Développez une preuve de concept

Joelle JEAN BAPTISTE - décembre 2025

Introduction

Objectifs du POC

Ce POC vise à comparer un modèle de segmentation classique (DeepLabV3+) avec une architecture récente basée Transformers (Mask2Former) afin d'évaluer leur performance sur un sous-ensemble simplifié du dataset Cityscapes dans un contexte de vision embarquée.

Plan de la présentation

- Démarche
- Modélisation
- Comparaison des résultats
- Développement de l’application

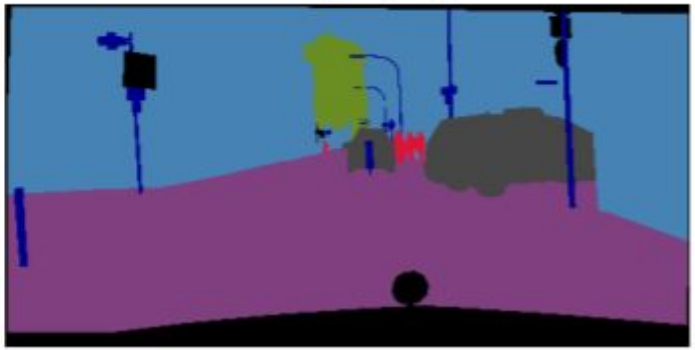
Démarche

Choix du dataset

Le dataset Cityscapes, composé de 2 975 images d'entraînement et 500 images de validation, offre des scènes urbaines annotées pixel par pixel idéales pour évaluer des modèles de segmentation.

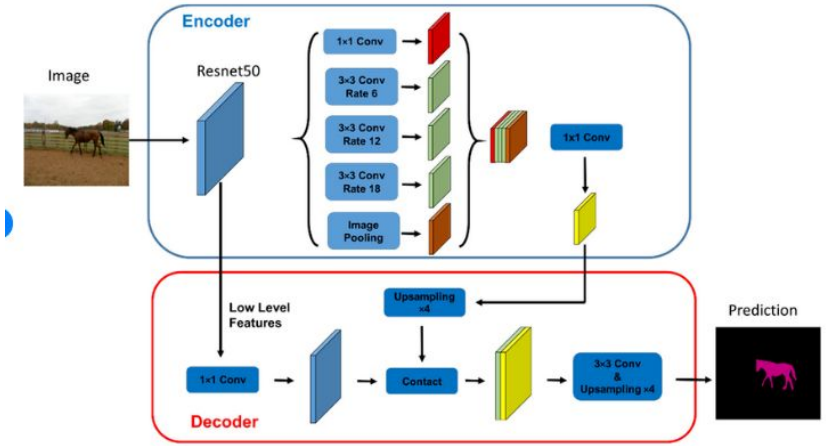
Image

Masque (GT)



Choix du modèle baseline

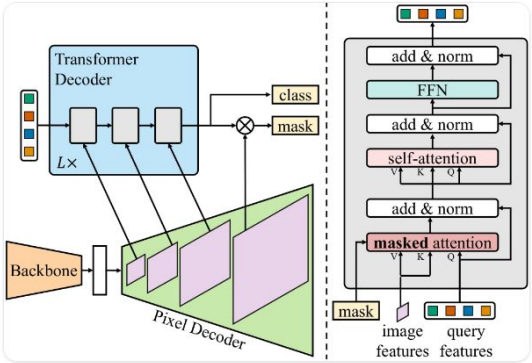
- **DeepLabV3+ (ResNet50)**
- Modèle CNN robuste et éprouvé pour la segmentation
- Sert de **référence stable** pour comparer Mask2Former



The network structure combining Deeplabv3 and ResNet 50.

Choix du modèle récent

- **Mask2Former (Transformer moderne – Meta AI, 2022)**
- Modèle **SOTA** en segmentation (sémantique, instance, panoptique)
- Capacité à **capturer le contexte global** grâce au Transformer
- Architecture **unifiée et plus flexible** que les CNN classiques



Mask2Former architecture. Taken from the original paper.

Sources et documentation

- **Paper Mask2Former (Meta AI, 2022)**
  https://arxiv.org/abs/2112.01527
- **Documentation HuggingFace — modèle Mask2Former**
  https://huggingface.co/docs/transformers/model_doc/mask2former
- **Fiche HuggingFace du paper**
  https://huggingface.co/papers/2112.01527
- **Article Medium — introduction & vulgarisation**

  https://medium.com/@HannaMergui/maskformer-per-pixel-classification-is-not-all-you-need-for-semantic-segmentation-1e2fe3bf31cb

Modélisation

Préparation du dataset

- **Réduction à 8 classes principales**
- **Redimensionnement des images** (format léger pour l'entraînement)
- **Remapping des masques** vers les classes retenues
- **Split final : 2975 train / 500 val**

