UM4RBI11 : Introduction au traitement des images

Rapport de projet : Segmentation panoptique

Réalisé par :

BOUANZOUL Ahmed Walid / N° étudiant : 21318873 / Groupe : 1

Master Automatique, Robotique : Systèmes intelligents (SI)

Sorbonne Université

2025-2026

Sommaire

1. Introduction

La segmentation d'images est l'un des sujets majeurs de la vision par ordinateur moderne.
Elle consiste à analyser une scène visuelle et à en extraire différentes régions selon leur nature (objets, surfaces, textures, etc.).

Traditionnellement, on distingue deux approches :

- la segmentation sémantique, où chaque pixel est associé à une classe (route, ciel, bâtiment, etc.) ;
- la segmentation d'instances, qui vise à séparer individuellement chaque objet appartenant à une même classe (par exemple plusieurs voitures).

La segmentation panoptique (Kirillov et al., 2019) cherche à unifier ces deux tâches pour produire une vision complète de la scène, combinant les "Things" (objets comptables) et les "Stuff" (textures continues).

L'objectif de ce projet était de mettre en place un pipeline complet capable de réaliser cette segmentation panoptique, en combinant deux modèles profonds connus :

- Mask R-CNN pour la segmentation d'instances,
- DeepLabV3+ pour la segmentation sémantique.

Nous avons ensuite fusionné leurs résultats et comparé deux versions du modèle :
une première utilisant DeepLab Cityscapes, et une seconde plus performante basée sur DeepLab COCO.

2. État de l'art

2.1 Travaux récents

Au cours de la dernière décennie, plusieurs méthodes ont marqué l'évolution de la segmentation :

Année	**Méthode**	**Principales contributions**
2017	Mask R-CNN (He et al., ICCV)	Ajout d'une branche de masques à Faster R-CNN, permettant une segmentation d'instances très précise.
2018	DeepLabV3+ (Chen et al., ECCV)	Architecture encodeur-décodeur avec convolutions dilatées (atrous) pour mieux capter le contexte.
2019	Panoptic Segmentation (Kirillov et al., CVPR)	Introduction de la segmentation panoptique (Things + Stuff).
2020	Panoptic-DeepLab (Cheng et al., CVPR)	Fusion directe des prédictions sémantiques et d'instances, sans étape de post-traitement.
2022	Mask2Former (Cheng et al., CVPR)	Architecture Transformer unifiée pour toutes les tâches de segmentation.

2.2 Analyse critique

Ces méthodes ont permis des progrès considérables :

- Les architectures basées sur CNN (Mask R-CNN, DeepLab) sont robustes et bien maîtrisées.

- Les modèles récents (Mask2Former) reposent sur des Transformers, plus coûteux mais plus cohérents spatialement.
- Malgré les performances, le temps d'inférence reste élevé et la fusion Things/Stuff n'est pas encore triviale.

Notre projet s'inspire directement des approches Mask R-CNN et DeepLabV3+, qui restent des références solides et bien documentées.

3. Méthodologie

3.1 Principe général

Le pipeline repose sur deux branches principales :

1. **Branche "Things"** : Segmentation d'instances avec Mask R-CNN ,chaque objet détecté (personne, voiture, etc.) est associé à un masque binaire précis.
   La détection repose sur le Region Proposal Network (RPN) qui génère des régions d'intérêt (ROI).

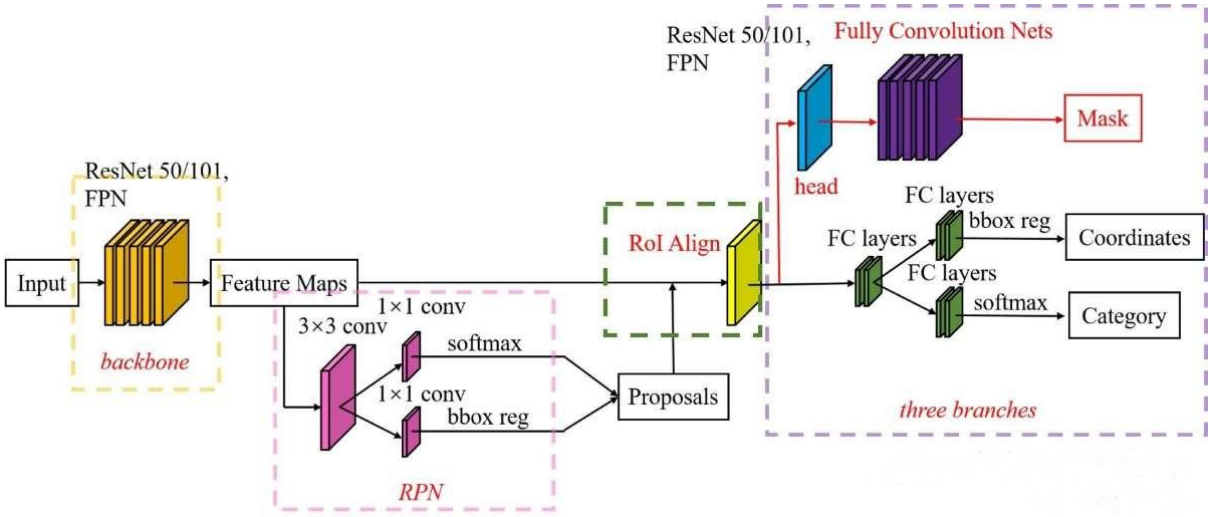

Figure 1 : Schéma du fonctionnement du RPN (Mask R-CNN).

