Exploration d'architectures de réseaux de neurones pour la segmentation sémantique d'images aériennes

Agathe ARCHET[1,2] François ORIEUX[2] Nicolas VENTROUX[1] Nicolas GAC[2]

[1]Thales Research and Technology, 1 Avenue Augustin Fresnel, 91120 Palaiseau, France

[2]Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des Signaux et Systèmes, 3 rue Joliot Curie, 91190 Gif-Sur-Yvette, France

Résumé – La segmentation sémantique d'images aériennes nécessite une extraction complexe d'informations contextuelles. Des Réseaux de Neurones (RdN) convolutifs spécialisés performants se développent pour répondre à ce besoin, mais dans un contexte embarqué, leur structure trop lourde est inadaptée pour une exécution sur cible contrainte en latence ou en consommation d'énergie. Nous proposons donc d'utiliser des méthodes de recherche automatique d'architectures neuronales (NAS) capable de prendre en compte ces contraintes embarquées. Nous démontrons que leur utilisation est possible et peut conduire à d'excellentes performances malgré une complexité réduite. Par exemple, comparé à DC-Swin, la méthode FastNAS atteint une mIoU similaire à 0,838 (-3%) pour une complexité calculatoire réduite de 88% (4,6 GMAC, Multiplication- Accumulations) pour le jeu de données Potsdam.

Abstract – Semantic segmentation of aerial images requires a complex extraction of contextual information. Specialized performant convolutional neural networks (CNNs) are being developed to meet this need, but in an embedded context, their cumbersome structures is unsuitable for execution on a latency or power constrained target. Therefore, we propose to use neural architecture search (NAS) methods capable of taking into account these embedded constraints. We demonstrate that their use is possible and can lead to excellent performance despite reduced complexity. For example,compared with DC-Swin, the FastNAS method achieves a similar mIoU (0.838, or -3%) for a computing complexity redued by 88% (4.6 GMAC, or Mutiply-Accumulates) for the Potsdam dataset.

1 Introduction

La segmentation sémantique consiste à catégoriser chaque pixel d'une image afin de générer une délimitation fine entre des objets d'une même scène. Cette classification pixel par pixel permet ainsi une meilleure identification du contenu de l'image. Elle est notamment utilisée en imagerie médicale pour compléter les diagnostics, et en conduite autonome pour détecter les obstacles sur la route.

Depuis l'utilisation de FCN [1] en 2015, les réseaux de neurones (RdN) ont surpassé les méthodes traditionnelles de segmentation grâce à une bonne hiérarchisation de l'information. En effet, la segmentation sémantique se base sur deux types d'informations : l'information spatiale locale qui traduit les informations géométriques fines au sein d'un voisinage de pixels proches (formes, textures) et l'information contextuelle globale, plus haut-niveau, pour des relations spatiales à plus grande échelle (lieux, environnement).

Pour tenir compte de ces dépendances multi-échelles, l'architecture des RdN convolutifs continue de se complexifier pour mieux exploiter efficacement l'information contextuelle des images. Dans le cas des images aériennes, les RdN doivent impérativement conserver l'information contextuelle des images car les éléments y sont plus petits, variés et nombreux. Ainsi, des RdN complexes spécifiques à la segmentation d'images aériennes se développent pour conserver une information haute-résolution tout le long du réseau. Mais leur structure rend difficile leur optimisation et leur exécution sur une cible embarquée contrainte en latence (ex : 50 millisecondes (ms) ou moins) ou en consommation d'énergie (ex : 60 Watts ou moins).

Nous proposons d'utiliser des méthodes de recherche automatique d'architectures neuronales (ou *Neural Architecture Search*, NAS) capables de prendre en compte ces contraintes embarquées. Mais peu de méthodes NAS pour la segmentation sémantique aérienne existent à ce jour [2] et celles existantes ont un espace de recherche trop complexe conduisant à l'élaboration de réseaux inadaptés pour des cibles embarquées. Ainsi, nous proposons de nous intéresser aux méthodes NAS ayant un espace de recherche plus simple mais initialement conçues pour d'autres problématiques comme la segmentation de rue ou d'objets. Nous étudions donc leur capacité à s'adapter aux enjeux apportés par les images aériennes, si leur conception doit être nécessairement complexifiée, et montrons néanmoins d'une architecture légère et performante peut être obtenue.

Les contributions de cette publication portent sur : (1) l'analyse de méthodes NAS de l'état de l'art en segmentation appliquées aux images aériennes, (2) l'étude de l'efficacité de leur espace de recherche par l'étude des tendances topologiques des RdN générés, et (3) le positionnement de leurs RdN générés vis-à-vis de RdN manuels optimisés.

La suite de cet article est organisée de la façon suivante. La section 2 présente les tendances de conception de RdN en segmentation d'images aériennes, la section 3 introduit ensuite deux méthodes NAS en segmentation classique. Le plan d'expérience est détaillé dans la section 4. Les résultats de ces expériences figurent dans la section 5. Finalement, une conclusion est apportée en section 6.

