

Adaptation de Yolov8 pour la détection d'objets avec peu d'exemples

Conférence **N**ationale en **I**ntelligence **A**rtificielle (**CNIA**) 2024

Guillaume Fourret, Christophe Fiorio, Gérard Subsol, Marc Chaumont

Contact : guillaume.fourret@lirmm.fr

4 juillet 2024



1. Introduction

Dans un cas réel : Beaucoup d'objets à détecter !



Problème du **Few-Shot Object Detection (FSOD)** : Apprentissage de nouveaux objets en N -way (classes) K -shots (bounding box labélisées).

2. Meta learning vs Transfer Learning

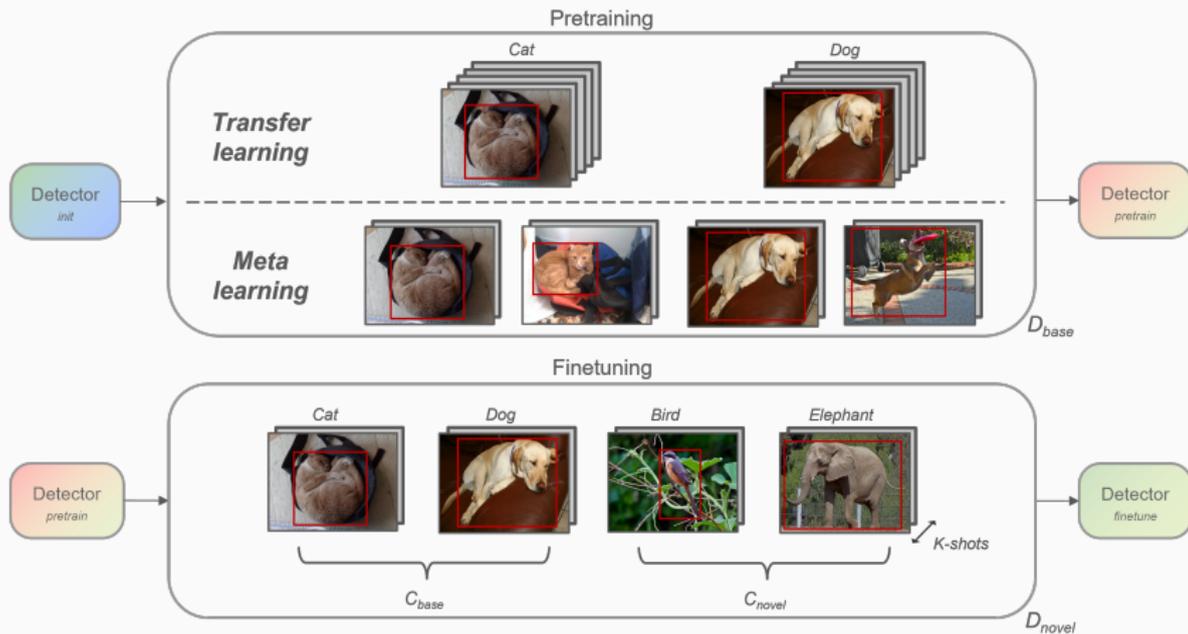
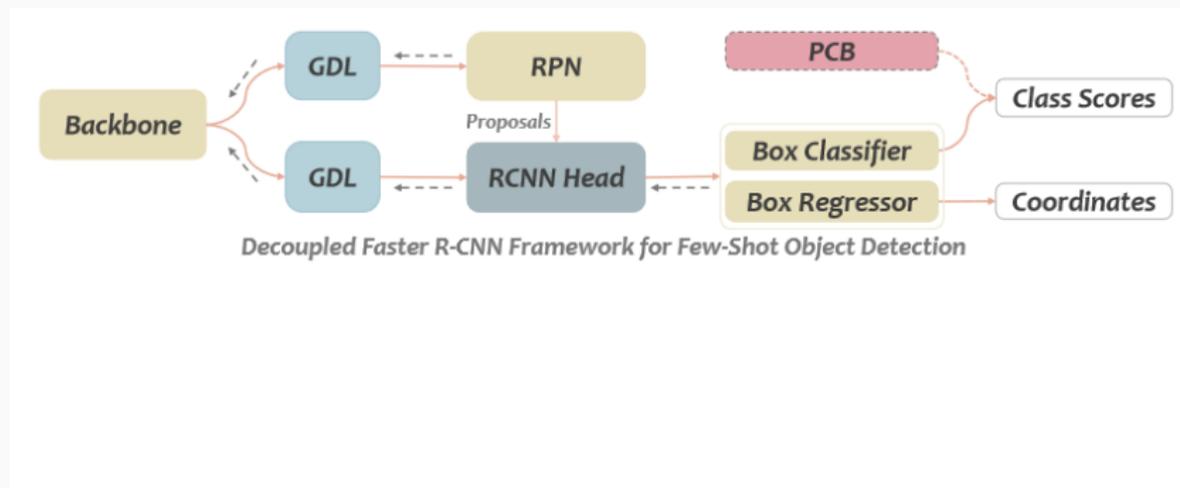


