOLO vision-langage pour le pré-entraînement et la détection à vocabulaire ouvert, et proposé RepVL-PAN, qui relie les informations visuelles et linguistiques au sein du réseau et peut être re-paramétré pour un déploiement efficace. Nous présentons également des schémas de pré-entraînement efficaces utilisant des données de détection, de grounding et image-texte, afin de doter YOLO-World de solides capacités de détection à vocabulaire ouvert. Les expériences démontrent la supériorité de YOLO-World en termes de rapidité et de performance à vocabulaire ouvert, et indiquent l'efficacité du pré-entraînement vision-langage sur de petits modèles, fournissant des perspectives pour les recherches futures. Nous espérons que YOLO-World pourra servir de nouveau benchmark pour la détection à vocabulaire ouvert en conditions réelles.

Bibliographie

[1] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, pp. 213–229, 2020.

[2] K. Chen, J. Wang, J. Pang, et al., "MMDetection : Open mmlab detection toolbox and benchmark," *arXiv preprint* arXiv :1906.07155, 2019.

[3] MMYOLO Contributors, "MMYOLO : OpenMMLab YOLO series toolbox and benchmark," GitHub repository, 2022. Available : https://github.com/open-mmlab/mmyolo

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT : pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, pp. 4171–4186, 2019.

[5] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg : Making vgg-style convnets great again," in *CVPR*, pp. 13733–13742, 2021.

[6] Y. Du, F. Wei, Z. Zhang, M. Shi, Y. Gao, and G. Li, "Learning to prompt for open-vocabulary object detection with vision-language model," in *CVPR*, pp. 14064–14073, 2022.

[7] C. Feng, Y. Zhong, Y. Gao, M. Scott, and W. Huang, "TOOD : task-aligned one-stage object detection," in *ICCV*, pp. 3490–3499, 2021.

[8] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX : exceeding YOLO series in 2021," *arXiv preprint* arXiv :2107.08430, 2021.

[9] R. Girshick, "Fast R-CNN," in *ICCV*, pp. 1440–1448, 2015.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, pp. 580–587, 2014.

[11] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation," in *ICLR*, 2022.

[12] A. Gupta, P. Dollár, and R. Girshick, "LVIS : A dataset for large vocabulary instance segmentation," in *CVPR*, pp. 5356–5364, 2019.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, pp. 770–778, 2016.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *ICCV*, pp. 2980–2988, 2017.

[15] D. Hudson and C. Manning, "GQA : A new dataset for real-world visual reasoning and compositional question answering," in *CVPR*, pp. 6700–6709, 2019.

[16] C. Jia, Y. Yang, Y. Xia, et al., "Scaling up visual and vision-language representation learning with noisy text supervision," in *ICML*, pp. 4904–4916, 2021.

[17] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," GitHub repository, 2023. Available : https://github.com/ultralytics/ultralytics

[18] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion, "MDETR : Modulated detection for end-to-end multi-modal understanding," in *ICCV*, pp. 1760–1770, 2021.

[19] W. Kuo, Y. Cui, X. Gu, A. Piergiovanni, and A. Angelova, "F-VLM : open-vocabulary object detection upon frozen vision and language models," *arXiv preprint* arXiv :2209.15639, 2022.

[20] C. Li, L. Li, H. Jiang, et al., "YOLOv6 : A single-stage object detection framework for industrial applications," *arXiv preprint* arXiv :2209.02976, 2022.