OPENCLASSROOMS

Projet 9

Preuve de Concept

Comparaison de modèles pour la segmentation d'images embarquées

Future Vision Transport - Équipe R&D - IA & Vision par Ordinateur

Anthony VIDRAGO – Avril 2025

I - Dataset Retenu

Dans le cadre de ce projet de segmentation d'images urbaines, j'ai utilisé le jeu de données Cityscapes, une référence dans le domaine de la vision par ordinateur.

Ce dataset propose des images haute résolution (2048×1024) capturées à partir de caméras embarquées sur des véhicules dans différentes villes européennes.

L'objectif est de réaliser une segmentation sémantique précise des scènes urbaines.

Le dataset contient :

- Environ 5 000 images annotées finement avec des masques pixel par pixel
- Une hiérarchie de 34 classes d'objets (voiture, routes, piétons, bâtiments…)
- Un découpage en trois splits : train (2 975 images), val (500 images) et test (1 525 images).

Pour des raisons de performance et de reproductibilité, j'ai restreint le jeu de données à :

- 2 000 images pour l'entrainement
- 500 pour la validation
- Redimensionnées à 256*256 pixels
- Et regroupées en 8 classes principales via un remapping des catégories initiales.

Les masques de segmentation ont été retraités afin de ne conserver que les catégories utiles à l'analyse du trafic et de la structure urbaine : chaussées, bâtiments, objets urbains, nature, ciel, humains, véhicule et les éléments non pertinents (void).

Ce prétraitement garantie la cohérence entre les modèles testés et permet de comparer efficacement les performance d'algorithmes récents avec une baseline maitrisée.

II – Les concepts de l'algorithme récent

Le modèle **SegFormer** est un algorithme de segmentation sémantique d'image introduit par NVIDIA en 2021.

Contrairement aux approches classiques basées sur des réseaux convolutionnels (CNN), SegFormer repose sur une architecture « pure Transformer » qui permet de mieux capturer les dépendance globales dans une images.

Il fait partie d'une nouvelle génération de modèles performants, combinant précision, rapidité et légèreté, ce qui le rend particulièrement adapté au cas d'usage en vision par ordinateur sur des plateformes variées (cloud, edge, embarqué).

Une architecture efficace et modulaire

SegFormer se distingue par sa structure composées de deux blocs principaux :

- Encodeur : MiT (Mix Transformer)
  - Il remplace des traditionnels backbones CNN par un transformer hiérarchique nommé « MiT » qui génère des représentations multi-échelles tout en conservant la structure spatiale des objets
  - Contrairement aux Vision Transformers classiques, MiT utilise des patches non chevauchants, du self-attention local pour des raisons d'efficacité et des strides dynamiques pour réduire progressivement la résolution des cartes de caractéristiques.
  - Ce design permet d'équilibrer le cout computationnel et la richesse des représentations spatiales.

- Decoder léger sans upsampling couteux
  - Contrairement à des modèles comme U-Net ou DeepLabV3+ qui utilise des décodeurs lourd ou à plusieurs étapes, SegFormer adopte un decoder simple et efficace : une simple concaténation des sortie multi-échelles suivie de couche linéaire.
  - Ce choix de simplicité est motivé par le fait que l'essentiel de la complexité est absorbé par le backbone Transformer rendant l'assemblage final plus rapide à exécuter et plus facile à entrainer.

Avantages clés

- **Performance** : SegFormer surpasse de nombreux modèles sur des benchmarks comme Cityscapes ou ADE20K, avec des scores mIoU et Dice élevées.
- **Polyvalence** : Disponible en plusieurs tailles (B0 à B5), il s'adapte aussi bien aux contraintes d'embarqué (B0) qu'aux environnement haute capacité (B5).
- **Sans dépendance à un tokenizer** : Contrairement à d'autre Transformer visuels, SegFormer ne requiert pas d'encodeur de position complexe ce qui simplifie son déploiement.
- **Optimisé pour l'efficacité mémoire** : Grace à une réduction progressive de la résolution, l'architecture est plus économe en mémoire GPU par rapport à d'autre modèles Transformer

SegFormer dans le cadre du Projet 9

Dans le Projet 9, j'utilise SegFormer-B5 comme modèle « récent » en le comparant à un modèle U-Net avec EfficientNetB3.

Le modèle est adapté pour segmenté 8 classes principales du dataset Cityscapes, avec un entrainement complet sur un sous ensemble réduit du dataset.

Le poids du modèle a été sauvegardé en « .pth » après fine-tuning et une API FastAPI a été déployée sur Hugging Face Space pour exposer ses prédictions.

III – La modélisation

Dans le cadre du Projet 9, l'objectif est de comparer un modèle récent (SegFormer-B5) à une architecture plus classique, le U-Net avec backbone EfficientNetB3 utilisé comme modèle baseline.