Illustration des deux paradigmes du FSOD.

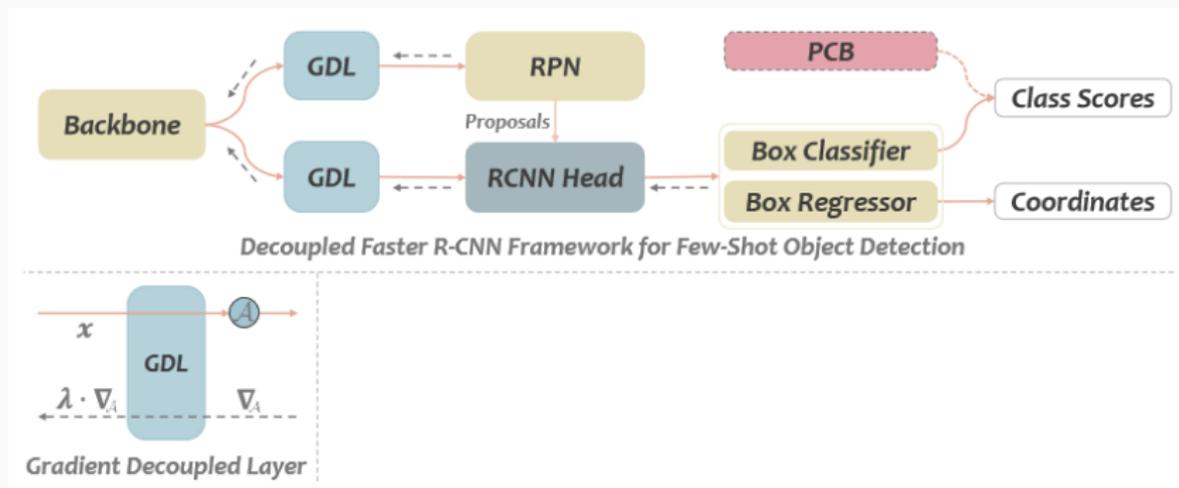
2. Fonctionnement de DeFRCN



Modules GDL et PCB dans DeFRCN¹.

¹al, Limeng Qiao et 2021. "Defrcn: Decoupled faster r-cnn for few-shot object detection" ICCV.