2 Les méthodes NAS pour la segmentation sémantique

Les réseaux de neurones pour la segmentation sémantique ont une topologie plus complexe que les réseaux de classification. Pour les images aériennes, trois familles de réseaux existent. Il y a les RdN qui combinent des combinaisons

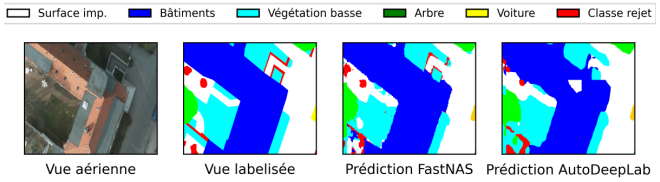

FIGURE 1 – Détail d'une prédiction sur Potsdam (top_3_13)

d'opérations multi-échelles fixes avec un champ récepteur limité [3], les Transformers qui modélisent les dépendances à grande échelle avec des modules d'attention lourds en calcul [4], et enfin les réseaux de neurones en graphes qui représentent des liens flexibles entre les pixels par des graphes au détriment d'un coût calculatoire exponentiel [5]. Pour des raisons d'embarquabilité, nous considérons uniquement les RdN dans la suite de cet article. L'enjeu est donc de concevoir des RdN conservant une information contextuelle d'images haute-résolution jusqu'à la fin du réseau.

En parallèle de la conception manuelle et optimisée de RdN, la recherche automatique d'architectures neuronales a émergé pour automatiser le processus difficile d'optimisation. Elles se distinguent par leur espace de recherche R (ensembles de squelettes et d'opérations autorisés), leur algorithme d'optimisation et leur(s) critère(s) C à optimiser (ex : le débit ou la latence). Pour un jeu de données d, le problème d'optimisation NAS se définit ainsi :

$$topologie^* = \underset{topologie \in R}{optimiser} \quad C(topologie, poids, d) \qquad (1)$$

En segmentation, leur emploi a conduit à des réseaux plus légers, plus rapides ou plus performants. Par exemple, AutoDeepLab [6] est plus précis sur CityScape et est plus léger que sa version manuelle dont il est issu (DeepLabV3 [7]). Depuis, d'autres méthodes NAS pour la segmentation ont enrichi leur espace de recherche avec de la fusion multi-échelle (FastNAS [8]), des couches densément connectées (DCNAS [9]), des connexions multi-échelles (DNAS [2]), ou des convolutions spécialisées (FasterSeg [10]). Un exemple de leurs prédictions sur une image du jeu de données Potsdam est représenté en Figure 1.

Dans la suite du papier, nous nous focaliserons sur l'étude des méthodes AutoDeepLab et FastNAS qui présentent des bonnes prédispositions par leur structure pour répondre au besoin de la segmentation sémantique d'images aériennes.

3 AutoDeepLab et FastNAS

L'espace de recherche d'AutoDeepLab est sélectionné comme base de comparaison simple de type encodeur-décodeur. Comme illustré Figure 2a, cet espace s'organise sur deux niveaux de finesse. Un premier sous-espace macroscopique détermine la structure globale du réseau, à travers le réglage d'un facteur d'échantillonnage (qui fixe la résolution des entrées et le nombre de filtres des opérations) pour 12 couches. À cela s'ajoute un second sous-espace qui détermine un bloc d'opérations identique pour chaque couche. Cet espace autorise la sélection de 10 opérations et de leur agencement au sein du bloc. Les opérations possibles sont des convolutions séparables 3x3 ou 5x5, dilatées 3x3 ou 5x5, des opérations de poolings, des identités ou aucune connexion. Les opérations choisies sont groupées deux à deux et peuvent être exécutées en parallèle ou les unes à la suite des autres.

L'autre espace de recherche, celui de FastNAS, est sélectionné comme exemple plus complexe permettant des connexions multi-échelles. Contrairement à AutoDeepLab, il ne se focalise que sur la deuxième partie d'un RdN comme illustré en Figure 2b. Le classifieur MobileNetV2 fait office d'encodeur du réseau et précède un décodeur construit sur deux niveaux de recherche. Un premier sous-espace détermine la structure globale du décodeur, à travers l'agencement et les données d'entrée de 6 blocs d'opérations tous identiques. Les données d'entrée des blocs peuvent venir de la couche finale de l'encodeur ou de ses couches intermédiaires, et l'organisation des blocs peut être consécutive ou parallèle. Le deuxième sous-espace est semblable à celui d'AutoDeepLab, à la différence que plus d'opérations interviennent (au nombre de 7) et que les convolutions séparables diluées sont disponibles. L'agencement des opérations est également plus flexible.

4 Plan d'expérience

Cette section décrit d'une part les conditions expérimentales sur lesquelles s'appuieront nos résultats d'analyse des deux méthodes NAS, AutoDeepLab et FastNAS, pour de la segmentation d'images aériennes, et d'autre part la comparaison des meilleurs réseaux obtenus par méthode NAS avec d'autres réseaux manuellement optimisés de l'état de l'art.