Un pipeline unifié

Workflow commun : **prétraitement → entraînement → évaluation → déploiement**

Architecture identique pour tester **DeepLabV3+ vs Mask2Former** dans les mêmes conditions

Tracking complet via **MLflow** (métriques + hyperparamètres + artefacts)

Pipeline reproductible, facilement **réexécutable et extensible**

Random Search

- Recherche automatique des **meilleurs hyperparamètres** pour chaque modèle
- 10 configurations testées pour : **learning rate, weight decay, epochs, optimizer**
- Suivi et comparaison des essais via **MLflow**
- Sélection de la **meilleure config** pour l'entraînement final de DeepLabV3+ et Mask2Former

Comparaison

Métriques suivies

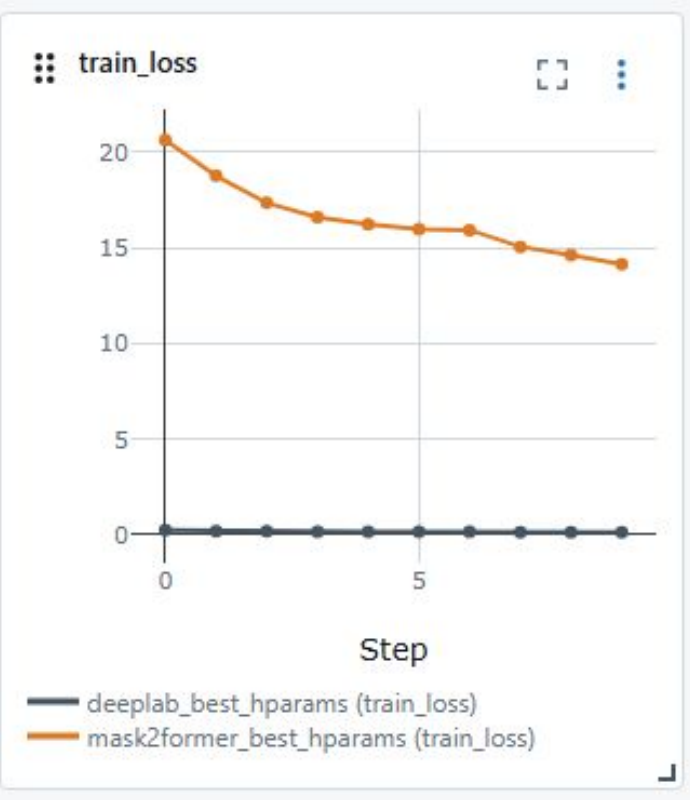
- **train_loss** — erreur sur le jeu d'entraînement
- **val_loss** — erreur sur le jeu de validation
- **pixel_acc** — précision pixel par pixel
- **mIoU** — qualité globale de la segmentation
- **imgs_per_sec** — vitesse d'inférence
- **train_time_sec** — temps moyen d'une epoch

Train Loss

Les deux modèles apprennent correctement avec une perte qui diminue au fil des epochs.

DeepLabV3+ présente une loss très faible (échelle différente → non comparable directement).

Mask2Former suit une décroissance régulière montrant une optimisation stable.

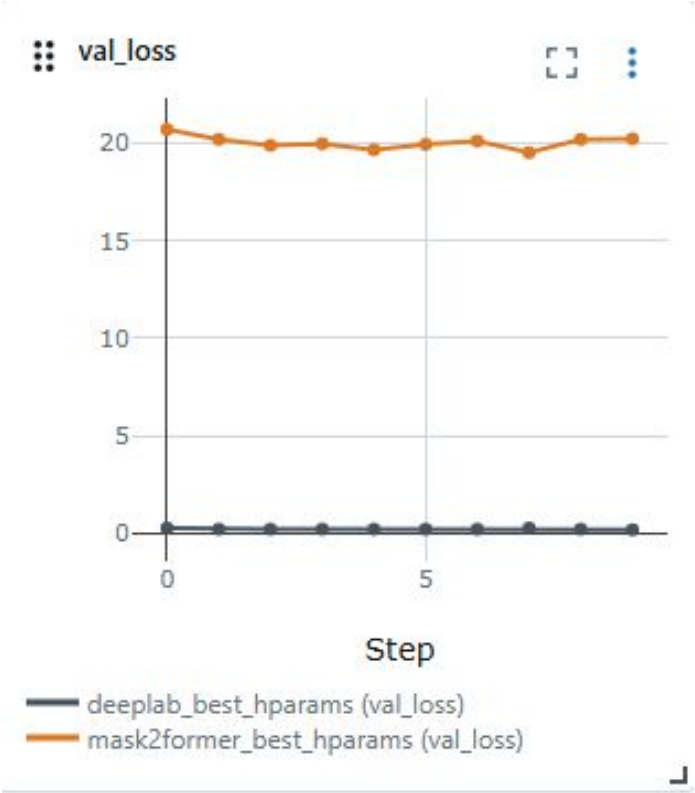


Validation Loss

Les deux modèles utilisent **des fonctions de perte différentes**, donc les valeurs ne sont **pas directement comparables**.