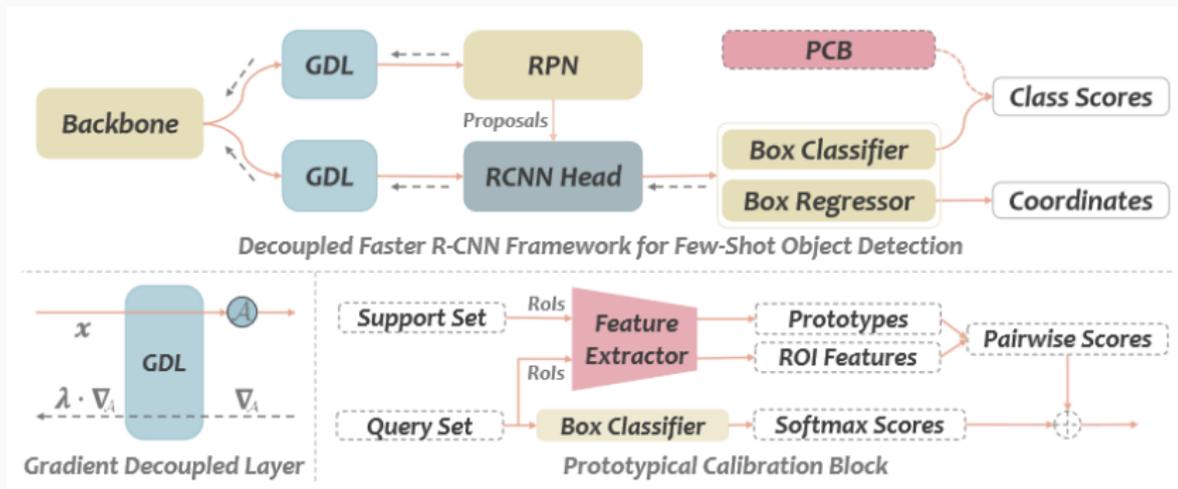
2. Fonctionnement de DeFRCN



Modules GDL et PCB dans DeFRCN¹.

¹al, Limeng Qiao et 2021. "Defrcn: Decoupled faster r-cnn for few-shot object detection" ICCV.

2. Fonctionnement de DeFRCN



Modules GDL et PCB dans DeFRCN¹.

¹al, Limeng Qiao et 2021. "Defrcn: Decoupled faster r-cnn for few-shot object detection" ICCV.

3. Yolov8 et adaptation au FSOD

- Les techniques de FSOD utilisent le Faster-RCNN (two-stage), ...

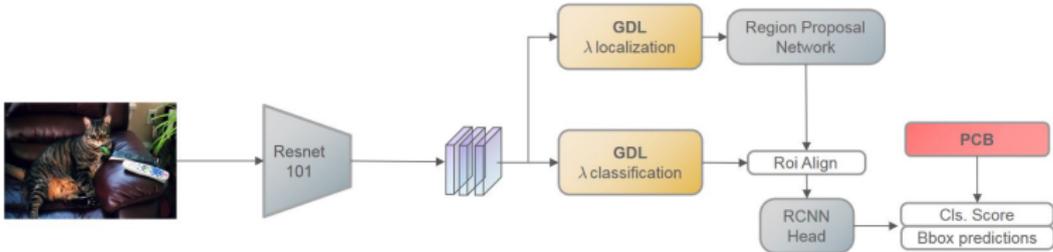
3. Yolov8 et adaptation au FSOD

- Les techniques de FSOD utilisent le Faster-RCNN (two-stage), ...
- Mais, les algorithmes YOLO (one-stage) obtiennent maintenant de meilleurs résultats en supervisé classique, avec un temps d'inférence beaucoup plus rapide !

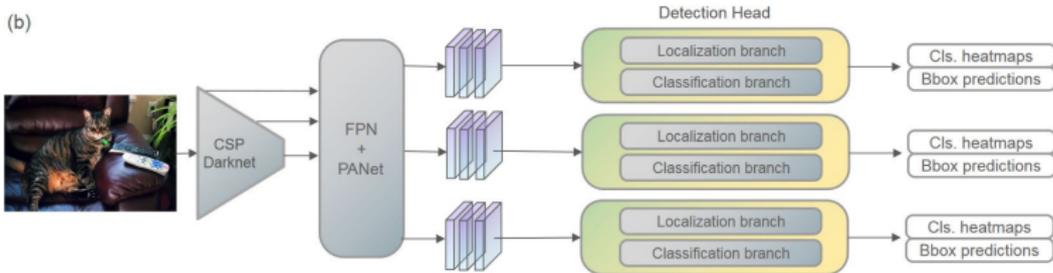
3. Yolov8 et adaptation au FSOD

Comparaison entre (a) DeFRCN et (b) Yolov8 :

(a)

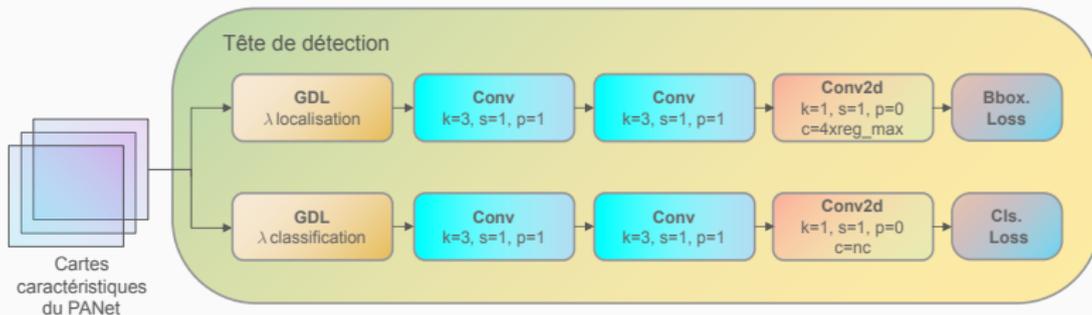


(b)