Les deux modèles sont entrainés sur le dataset Cityscapes, remappé en 8 classes principales et dans des conditions rigoureusement identiques pour garantir une comparaison équitable.

Méthodologie de modélisation :

Dataset et prétraitement

- Source: Cityscapes (https://www.cityscapes-dataset.com/)
- Nombre d'images utilisées:
  - Train: 2000 images
  - Validation: 500 images
- Taille des images : redimensionnées à 256x256 pixels
- Classes : Remappage des 34 classes original en 8 grandes catégories cohérentes pour un usage urbain (road, vehicle, humain, etc).
- Masque : format indexé 0-7, colorisés à l'affichage.

Modèle testés

- Baseline : U-Net avec backbone EfficientNetB3 (Tensorflow, Keras)
- Modèle récent : SegFormer B5 (Pytorch, Transformers)

Pipeline de modélisation

- Prétraitement avec Albumentations (normalisation, redimensionnement)
- Entrainement avec suivi des performances (métriques, logs, CSV, courbes)
- Comparaison avec interface Streamlit et API FastAPI

Métriques d'évaluation retenues

Pour évaluer la qualité des prédictions, j'ai utilisé des métriques pertinentes pour la segmentation sémantique multi-classe :

METRIQUE	DESCRIPTION
Accuracy	Pourcentage de pixels correctement classsés
Dice Coefficient	Mesure de similarité entre les masque (2 * intersection / union des pixels)
Mean IoU	Intersection over Union moyen par classe
Loss	CrossEntropyLoss pour le modèle Pytorch ; Dice + CE pondérée pour le Keras
Temps total	Temps d'entrainement complet en secondes (minoré et loggé dans chaque run)

Ces métriques sont enregistrées par epoch pour le train et la validation.

Un logger CSV permet de tracer leur évolution sur des graphiques pour chaque modèle.

Stratégie d'optimisation

Pour le modèle SegFormer B5 (Pytorch)

- Optimiseur : Adam, avec un taux d'apprentissage initiale de 1^{e}-4
- Scheduler : ReduceLROnPlateau pour réduire le learning rate si la loss stagne
- Mixed Précision Training : Utilisation de « torch.cuda.amp » pour accélérer l'entrainement sur GPU tout en réduisant l'utilisation mémoire.
- Upsampling dynamique : les logits sont interpolés pour correspondre à la taille des masque avant calcul de la loss

Pour le modèle EfficientNetB3 (Tensorflow)

- Fonction de perte personnalisées : combinaison de Categorical Crossentropy pondérée + Dice loss
- Callback : EarlyStopping, ReduceLROnPlateau, ModelCheckpoint, CSVLogger
- Augmentation des données : flips, rotation, scale via Albumentations
- One-Hot Encoding pour la supervision multi-classe du masque