DeepLabV3+ affiche une loss très basse, tandis que Mask2Former évolue sur une échelle plus élevée.

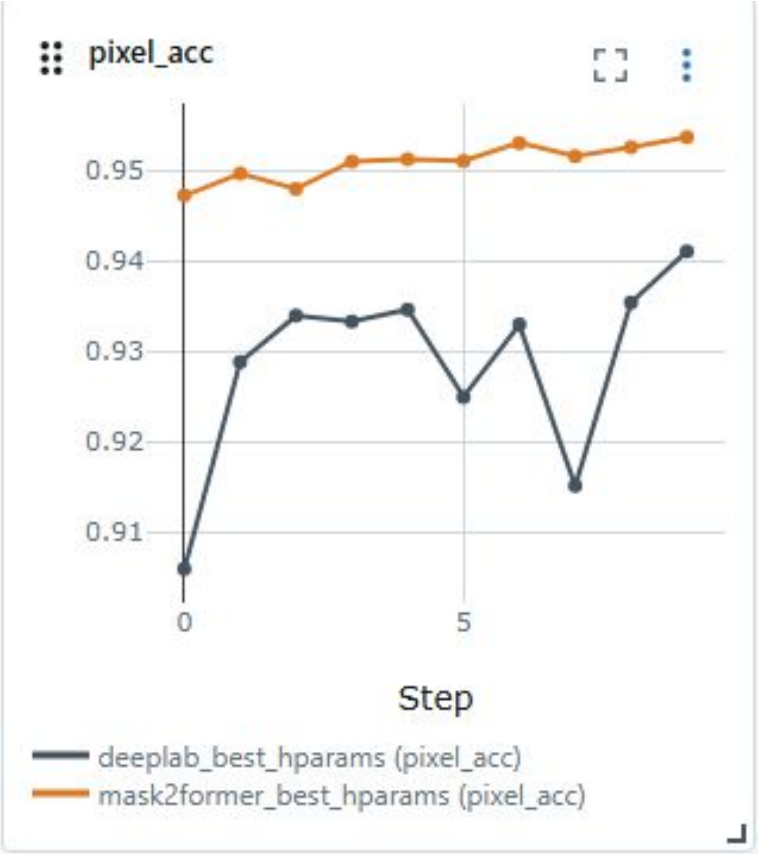
Conclusion : la *validation loss* ne permet pas d'évaluer lequel des deux modèles performe le mieux — il faut s'appuyer sur des métriques comparables comme le **mIoU** ou la **pixel accuracy**.



Pixel Accuracy

- **Mask2Former maintient une précision pixel globale plus élevée que DeepLabV3+ sur l'ensemble des epochs.**
- Cette métrique montre que le modèle Transformer produit des cartes de segmentation plus cohérentes pixel par pixel.

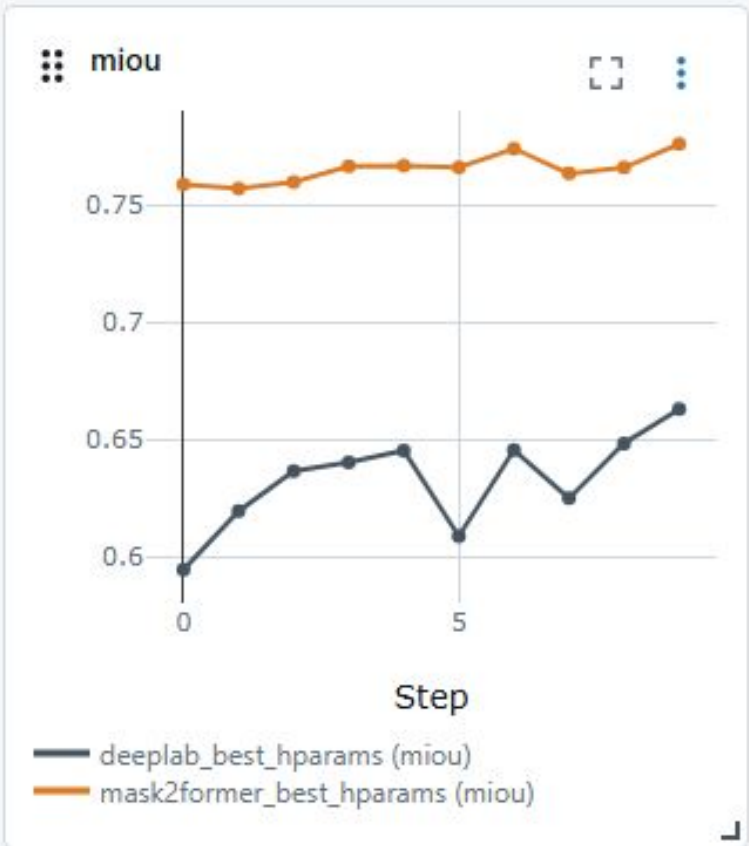
Avantage net pour Mask2Former, qui capture mieux la structure générale de la scène.



