Note Technique sur la Segmentation Sémantique

1. Introduction

Cette note technique présente une vue d'ensemble des approches utilisées, des résultats obtenus et des améliorations potentielles pour la segmentation sémantique.

Le but de ce projet était de tester deux approches de segmentation sémantique afin de les intégrer dans des véhicules autonomes. La segmentation sémantique est une technique en vision par ordinateur qui consiste à attribuer une classe (comme "route", "bâtiment", "véhicule") à chaque pixel d'une image. Cela permet une compréhension détaillée de l'image et est crucial pour des applications comme la conduite autonome afin que les véhicules comprennent et analysent leur environnement en temps réel pour une navigation sûre et efficace.

Contrairement à une simple classification d'image où l'on attribue une étiquette unique à l'image entière (par exemple, "chien" ou "chat"), la segmentation sémantique implique de classer chaque pixel individuellement. Cela signifie qu'une image doit être décomposée pixel par pixel pour attribuer à chaque partie de l'image une étiquette spécifique. Ce niveau de détail ajoute une couche de complexité considérable car le modèle doit non seulement reconnaître des objets, mais aussi délimiter précisément leurs contours et distinguer les différentes classes au sein d'une même image.

Image originale

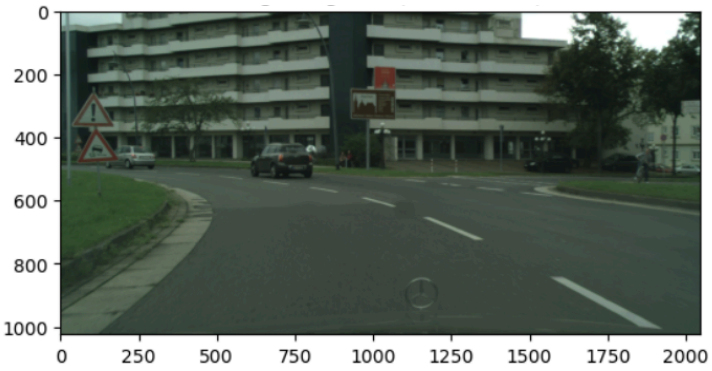
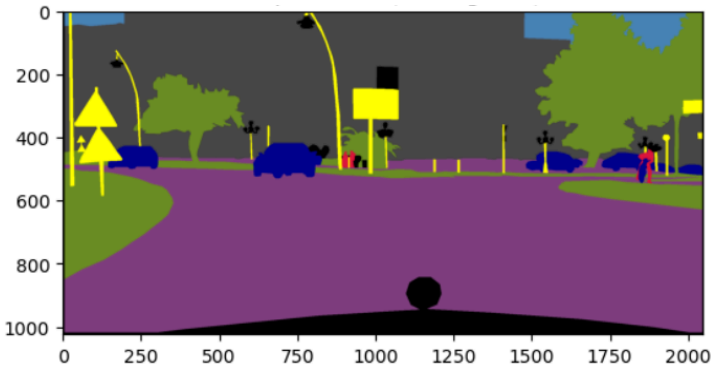


Image segmentée



Exemple de segmentation sémantique d’un paysage urbain

2. Approches et Synthèse de l'État de l'Art

Les approches pour la segmentation sémantique ont évolué avec le temps. On est passé de méthodes basées sur des filtres simples à des modèles complexes de réseaux de neurones.

Méthodes Classiques

Les méthodes classiques de segmentation d'images reposaient principalement sur des techniques de traitement d'image traditionnelles. Ces méthodes incluent :

- Filtres de Sobel : Utilisés pour détecter les bords dans les images en calculant la dérivée de l'image par rapport aux axes x et y.
- Segmentation par Seuillage : Technique simple où les pixels d'une image sont classés en fonction de leur intensité. Par exemple, on peut définir un seuil pour distinguer l'avant-plan de l'arrière-plan.
- Régions Croissantes : Méthode qui commence à partir de "graines" initiales et ajoute des pixels voisins qui ont des propriétés similaires (comme la couleur ou l'intensité) pour former des régions segmentées.
- K-means Clustering : Algorithme de regroupement qui partitionne l'image en K groupes en minimisant la variance au sein de chaque groupe.