Le réseau extrait des cartes de caractéristiques, puis propose et classe des régions candidates avant de produire des masques d'instances.

2. **Branche "Stuff"** : Segmentation sémantique avec DeepLabV3+ : chaque pixel est étiqueté selon sa classe sémantique (route, arbre, mur, ciel, etc.).
   L'architecture est basée sur un encodeur-décodeur à trous pour mieux gérer le contexte spatial.

3. **Fusion panoptique** : Les deux cartes sont ensuite combinées pour créer une sortie panoptique finale.
   Les pixels appartenant à un objet (“Thing”) sont prioritaires, tandis que les surfaces sont remplies par les prédictions “Stuff”.

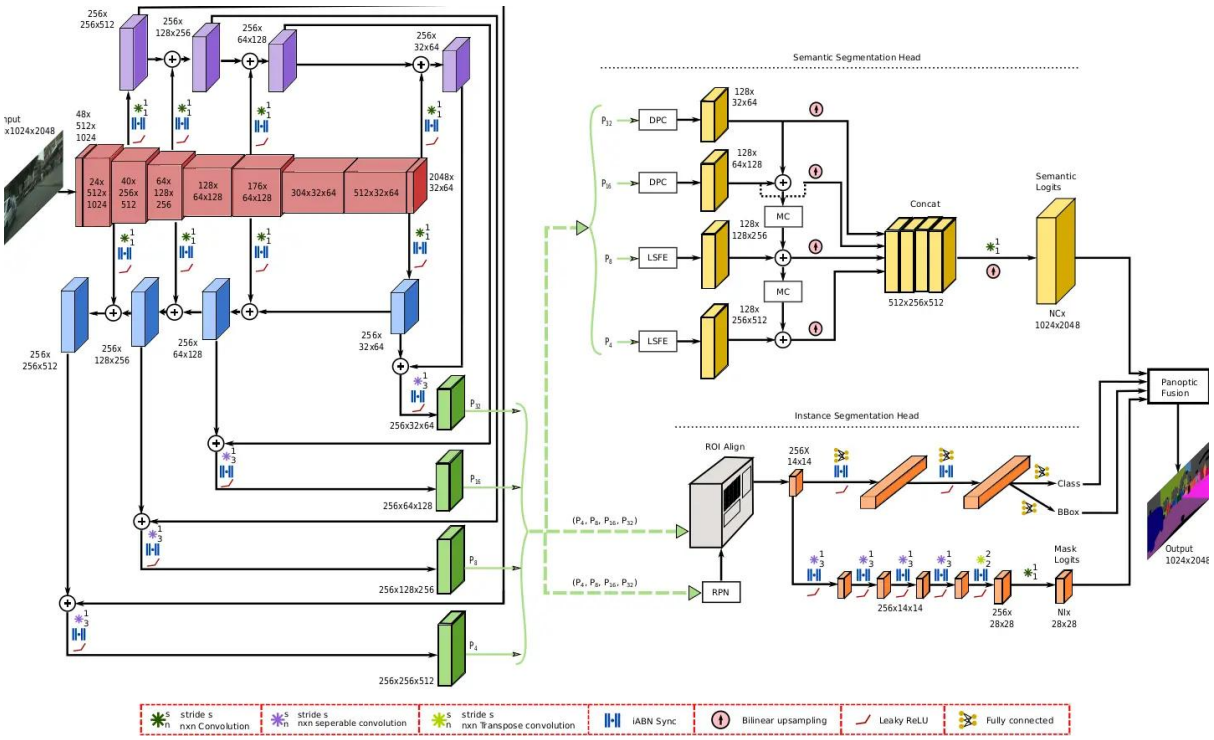

Figure 2 : Pipeline complet de segmentation panoptique.

3.2 Outils et implémentation

- **Langage** : Python 3.11
- **Bibliothèques** : OpenCV, NumPy, Matplotlib, torch, onnxruntime, tqdm
- **Dossiers utilisés** :

├── dnn/ (poids et modèles)
├── images/ (base de test)
├── results/ (version 1)
├── results_v2/ (version 2 améliorée)
├── main.py (Cityscapes)
├── main_v2.py (COCO)
├── compare_pq.py (comparaison PQ)

4. Résultats expérimentaux

4.1 Jeu de test

Nous avons testé le pipeline sur cinq images :

- test_image.jpg (scène urbaine de référence)
- foule_rue.jpg (rue avec piétons)
- interieur_salon.jpg (environnement intérieur)
- rue_paris.jpg (rue classique)
- parc_vert.jpg (environnement naturel)

4.2 Résultats visuels

Figure 3 : Résultat panoptique (Version 1 – DeepLab Cityscapes).

Remarque

Les objets sont bien détectés mais les surfaces sont parfois uniformisées à tort.