mIoU

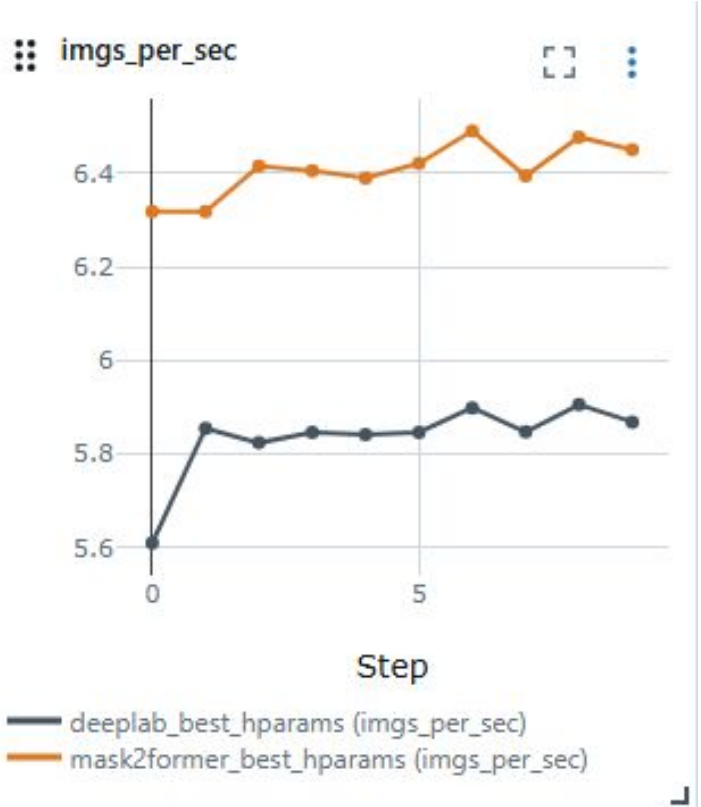
- **Mask2Former** maintient une mIoU **nettement plus élevée** (≈ 0.75–0.78)
- **DeepLabV3+** reste en retrait (≈ 0.60–0.66) malgré une progression régulière
- La séparation entre les deux courbes reste **constante sur l'ensemble des epochs**
- Mask2Former capture mieux le **contexte global** et produit des masques plus cohérents

Mask2Former délivre une qualité de segmentation sensiblement supérieure.