Ces méthodes étaient limitées par leur capacité à capturer des informations complexes et leur manque de généralisation à des scénarios variés.

Méthodes Modernes:

Avec l'avènement de l'apprentissage profond, les méthodes modernes de segmentation d'images ont radicalement amélioré la précision et la robustesse des modèles. Les principales innovations incluent :

- Réseaux de Neurones Convolutifs (CNN) : Ces réseaux utilisent des couches de convolution pour extraire des caractéristiques locales de l'image. Ils sont capables de détecter des motifs complexes et des structures variées dans les images.
- Fully Convolutional Networks (FCN) : Un type de CNN spécialement conçu pour la segmentation d'images. Contrairement aux CNN traditionnels qui produisent des

étiquettes pour une image entière, les FCN génèrent une carte de segmentation où chaque pixel est classifié.

- UNet : Ce modèle combine un chemin de contraction et un chemin d'expansion avec des connexions directes entre eux pour segmenter les images de manière précise. Il est particulièrement efficace pour les tâches nécessitant une segmentation pixel par pixel.
- SegNet : Un autre type de réseau de neurones convolutionnel conçu pour la segmentation sémantique. Il utilise un processus de codage et de décodage similaire à celui de l'UNet.
- DeepLab : Une architecture qui utilise des convolutions à trous (atrous convolutions) et des CRF (Conditional Random Fields) pour améliorer la précision des contours segmentés.

Ces méthodes modernes tirent parti des grandes quantités de données et de la puissance de calcul pour apprendre des représentations riches et complexes, permettant des segmentations beaucoup plus précises et robustes que les méthodes classiques. Pour ce projet nous nous sommes intéressés à UNet.

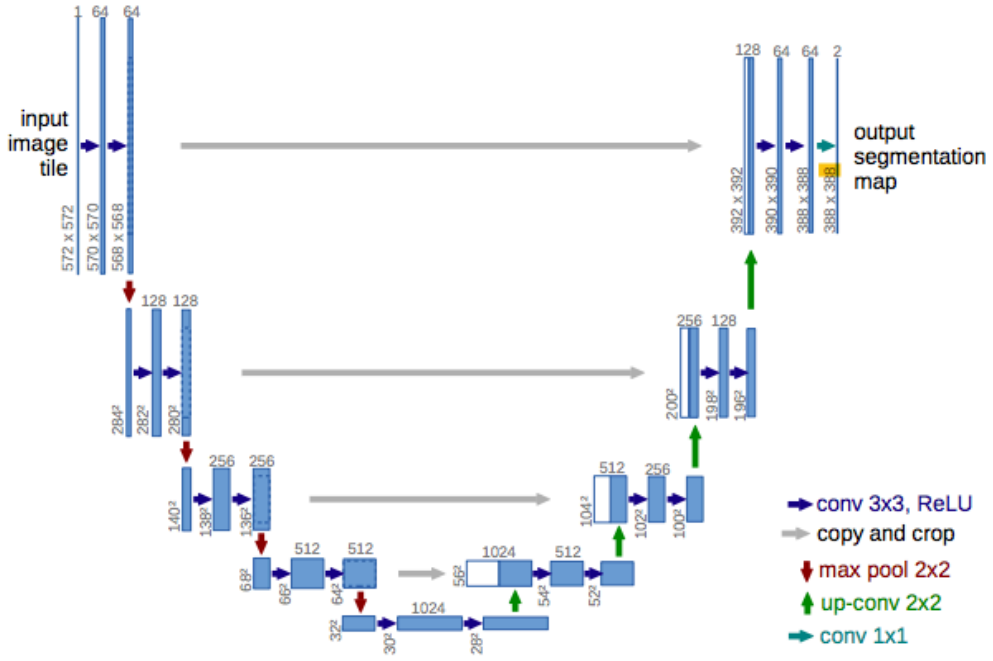
3. Modèle et Architecture Retenue

Modèle UNet

Le modèle UNet est un type de réseau de neurones spécialement conçu pour la segmentation sémantique d'images. Son architecture est en forme de "U" ce qui lui permet de capturer à la fois des détails fins et le contexte global de l'image. Il est composé de 3 blocs :

- l' Encoder : Cette partie du réseau réduit progressivement la taille de l'image tout en capturant ses caractéristiques importantes. Cela se fait en utilisant des opérations de convolution et de max-pooling qui permettent de résumer les informations de l'image.
- le Pont : Il relie l'encodeur et le réseau de décodeur en transmettant les détails fin et le contexte de l'image qui ont été capturés lors de l'encodage.