3. Yolo_DeFRCN : Intégration du GDL

Modification des têtes de YOLOv8 :

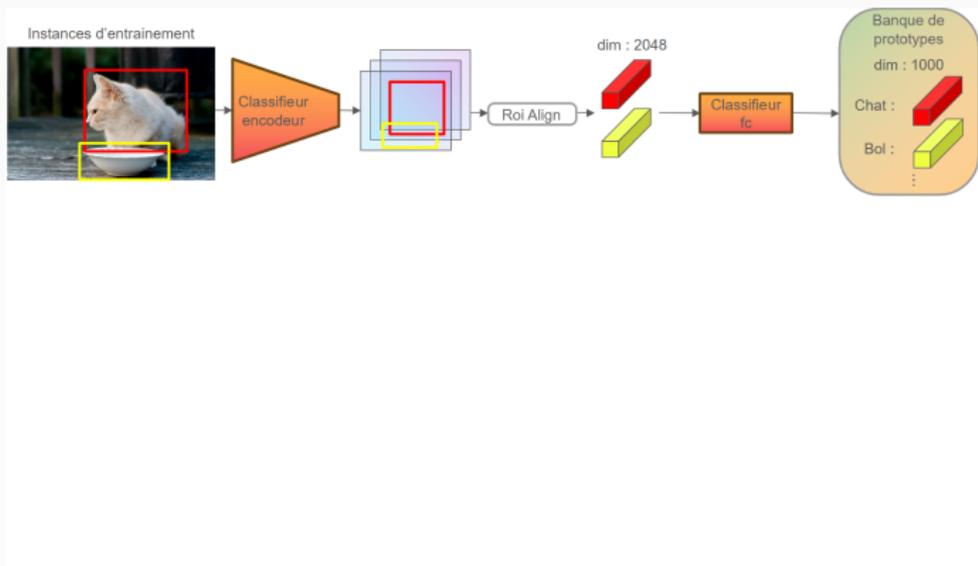


La rétropropagation devient alors :

$$\theta_{PANet} \leftarrow \theta_{PANet} - \gamma \left(\lambda_{loc} \frac{\partial \mathcal{L}_{loc}}{\partial \theta_{PANet}} + \lambda_{cls} \frac{\partial \mathcal{L}_{cls}}{\partial \theta_{PANet}} \right) \quad (1)$$

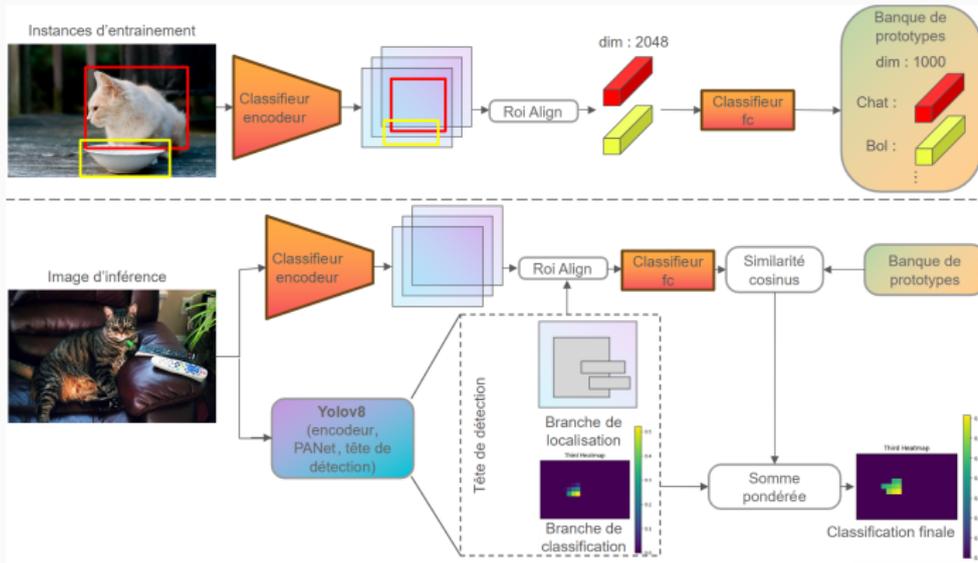
3. Yolo_DeFRCN : Intégration du module PCB