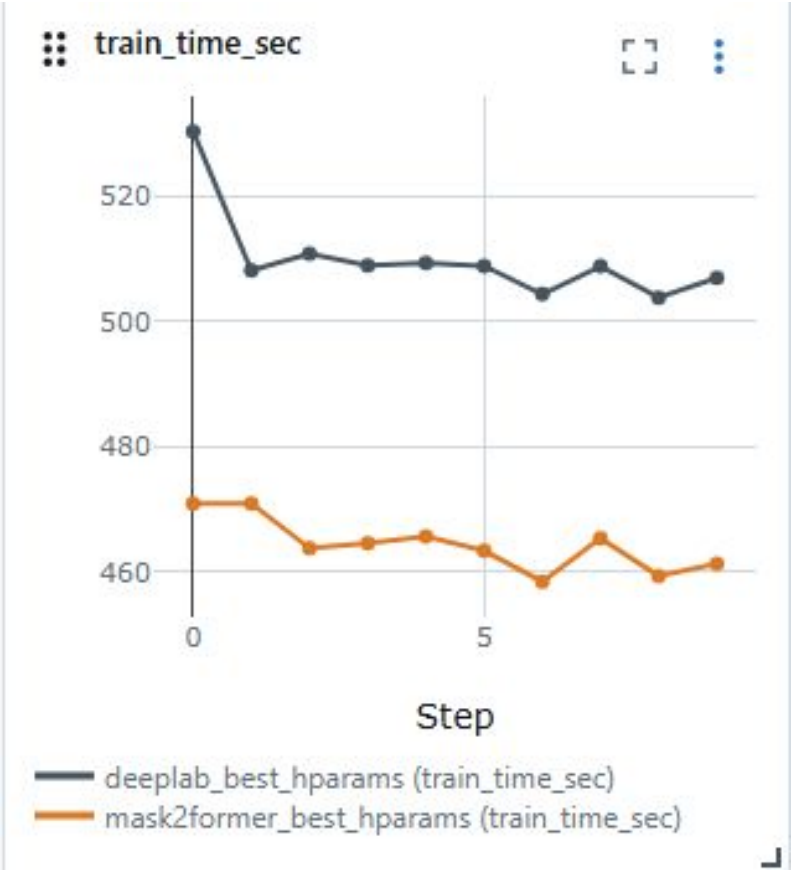
Images/sec

Graphique comparant la vitesse d'inférence des deux modèles : Mask2Former traite environ **6.3 à 6.5 images/seconde**, de manière stable, tandis que DeepLabV3+ reste autour de **5.7 à 5.9 images/seconde**. Mask2Former est systématiquement le plus rapide.



Training time

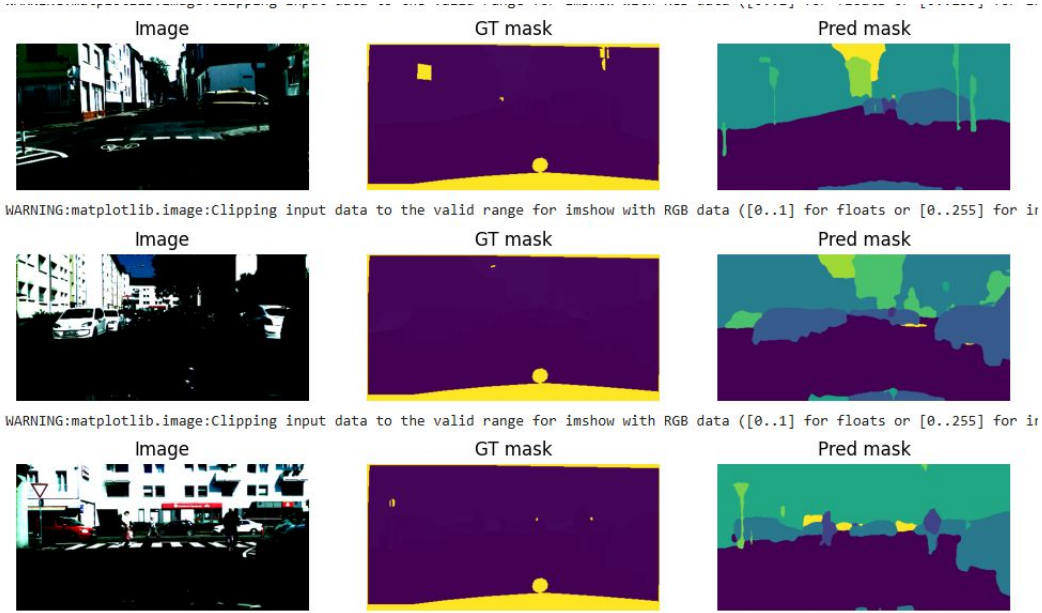
Graphique montrant le temps d'entraînement par epoch : Mask2Former tourne autour de **460 secondes par epoch**, tandis que DeepLabV3+ est plus lent avec environ **500 à 520 secondes par epoch**. Mask2Former s'entraîne donc légèrement plus vite