- le Decoder : Cette partie agrandit l'image de manière à la restaurer à sa taille originale. Elle combine les informations résumées avec les détails fins capturés précédemment pour produire une segmentation précise. Les opérations utilisées ici incluent des convolutions transposées, qui sont l'inverse des convolutions.



Architecture en "U" d'UNet - source : blent.ai/blog/a/unet-computer-vision

Modèle Mini-UNet

Le Mini-UNet est une version simplifiée du modèle UNet. Il conserve la structure de base en "U" mais avec moins de couches et de paramètres. Cela le rend plus rapide à entraîner et à exécuter, ce qui est un avantage dans des environnements avec des ressources limitées.

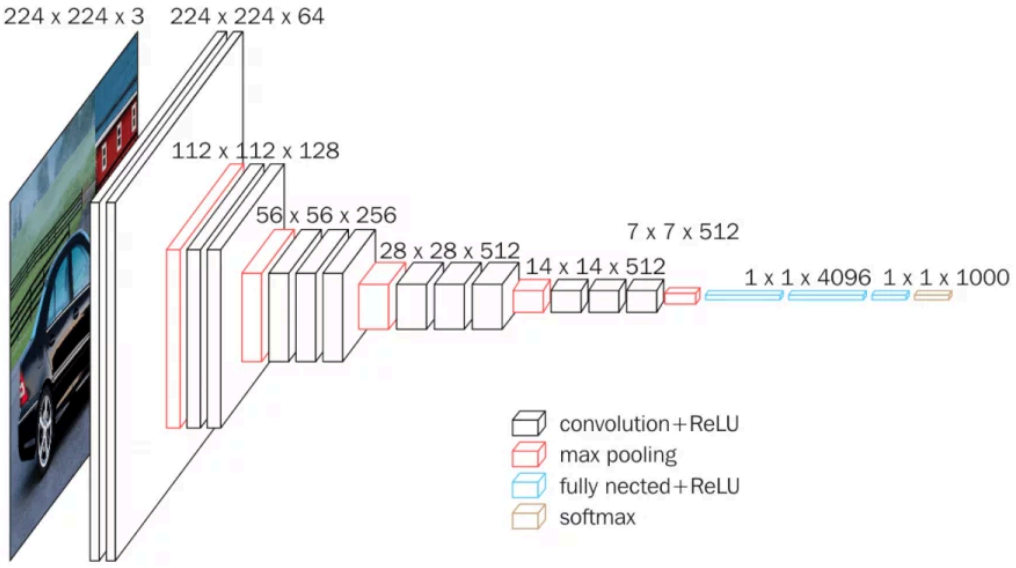
- Avantages:
    - Rapide à Entraîner: Moins de paramètres signifie moins de temps pour l'entraînement.
    - Exécution Rapide: Idéal pour les environnements où la puissance de calcul est limitée.

- Inconvénients:
    - Moins Précis: Moins de couches et de paramètres signifient une perte de précision par rapport à un modèle UNet complet.

Modèle VGG16-UNet

Le modèle VGG16-UNet combine les avantages de l'UNet avec ceux d'un autre réseau de neurones connu sous le nom de VGG16.

- VGG16: C'est un réseau de neurones pré-entraîné sur un grand nombre d'images. Il est très bon pour extraire des caractéristiques complexes des images grâce à ses nombreuses couches de convolution. En utilisant VGG16 comme base (ou encodeur), on peut profiter de ces caractéristiques pré-apprises, ce qui améliore la précision du modèle pour la segmentation.



Architecture VGG16 - source : neurohive.io/en/popular-networks/vgg16/

- UNet: La partie UNet du modèle est utilisée pour la partie d'expansion (decoder), qui transforme les caractéristiques extraites par VGG16 en une image segmentée.

L'utilisation de VGG16 permet au modèle de mieux comprendre les détails complexes et les structures des images, rendant ainsi la segmentation plus précise.