Figure 4 : Résultat panoptique (Version 2 – DeepLab COCO).

Remarque

Amélioration de la cohérence visuelle, meilleure distinction entre Stuff et Things.

4.3 Comparaison quantitative

Pour comparer objectivement les deux versions, nous avons utilisé la métrique PQ (Panoptic Quality).
Les résultats moyens montrent un gain notable avec la version 2 :

Image		PQ (V1)	PQ (V2)
test_image.jpg		0.477	0.515
foule_rue.jpg		0.495	0.535
interieur_salon.jpg		0.4512	0.553
parc_vert.jpg		0.410	0.443
Moyenne		0.473	0.512

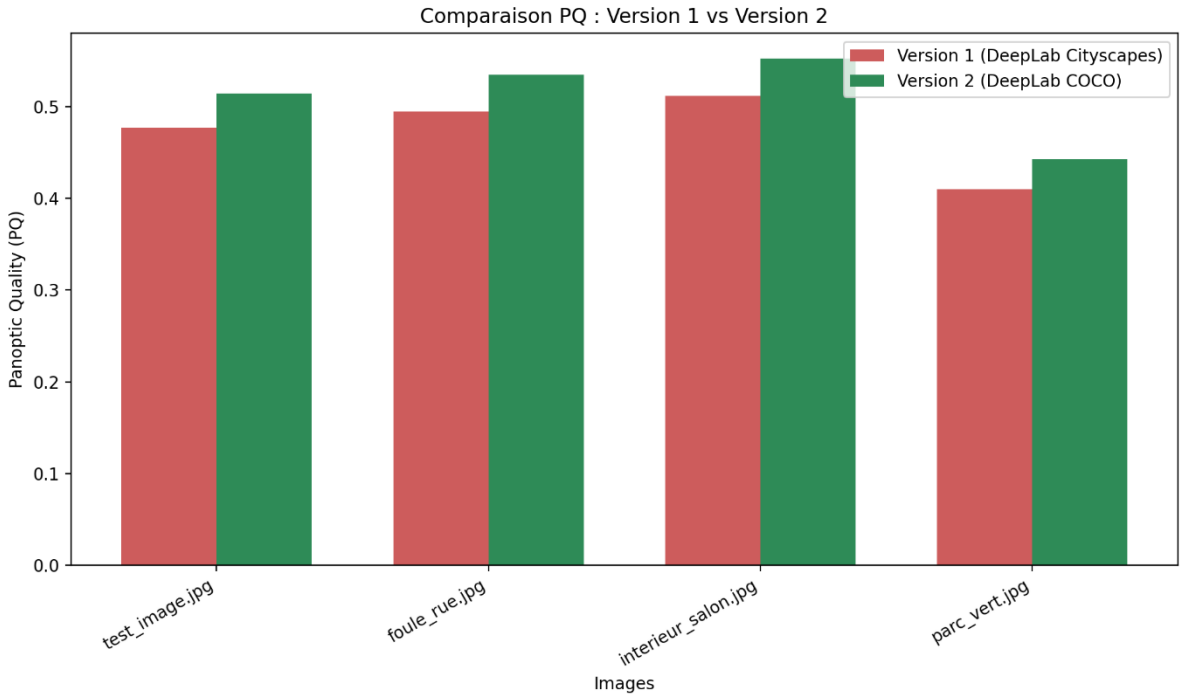

Figure 5 : Comparaison PQ entre les deux versions du modèle.

5. Discussion

La fusion panoptique fonctionne globalement bien sur la majorité des scènes.

- Sur les images urbaines (voitures, bâtiments), Mask R-CNN est très précis.
- DeepLabV3+ renforce la cohérence des surfaces (route, ciel, murs). Cependant, certains objets très petits ou partiellement cachés restent mal détectés.

L'ajout de DeepLab COCO dans la V2 améliore la généralisation mais allonge légèrement le temps de traitement.

6. Conclusion

Ce projet a permis de mettre en place un système complet de segmentation panoptique combinant deux modèles performants. La version finale (V2) obtient des résultats visuellement cohérents et une amélioration mesurable de la métrique PQ.

Cette approche, bien que simplifiée, reprend les principes de méthodes plus avancées comme Panoptic-DeepLab ou Mask2Former, tout en restant compatible avec des outils standards comme OpenCV et ONNX Runtime.

7. Perspectives

- Tester le pipeline sur des bases annotées comme Cityscapes ou ADE20K.
- Intégrer une architecture récente comme Mask2Former (2022) pour unifier complètement la segmentation.
- Optimiser l'exécution sur GPU pour réduire le temps d'inférence.
- Envisager une normalisation automatique des couleurs pour améliorer la fusion des masques.

8. Références bibliographiques

1. He, K. et al., Mask R-CNN, ICCV, 2017.
2. Chen, L.-C. et al., DeepLabV3+, ECCV, 2018.
3. Kirillov, A. et al., Panoptic Segmentation, CVPR, 2019.
4. Cheng, B. et al., Panoptic-DeepLab, CVPR, 2020.
5. Cheng, B. et al., Mask2*Former*, CVPR, 2022.