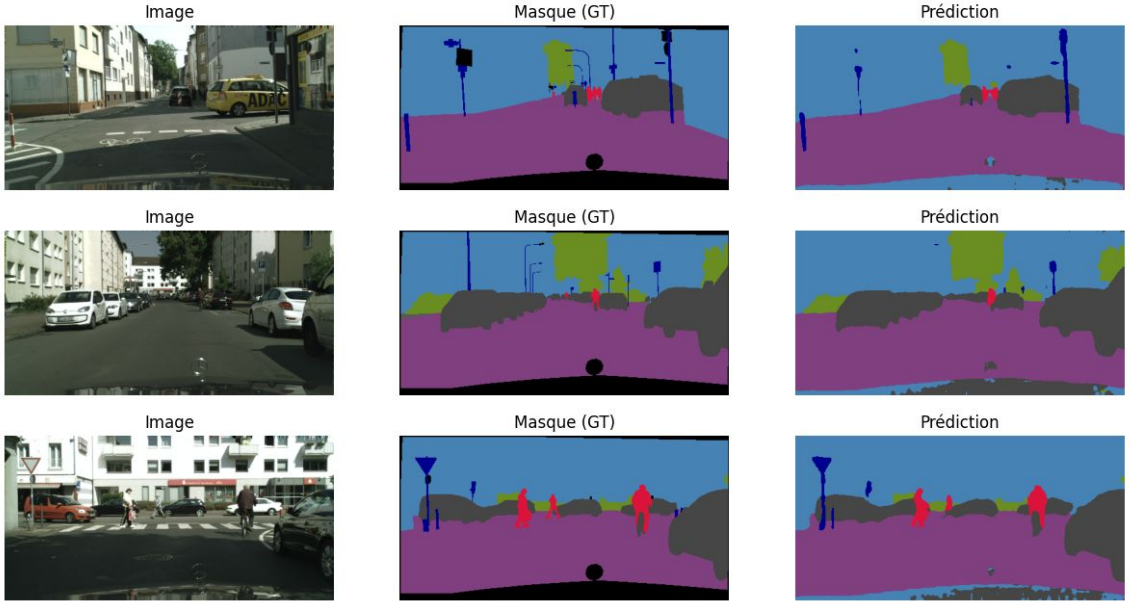
Synthese

- **Mask2Former surpasse DeepLabV3+** : meilleure mIoU (+12–18 pts) et meilleure pixel accuracy.
- **Stabilité et généralisation** supérieures grâce à l'architecture Transformer.
- **Performances pratiques meilleures** : entraînement plus rapide et inférence plus fluide.
- **Conclusion** : Mask2Former est le choix le plus robuste pour une segmentation urbaine fiable et scalable.