4. Métriques d'Évaluation de la Segmentation Sémantique

Pour évaluer la segmentation sémantique, deux métriques sont usuellement utilisées:

Dice Coefficient

Le Dice Coefficient est une métrique couramment utilisée pour évaluer les performances de segmentation. Il mesure la similarité entre les pixels prédits et les pixels réels. Imaginons deux cercles représentant deux ensembles de pixels : le cercle A (pixels prédits comme appartenant à une classe) et le cercle B (pixels réellement appartenant à cette classe). Le Dice Coefficient calcule combien ces deux cercles se chevauchent. Plus le chevauchement est important, plus le score est élevé. S'ils se chevauchent parfaitement, le score est de 1 tandis qu'un score de 0 indique aucun chevauchement.

Le Dice Coefficient est utile pour les tâches où les objets à segmenter sont de taille similaire et il favorise les régions avec une grande intersection. Par exemple, dans une image où l'on veut segmenter les cellules dans un échantillon biologique, cette métrique est pertinente car les cellules ont généralement des tailles comparables.

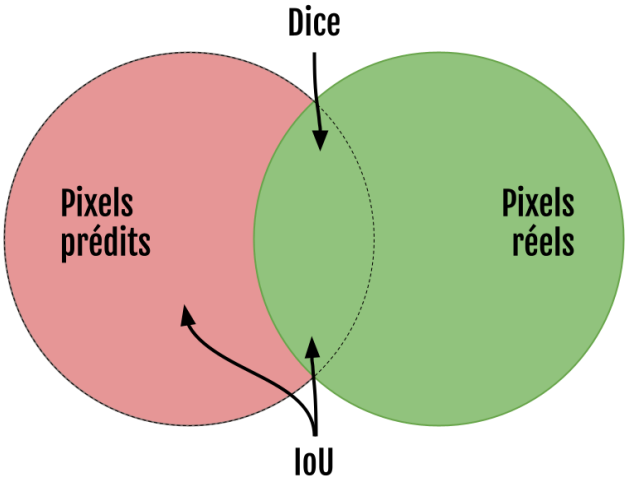
Intersection over Union (IoU)

L'Intersection over Union (IoU), aussi connue sous le nom de Jaccard Index, est une autre métrique populaire pour évaluer la segmentation. Reprenons l'exemple des deux cercles A et B. L'IoU mesure le ratio de l'intersection par rapport à l'union totale des pixels prédits et réels. Cela signifie qu'il pénalise davantage les erreurs (les zones des cercles qui ne se chevauchent pas). Comme le Dice Coefficient, un score de 1 indique une correspondance parfaite et un score de 0 indique aucune correspondance.

Cette métrique est souvent utilisé en segmentation urbaine pour les raisons suivantes:

- Variabilité des Objets: Dans les environnements urbains, les objets varient beaucoup en taille et en forme. L'IoU est moins sensible aux tailles des objets, ce qui en fait une mesure plus équilibrée.
- Gestion des Fausses Prédictions: L'IoU pénalise davantage les fausses prédictions (pixels prédits à tort comme appartenant à une classe), ce qui est crucial pour les applications de sécurité comme les véhicules autonomes.

- Robustesse: L'IoU offre une évaluation plus robuste lorsque les objets à segmenter sont de tailles très différentes, ce qui est fréquent dans les scènes urbaines avec des bâtiments, des routes, des piétons, et des véhicules.



5. Synthèse des Résultats

Modèle	Learning rate	Training time (min)	IoU train	IoU test
Mini_Unet	1e-4	44	0.81	0.78
	1e-3	45	0.83	0.81
	1e-2	45	0.82	0.80
Mini_Unet + data augmentation	1e-4	48	0.80	0.76
	1e-3	47	0.83	0.82
	1e-2	47	0.82	0.81
VGG16	1e-4	51	0.93	0.86
	1e-3	52	0.87	0.84
	1e-2	60	0.80	0.78
VGG16 + data augmentation	1e-4	55	0.92	0.87
	1e-3	53	0.87	0.85
	1e-2	55	0.72	0.71

Tableau comparatif des performances en fonction des différentes approches

Modèle Simple: Mini-UNet

Le modèle Mini-UNet montre de bons résultats initiaux:

- IoU Train et Test : Les valeurs de l'IoU pour l'entraînement et les tests montrent que le modèle est assez efficace pour la segmentation.
    - IoU Train : 0.81 à 0.83
    - IoU Test : 0.78 à 0.82
- Temps d'entraînement : Environ 44 à 45 minutes, ce qui est relativement rapide.
- Impact du Learning Rate : Le taux d'apprentissage (learning rate) n'a pas eu un impact significatif sur le temps d'entraînement, mais il a légèrement influencé les scores IoU.