Le module PCB est utilisé durant l'inférence :



3. Yolo_DeFRCN : Intégration du module PCB

Le module PCB est utilisé durant l'inférence :



4. MSCOCO pour le FSOD

Utilisation du benchmark MSCOCO pour le FSOD² :

- 60 des 80 classes de MSCOCO sont considérées comme classe de base, utilisées pendant le pré-entraînement.

²Wang, Xin et al. 2020. "Frustratingly Simple Few-Shot Object Detection" *ICML*.

4. MSCOCO pour le FSOD

Utilisation du benchmark MSCOCO pour le FSOD² :

- 60 des 80 classes de MSCOCO sont considérées comme classe de base, utilisées pendant le pré-entraînement.
- Les 20 dernières sont les nouvelles classes à apprendre pendant la phase de finetuning K -shots ($K=1,2,3,5,10,30$).

²Wang, Xin et al. 2020. "Frustratingly Simple Few-Shot Object Detection" *ICML*.

4. MSCOCO pour le FSOD

Utilisation du benchmark MSCOCO pour le FSOD² :

- 60 des 80 classes de MSCOCO sont considérées comme classe de base, utilisées pendant le pré-entraînement.
- Les 20 dernières sont les nouvelles classes à apprendre pendant la phase de finetuning K -shots ($K=1,2,3,5,10,30$).
- Les métriques utilisées sont le bAP50/bAP5095 et le nAP50/nAP5095 ("base" et "novel" Average Precision).

²Wang, Xin et al. 2020. "Frustratingly Simple Few-Shot Object Detection" *ICML*.

4. MSCOCO : Finetuning K-shots

Résultats sur MSCOCO avec module GDL :

	<i>3-shots</i>				<i>5-shots</i>			
	bAP50	bAP5095	nAP50	nAP5095	bAP50	bAP5095	nAP50	nAP5095
Yolov8_vanilla	19.1	13.4	9.3	5.6	19.3	13.6	11.7	7.4
Yolov8_GDL	56.0	40.7	15.1	10.2	53.2	38.2	21.3	14.0
DeFRCN_GDL	50.6	33.1	21.9	12.3	51.5	33.6	26.1	14.2

⇒ Yolov8_vanilla obtient de bien moins bonnes performances.

4. MSCOCO : Finetuning K-shots

Résultats sur MSCOCO avec module GDL :

	<i>3-shots</i>				<i>5-shots</i>			
	bAP50	bAP5095	nAP50	nAP5095	bAP50	bAP5095	nAP50	nAP5095
Yolov8_vanilla	19.1	13.4	9.3	5.6	19.3	13.6	11.7	7.4
Yolov8_GDL	56.0	40.7	15.1	10.2	53.2	38.2	21.3	14.0
DeFRCN_GDL	50.6	33.1	21.9	12.3	51.5	33.6	26.1	14.2

⇒ Yolov8_vanilla obtient de bien moins bonnes performances.

⇒ Yolov8_GDL est meilleur que DeFRCN_GDL sur les classes de base, mais légèrement moins bon sur les nouvelles.

4. MSCOCO : Finetuning K-shots

Résultats sur MSCOCO avec module GDL :

	<i>3-shots</i>				<i>5-shots</i>			
	bAP50	bAP5095	nAP50	nAP5095	bAP50	bAP5095	nAP50	nAP5095
Yolov8_vanilla	19.1	13.4	9.3	5.6	19.3	13.6	11.7	7.4
Yolov8_GDL	56.0	40.7	15.1	10.2	53.2	38.2	21.3	14.0
DeFRCN_GDL	50.6	33.1	21.9	12.3	51.5	33.6	26.1	14.2

⇒ Yolov8_vanilla obtient de bien moins bonnes performances.