Condition expérimentales

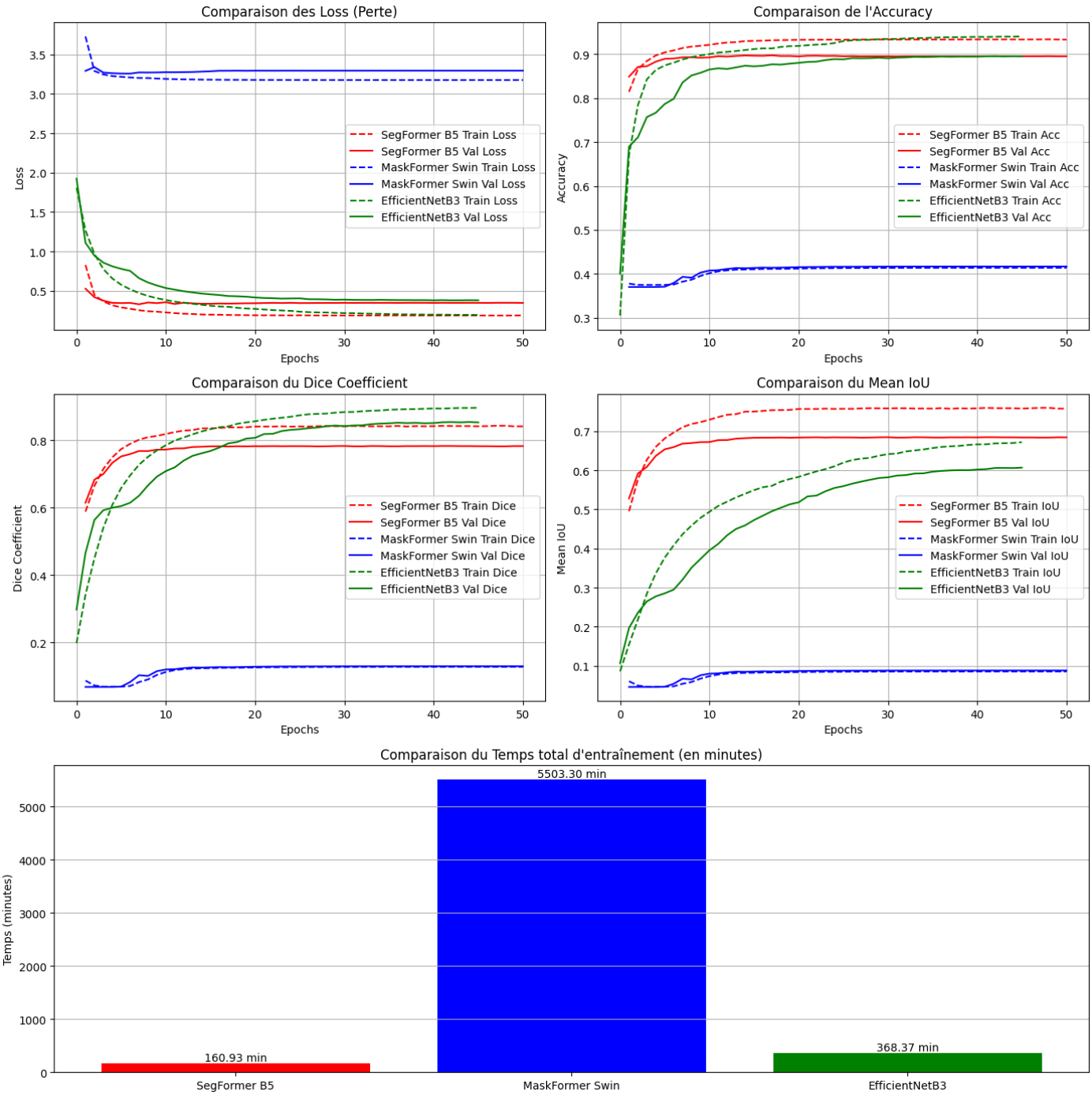
- Batch size : 8, 16, 32, 64 et 128 ont été testé
- Epoch : 50 (early stopping possible)
- Hardware : entrainement sur GPU NVIDIA RTX 3080Ti
- Format de sauvegarde :
    - EfficientNetB3 : model_efficientnet.keras
    - Segformer_B5.pth

IV – Synthèse des résultats

Cette section compare les performances des modèles testés dans le cadre de ce projet.

Le modèle SegFormer B5 retenu comme approche récente, face au modèle U-Net EfficientNetB3, utilisé comme baseline.

Un troisième modèle « MaskFormer Swin » a été testé mais rapidement écarté en raison de performances largement inférieures.



MODELE	ACCURACY	DICE	mIoU	Loss	TEMPS
Segformer B5	~0.90	~0.83	~0.70	~0.20	160.93min
EfficientNetB3	~0.91	~0.86	~0.67	~0.45	368.37min
MaskFormer Swin	~0.42	~0.18	~0.13	~3.7	5503.30min

Analyse des courbes

- Loss (Perte) : SegFormer B5 atteint une perte de validation nettement inférieur (~0.20) et beaucoup plus stable qu'EfficientNetB3. Cela indique une meilleur généralisation.
- Accuracy : Les deux modèles principaux ont une précision proche, autour de 0.90-0.91 ce qui confirme leur fiabilité globale.
- Dice Coefficient : Le modèle EfficientNetB3 conserve un très léger avantage final (~0.86) mais SegFormer atteint très rapidement un score stable de 0.83 ce qui est excellent pour un temps d'entrainement 2 fois plus court.
- Mean IoU : SegFormer B5 se distingue avec un IoU moyen plus élevé (~0.70) prouvant sa meilleure capacité a identifier correctement toutes les classes y compris les plus complexes

Temps d'entrainement

L'entrainement de SegFormer a durée 160.93 minutes, soit près de 2.3 fois plus rapide qu'EfficientNetB3 tout en maintenant un excellent niveau de performance.

Conclusion

- MaskFormer Swin est définitivement écarté, il offre des performance très faible malgré un cout énorme (près de 4 jours)
- EfficientNetB3 reste robuste, mais demande un temps d'entrainement plus élevé pour un gain marginal
- SegFormer B5 devient le modèle le plus équilibré, combinant :
    - Performance de segmentation solide (Dice & IoU élevé)
    - Courbes stables dès les premières époques
    - Temps d'entrainement optimisé
    - Meilleur capacité à généraliser

Ce modèle constitue **une preuve de concept convaincante** pour des système de segmentation robuste, performant et déployable.