Modèle Avancé: VGG16-UNet

Le modèle VGG16-UNet a montré des améliorations significatives par rapport au Mini-UNet :

- IoU Train et Test : Les valeurs de l'IoU pour l'entraînement et les tests sont plus élevées, montrant une meilleure précision.
    - IoU Train : 0.80 à 0.93
    - IoU Test : 0.78 à 0.86
- Temps d'entraînement : Environ 51 à 60 minutes, légèrement plus long que le Mini-UNet en raison de la complexité accrue du modèle.
- Impact du Learning Rate : Un learning rate de 1e-4 a donné les meilleurs résultats, avec un IoU test de 0.86.

Gains Obtenus avec les Approches d'Augmentation des Données

L'augmentation des données a globalement eu un impact positif sur les performances des modèles.

- Mini-UNet avec Augmentation de Données :
    - IoU Train : 0.80 à 0.83
    - IoU Test : 0.76 à 0.82
    - Temps d'entraînement : Environ 47 à 48 minutes, légèrement plus long en raison de l'augmentation des données.
- VGG16-UNet avec Augmentation de Données :
    - IoU Train : 0.72 à 0.92

- IoU Test : 0.71 à 0.87
- Temps d'entraînement : Environ 53 à 55 minutes, montrant que l'augmentation des données peut légèrement augmenter le temps d'entraînement mais améliore la robustesse.

6. Conclusion et Pistes d'Amélioration

La segmentation sémantique est un domaine prometteur avec des applications pratiques variées. Les modèles basés sur Mini-UNet et VGG16-UNet ont montré des résultats prometteurs avec une IoU de test atteignant les 87%. Nous pourrions chercher à augmenter ses performances avec les approches suivantes :

Augmentation Avancée des Données

L'augmentation des données est une technique qui consiste à créer des versions modifiées des images d'entraînement pour augmenter la diversité des données sans collecter de nouvelles images. Pour aller au-delà des techniques de base (flip, ajustement de la luminosité et du contraste), nous pouvons utiliser :

- Transformations Géométriques : Rotation, translation, échelle, cisaillement.
- Ajustements de Couleur : Jittering de couleur, équilibration d'histogramme.

Optimisation des Hyperparamètres

Les hyperparamètres jouent un rôle crucial dans la performance des modèles de réseaux de neurones. Voici quelques hyperparamètres clés à optimiser :

- Batch Size : Essayer différentes tailles de batch (ex: 8, 16, 32, 64) pour trouver un bon équilibre entre la stabilité de l'entraînement et l'efficacité du calcul.
- Nombre de Couches et de Filtres : Ajuster la profondeur du modèle et le nombre de filtres par couche pour trouver la meilleure architecture.
- Dropout Rate : Tester différentes valeurs de dropout pour éviter le surapprentissage (ex: 0.3, 0.5, 0.7).
- Optimizer : Comparer différents optimizers (ex: Adam, SGD, RMSprop) pour voir lequel donne les meilleurs résultats.: Ajustement fin des paramètres du modèle pour améliorer encore les performances.

Ensembles de Modèles

L'utilisation d'ensembles de modèles consiste à combiner plusieurs modèles pour améliorer les performances globales en tirant parti des forces de chaque modèle. Voici quelques stratégies possibles :

- Ensembles de Modèles Similaires : Entraîner plusieurs instances du même modèle (ex: plusieurs VGG16-UNet avec différentes initialisations) et combiner leurs prédictions (technique de bagging).
- Ensembles de Modèles Différents : Combiner des modèles différents comme Mini-UNet, VGG16-UNet, et d'autres architectures populaires comme ResNet-UNet ou DenseNet-UNet. Chaque modèle peut capturer différents aspects des données, et leur combinaison peut améliorer la robustesse.