⇒ Yolov8_GDL est meilleur que DeFRCN_GDL sur les classes de base, mais légèrement moins bon sur les nouvelles.

⇒ 2x moins de paramètres (25M vs 51M) et 10x plus de FPS.

4. MSCOCO : Finetuning K-shots

Ajout du module PCB à Yolov8 et DeFRCN :

	<i>3-shots</i>				<i>5-shots</i>			
	bAP50	bAP5095	nAP50	nAP5095	bAP50	bAP5095	nAP50	nAP5095
Yolov8_vanilla	19.1	13.4	9.3	5.6	19.3	13.6	11.7	7.4
Yolov8_DeFRCN	55.9 (-0.1)	40.4 (-0.3)	15.9 (+0.8)	10.8 (+0.6)	53.0 (-0.2)	38.1 (-0.1)	23.0 (+1.7)	15.0 (+1.0)
DeFRCN	49.8 (-0.8)	32.5 (-0.6)	24.0 (+2.1)	13.4 (+0.9)	50.6 (-0.9)	33.1 (-0.5)	28.4 (+2.3)	15.3 (+1.1)

⇒ Diminue légèrement les classes de base (déjà bien apprises).

⇒ Augmente plus significativement les nouvelles classes.

5. Biais : Méthodologie d'évaluation sur MSCOCO

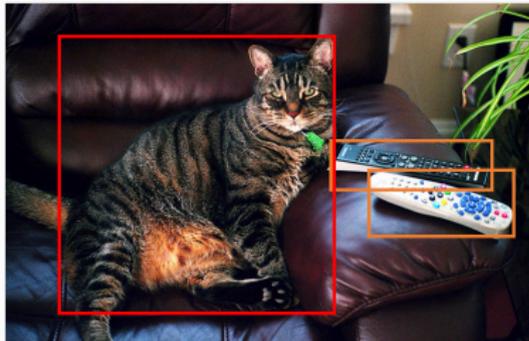
Biais de l'évaluation sur MSCOCO :

- Biais défavorable : des instances des nouvelles classes apparaissent déjà pendant le pré-entraînement sans être annotées.

Pré-entraînement (60 classes)



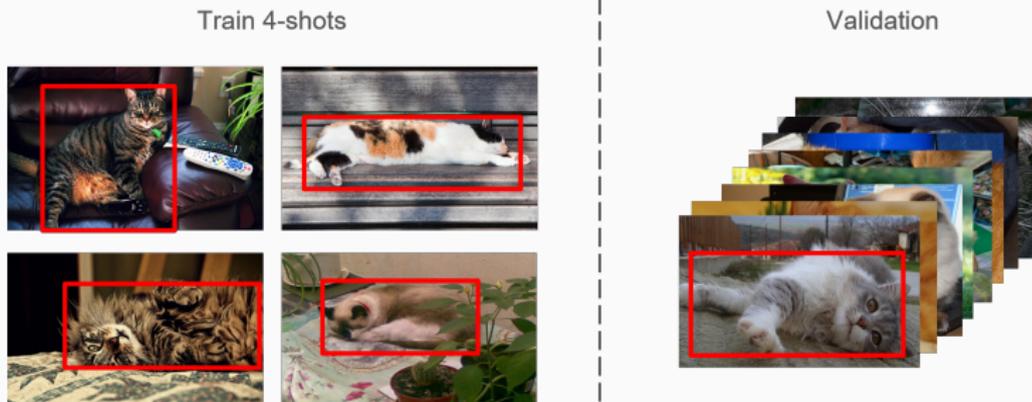
Fine-tune K-shot (80 classes)



5. Biais : Méthodologie d'évaluation sur MSCOCO

Biais de l'évaluation sur MSCOCO :

- Biais favorable : L'ensemble de validation contient des instances des nouvelles classes.



⇒ -9.7 bAP50 et -2.4 nAP50 sans validation.