V – Analyse de la feature importance globale et locale du nouveau modèle

Dans un problème de segmentation sémantique, la notion classique de feature importance ne repose pas sur des variables tabulaires, mais sur des caractéristique extraites des images.

Ces caractéristiques sont apprises automatiquement par le modèles à travers des couches convolutives et de type Transformer.

Je présente ici les approches retenues pour évaluer l'importance des éléments visuels, tant globalement (quelles zone influencent le plus la prédiction) que localement (comment le modèle segmente une image particulière).

Feature importance globale : attention spatiale et classes dominantes

Le modèle SegFormer intègre des blocs MiT (Mix Transformer) qui génèrent des cartes d'attention multi-échelles.

Ces mécanismes permettent d'identifié à différentes résolution, les régions de l'image qui influence le plus la prédiction finale.

Analyse globale effectuée :

- Nous avons analysé les masques prédits sur un échantillons de validation, puis compté les pixels prédits par classe pour déterminer les classes dominantes.
- La classe « flat » est la plus représenté suivi par la classe « construction », les classes « human » et « object » sont quand à elle les moins représenté.
- Grace aux mécanisme d'attention, le modèle SegFormer apprend à accorder plus de poids aux bords, objets et intersection de classes, ce qui se traduit par de meilleurs performances sur des classes complexes comme les piétons ou les poteaux.

Analyse locale : précision sur des cas concrets

J'ai évalué les prédiction sur plusieurs images avec des masque réels.

Le modèle SegFormer B5 montre :

- Une précision remarquable sur les classes fréquentes comme les routes (flat), bâtiments (construction) et végétation (nature).
- Une capacité à bien détecter les véhicules même lorsqu'ils sont partiellement masqués ou sur des route encombrées.
- Des résultats plus mitigés sur les piétons (humain) et objets isolés, en raison de leurs faible présence dans le dataset.

Pour quantifier cette analyse, j'ai mesuré les scores Dice et IoU image par image et observé que la prédiction est généralement cohérente sur les contours même en cas de bruit ou d'obstacles.

Interprétabilité du modèle

Contrairement aux modèle tabulaires, les transformer visuels comme SegFormer ne permettent pas d'extraire une « feature importance » explicite pour chaque pixels ou patch. Toutefois :

- L'analyse des carte d'attention interne (non activés par défaut dans transformers) pourrait révéler les régions les plus influentes
- Des techniques comme Grad-CAM ou Score-CAM sont difficiles à appliquer directement, mais pourrait être envisagées dans une démarche future avec un wrapper CNN+SegFormer.

Conclusion

L'importance des classes dans le dataset guide fortement l'apprentissage du modèle.

SegFormer exploite efficacement cette structure grâce a son architecture Transformer :

- Les zones fréquentes sont bien segmentées (route, bâtiments, véhicules).
- Les zones complexes ou peu fréquentes (piétons, poteaux, objets) restent perfectibles.
- Son attention hiérarchique permet une interprétation indirecte via les résultats prédits visible à travers la cohérence des contours et la stabilité des performances sur différentes classes.

VI – Limites et améliorations possible

Limite identifiées

Malgré les performances convaincantes du modèle SegFormer B5, plusieurs limites techniques et méthodologique subsistent dans notre approche :

- Déséquilibre des classes
  Le dataset Cityscape, même réduit à 2500 images (train + val), présentes une forte hétérogénéité dans la distribution des pixels : certaines classes comme « flat », « construction » ou « nature » sont largement majoritaires, tandis que les classes « objets » et « human » sont sous représentés.
  Cela impacte négativement la généralisation du modèle sur les classes rare malgré une architecture robuste.
- Résolution réduite des images (256x256)
  Le redimensionnement à 256x256 pour accélérer l'entrainement nuit à la précision des contours et à la finesse de segmentation en particulier pour les objets petits ou éloignés.
- Entrainement couteux pour certains modèles
  Le modèle MaskFormer Swin par exemple a été écarté après constatation d'un temps d'entrainement excessif (5503.30min) pour un gain de performance marginal.

Amélioration envisagées

- Equilibrage des classes
  Utilisation de techniques de data augmentation ciblé pour les classes rares (oversampling, mixup copy-paste, etc…)
- Utilisation d'une résolution intermédiaire (512x512)
  Compromis entre précision spatiale et temps d'entrainement acceptable sur GPU
  Cela permettrait d'améliorer la détection d'objets fins et bords complexes.
- Visualisation des attentions internes
  Exploitation des couches d'attention de SegFormer pour générer des attention maps permettant une meilleure compréhension des décision du modèles.
- Affinage de la stratégie d'optimisation
  Tester avec d'autre planification de taux d'apprentissage (oneCycle, Cosine Annealing) et d'autre optimiseurs comme AdamW pour un entrainement plus stable.