Résultats DeeplabV3



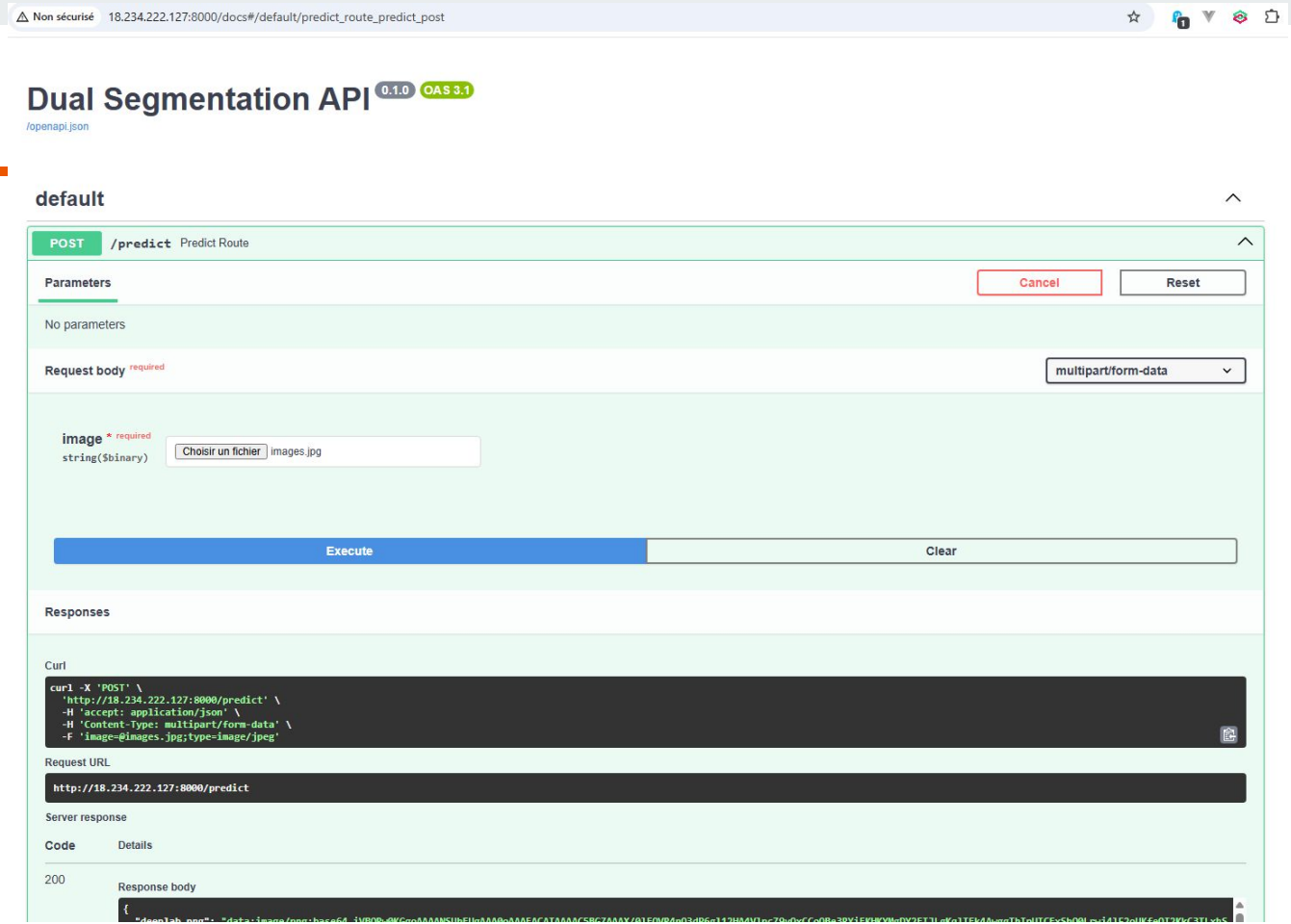
Résultats Mask2Former



Application

Architecture de l'API

- **FastAPI** pour l'exposition d'un endpoint /predict
- **Chargement dynamique** du modèle (DeepLabV3+ ou Mask2Former)
- **Prétraitement serveur** : redimensionnement, normalisation, encodage
- **Inférence** → génération du masque + colorisation
- **Réponse JSON + base64** (image segmentée)
- **Déploiement** : instance cloud (Ubuntu), accès sécurisé via clé SSH


Non sécurisé
18.234.222.127:8000/docs#/default/predict_route_predict_post
Dual Segmentation API
0.1.0
OAS 3.1
/openapi.json
default
POST
/predict
Predict Route
Parameters
Cancel
Reset
No parameters
Request body
required
multipart/form-data
image
* required
string($binary)
Choisir un fichier
images.jpg
Execute
Clear
Responses
Curl
curl -X 'POST' \
'http://18.234.222.127:8000/predict' \
-H 'accept: application/json' \
-H 'Content-Type: multipart/form-data' \
-F 'image=@images.jpg;type=image/jpeg'
Request URL
http://18.234.222.127:8000/predict
Server response
Code
Details
200
Response body

Présentation de l'interface Streamlit

- Interface simple et accessible pour tester les deux modèles
- Téléversement d'une image → envoi automatique vers l'API
- Affichage côte-à-côte des prédictions DeepLabV3+ vs Mask2Former
- Légende des classes intégrée pour faciliter l'interprétation
- Design épuré + respect des normes d'accessibilité (contraste, texte alternatif)

Accessibilité

- Palette adaptée **WCAG AA** (contrastes renforcés)
- **Textes alternatifs systématiques** pour chaque image et graphique
- Structure visuelle simplifiée : titres clairs, contenu hiérarchisé
- Interface utilisable via **clavier uniquement** (navigation tab)

Preuve de Concept : Segmentation d'images avec DeepLabV3+ et Mask2Former

Dataset : Cityscapes (version réduite à 8 classes)

Ce projet utilise une version simplifiée du dataset **Cityscapes**, largement utilisé pour la recherche en **segmentation d'images dans les systèmes de conduite autonome**.

Les images proviennent de scènes urbaines (Allemagne) capturées depuis un véhicule,
et chaque image possède un **masque sémantique** où chaque pixel correspond à une classe.

Nous utilisons ici une version regroupée du dataset, limitée à **8 grandes catégories** :

Les 8 classes retenues

- **flat** — route, trottoir
- **human** — piétons
- **vehicle** — voitures, bus, camions
- **construction** — bâtiments, structures
- **object** — panneaux, barrières, poteaux
- **nature** — végétation, arbres
- **sky** — ciel
- **void** — pixels ignorés ou non pertinents

Répartition du dataset

Deux configurations ont été utilisées pour analyser le comportement des modèles :

- **Train : 300** images | **Val : 50** images
- **Train : 2975** images | **Val : 500** images

La première (300/50) sert à tester l'apprentissage avec peu de données.
La seconde (2975/500) permet d'évaluer les modèles à plus grande échelle.

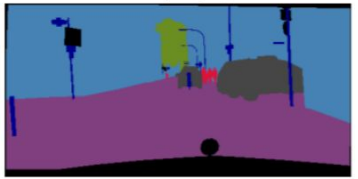
:

Exemple d'image et masque de vérité terrain (GT)

Voici un exemple permettant de visualiser ce à quoi ressemblent les données utilisées pour l'entraînement du modèle.

Image

Masque (GT)



Brève description de l'entraînement

Chaque modèle apprend à prédire, pour **chaque pixel**, la classe correcte parmi 8 catégories.

L'entraînement suit les étapes suivantes :

1. **Prétraitement** : redimensionnement, normalisation, encodage des masques.
2. **Architecture du modèle** :
   - **DeepLabV3+ (ResNet50)** — CNN avec décodeur dilaté, rapide et stable
   - **Mask2Former** — architecture transformer moderne, très performante
3. **Suivi des métriques** :
   - mIoU
   - pixel accuracy
   - pertes
   - vitesse
4. **Validation** :
   - le modèle est testé sur un jeu **jamais vu**
   - comparaison systématique DeepLabV3+ vs Mask2Former sur 10 epochs

L'objectif final du projet est de comparer la performance et l'efficacité des deux modèles.