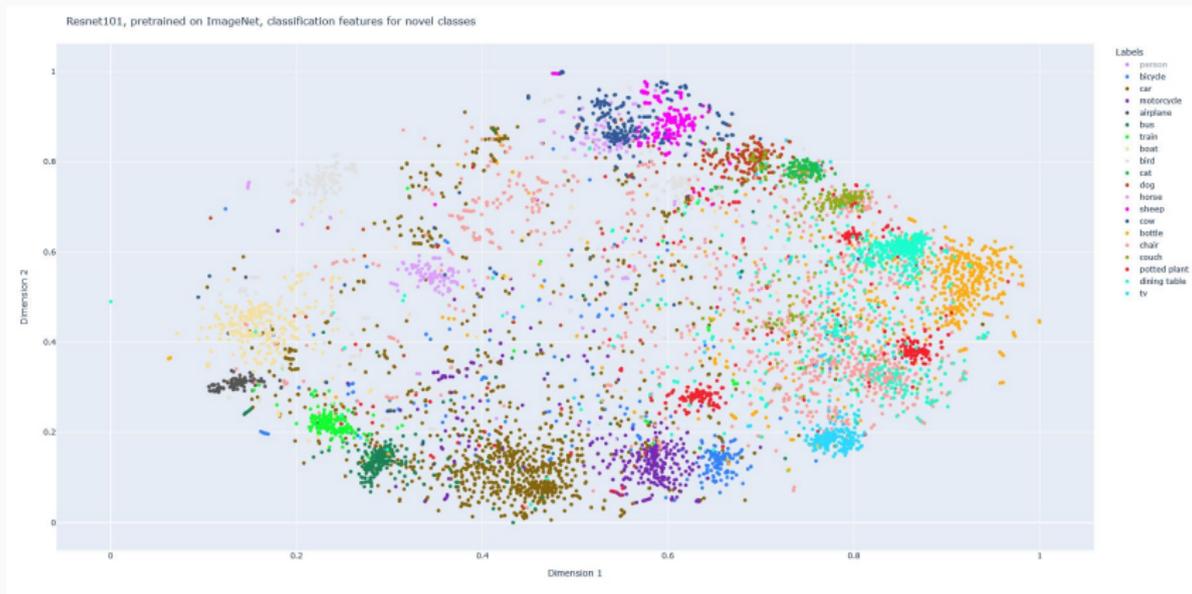
5. Extracteurs pré-entraînés

- Yolov8 est pré-entraîné sur 60 classes de *MSCOCO*...
- MAIS utilise le module PCB avec un Resnet pré-entraîné sur *ImageNet* :



5. Extracteurs pré-entraînés

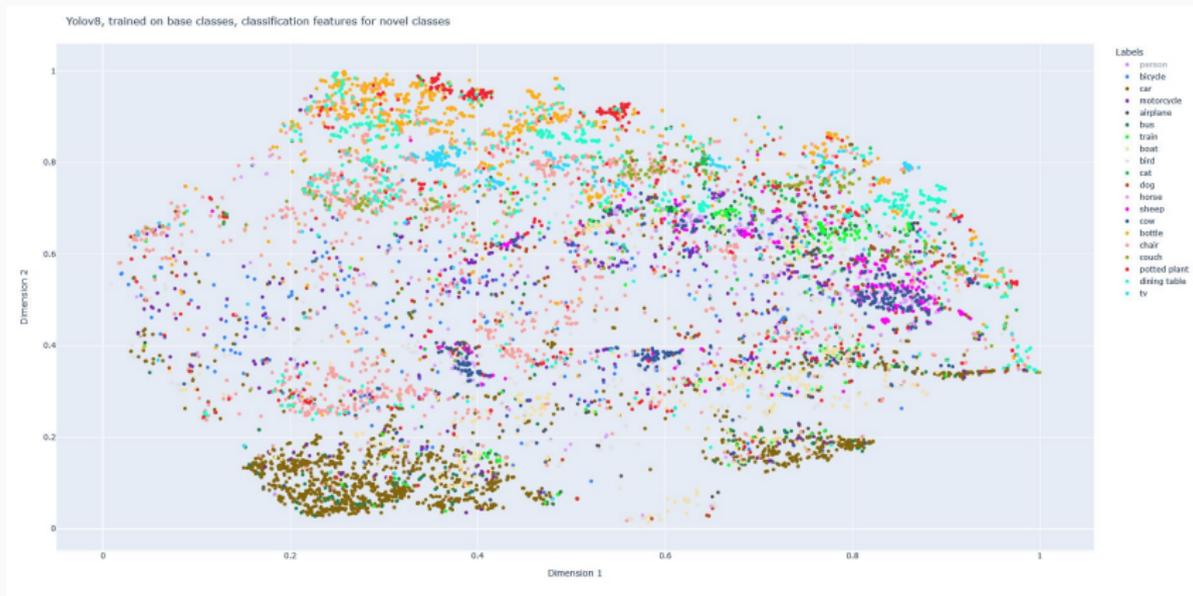
Biais avec les extracteurs de caractéristiques pré-entraîné :



Visualisation t-SNE des features des nouvelles classes par Resnet101 après pré-entraînement sur ImageNet.

5. Extracteurs pré-entraînés

Biais avec les extracteurs de caractéristiques pré-entraîné :



Visualisation t-SNE des features des nouvelles classes par Yolov8 après pré-entraînement sur les 60 classes de base.

6. Conclusion

1. L'intégration des modules GDL et PCB dans Yolov8 améliore ses performances en FSOD.

6. Conclusion

1. L'intégration des modules GDL et PCB dans Yolov8 améliore ses performances en FSOD.
2. Yolov8_DeFRCN possède 2 fois moins de paramètres et 10 fois plus de FPS.

6. Conclusion

1. L'intégration des modules GDL et PCB dans Yolov8 améliore ses performances en FSOD.
2. Yolov8_DeFRCN possède 2 fois moins de paramètres et 10 fois plus de FPS.
3. Des biais influencent l'évaluation sur MSCOCO et peuvent poser soucis dans un cas réel.

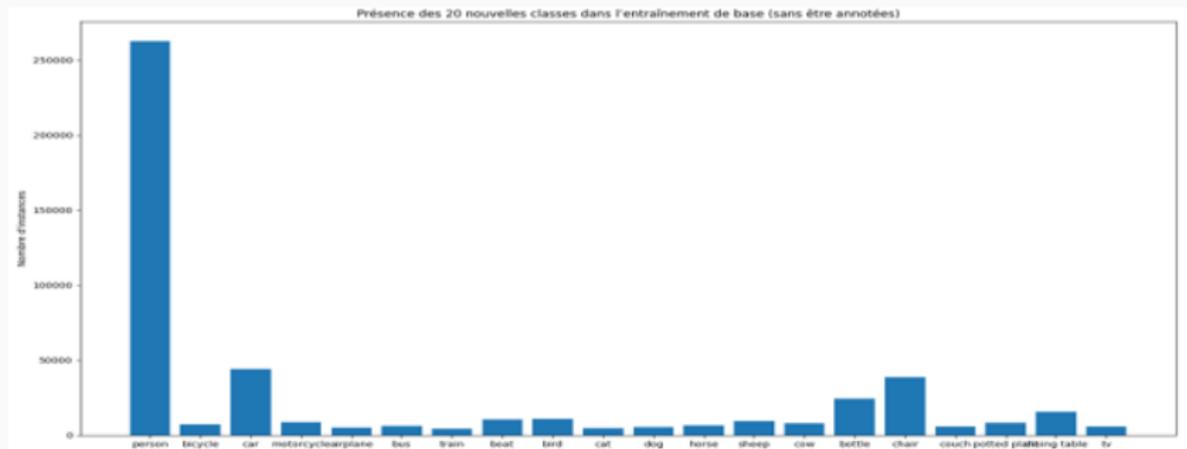
6. Conclusion

1. L'intégration des modules GDL et PCB dans Yolov8 améliore ses performances en FSOD.
2. Yolov8_DeFRCN possède 2 fois moins de paramètres et 10 fois plus de FPS.
3. Des biais influencent l'évaluation sur MSCOCO et peuvent poser soucis dans un cas réel.
4. Les méthodes utilisant du Self-Supervised Learning (SSL) sont également prometteuses dans le FSOD mais très peu explorées avec Yolo.

Merci de votre écoute !

-  al, Limeng Qiao et (2021). “**Defrcn: Decoupled faster r-cnn for few-shot object detection**”. In: *ICCV*, pp. 8681–8690.
-  Wang, Xin et al. (2020). “**Frustratingly Simple Few-Shot Object Detection**”. In: *ICML* 119, pp. 9919–9928.

Impact de l'apparition des nouvelles classes pendant le pré-entraînement sans être annotées :



Impact de l'apparition des nouvelles classes pendant le pré-entraînement sans être annotées :