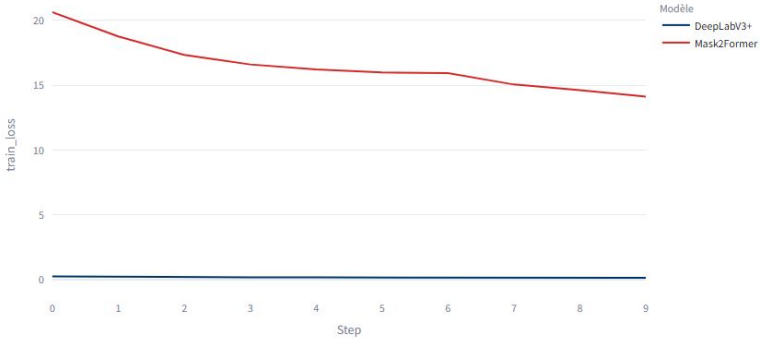
Comparaison des modèles sur les métriques clés

Sélectionnez une métrique à comparer :

train_loss

Training Loss

Comparaison train_loss — DeepLabV3+ vs Mask2Former



Description alternative (accessibilité) :

DeepLabV3+ conserve une perte d'entraînement quasi nulle tout au long des 10 époques, restant stable autour de 0. Mask2Former commence avec une perte élevée autour de 20, puis diminue régulièrement jusqu'à environ 14 à la fin de l'entraînement. La courbe Mask2Former descend progressivement alors que DeepLab reste extrêmement bas et stable. La différence entre les deux modèles est constante et très marquée, indiquant que DeepLab converge beaucoup plus rapidement sur ce jeu d'entraînement.

La *train loss* représente l'erreur du modèle pendant la phase d'entraînement. Elle mesure à quel point les prédictions du modèle s'éloignent des masques de vérité terrain sur le **jeu d'entraînement** uniquement.

- Une train loss basse indique une bonne capacité à apprendre les patterns du dataset.
- Une loss plus élevée ne signifie pas nécessairement un mauvais modèle : cela dépend de la fonction de perte utilisée.
- Dans la segmentation, **les fonctions de perte peuvent être très différentes entre les architectures**, rendant les valeurs non comparables directement.

Analyse du graphe — DeepLabV3+ vs Mask2Former

Sur ce graphique :

- **DeepLabV3+** apparaît en bleu et affiche une train loss extrêmement faible (proche de 0).
- **Mask2Former**, en rouge, démarre autour de ~20 et descend lentement vers ~15.

Cette différence ne signifie pas que DeepLabV3+ est meilleur :

- Les deux modèles utilisent des **fonctions de perte différentes**.
- La loss de Mask2Former intègre une perte panoptique complexe (avec plusieurs composantes), ce qui génère des valeurs plus élevées.

Conclusion

- La **forme de la courbe** est plus informative que les valeurs absolues.
- **Mask2Former** montre une décroissance progressive et stable → apprentissage cohérent.
- **DeepLabV3+** atteint très vite une loss basse → fonction de perte plus simple.

Les train_loss ne doivent pas être comparées en valeur absolue entre ces deux modèles.
C'est le comportement de la courbe qui compte.

Tester les modèles sur une nouvelle image

Légende des classes

Flat
Surfaces planes (routes, trottoirs)

Human
Personnes (piétons, silhouettes)

Vehicle
Véhicules (voitures, bus, motos…)

Construction
Éléments de construction (bâtiments, murs)

Object
Objets urbains (poteaux, panneaux…)

Nature
Nature (arbres, herbes, végétation)

Sky
Ciel

Void
Régions non pertinentes / inconnues

Sélectionnez une image (JPEG/PNG)

Drag and drop file here
Limit 200MB per file • JPG, PNG, JPEG

Browse files

images.jpg 6.5KB

Envoi de l'image à l'API…

DeepLabV3+ – Résultat

Prédiction DeepLab – Masque segmenté

Mask2Former – Résultat

Prédiction Mask2Former – Masque segmenté

Conclusion

- Le POC démontre que **Mask2Former** surpasse nettement **DeepLabV3+** sur les métriques clés (*mIoU*, *pixel accuracy*, vitesse).*
- Le pipeline complet (prétraitement → entraînement → API → interface Streamlit) est **fonctionnel**, **reproductible**, et déjà **déployé dans le cloud**.
- La solution propose une segmentation efficace pour des scènes urbaines, ouvrant la voie à une intégration dans un système embarqué.

Ouverture

Ce travail pose les bases d'une extension vers la **segmentation panoptique** — un axe essentiel pour les futures applications embarquées dans les véhicules intelligents, permettant une compréhension plus fine et unifiée de la scène.