4.1 Adaptation des méthodes NAS

Tout d'abord, l'algorithme de gradient d'AutoDeepLab n'a pas été conservé pour permettre la comparaison de critères autres que le score de segmentation. La structure de super-réseau a été décomposée pour revenir à des algorithmes stochastiques sans gradient. Des opérations de mutations ont été ajoutées pour utiliser un algorithme génétique.

Ensuite, pour une meilleure comparaison sur la topologie, l'algorithme de Renforcement Learning et les cellules auxiliaires de FastNAS n'ont pas été utilisés. Des opérations de mutations ont également été ajoutées.

4.2 Jeux de données

L'entraînement et la validation des réseaux de neurones seront effectués sur deux jeux de données d'images aériennes fournis par la ISPRS [11], nommés Potsdam et Vaihingen. Leur résolution spatiale est de l'ordre de 5 cm et six classes sont annotées : les surfaces imperméables, les bâtiments, la végétation basse, les arbres, les voitures et l'arrière-plan (classe de rejet).

Le jeu de données Postdam rassemble 38 images orthographiques 6000x6000 d'une ville historique. Ces images sont encodées en canaux RGB avec une bande supplémentaire dédiée aux données d'élévation (DSM). Le jeu de données Vaihingen contient 33 images orthographiques 2494x2064 d'un petit village. Les canaux des images sont l'infra-rouge, le rouge et le vert en plus de l'élévation.

Il est à noter que des distorsions existent à cause des opérations de post-traitement lors de la création des images, ce qui peut modifier les contours. Par la suite, seuls les canaux couleurs et les vérités terrain sans frontières entre les classes seront utilisés. Une augmentation des données sera réalisée avec le projet Geoseg [12] pour améliorer l'apprentissage et faire davantage apparaître les classes difficiles à segmenter (ex : la classe voiture). Des sous-images de dimension 512x512 seront utilisées pour l'entrainement.

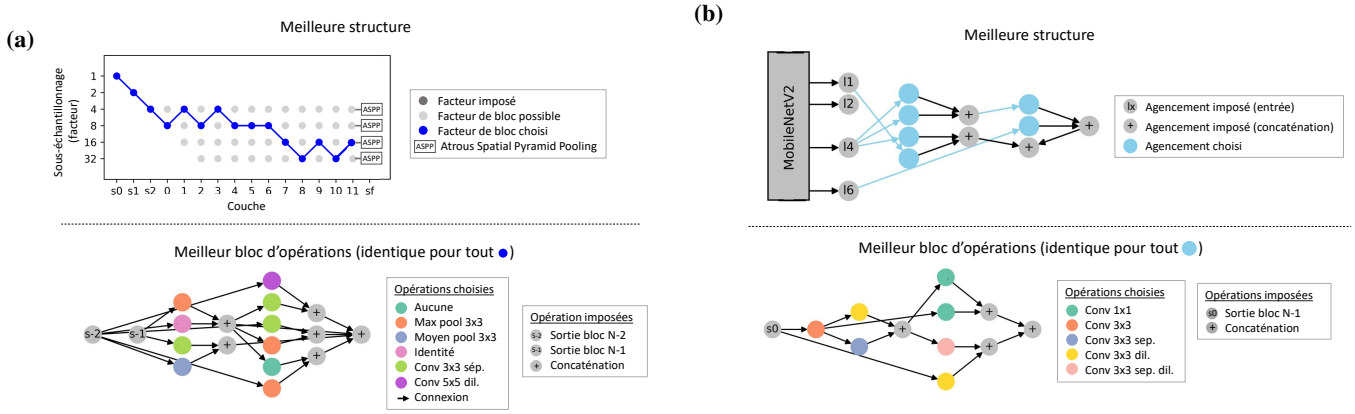

FIGURE 2 – Meilleures architectures obtenues pour AutoDeepLab (a) et FastNAS (b) sur Potsdam.

4.3 Détails d'implémentation

Les méthodes NAS employées par la suite reposent sur des algorithmes itératifs. Pour initialiser un algorithme, un ensemble de réseaux aux architectures différentes doit être entrainé pour constituer une population initiale de topologies. Ensuite, lors d'une itération, un réseau est créé parmi l'ensemble des topologies disponibles, puis entrainé sur 30 passages complets du jeu de données selon une fonction de coût. Cette dernière est une combinaison d'une entropie croisée moyennée sur les labels (label smothing cross-entropy) et de l'indice de Sørensen-Dice (dice loss). Une fois entrainé, ce réseau est comparé aux autres réseaux déjà créés. Suivant l'algorithme d'optimisation, ce réseau sera conservé et un nouveau sera généré de façon aléatoire (algorithme aléatoire), ou sera permuté avec un réseau moins performant dans la population en cours avant une prochaine mutation (algorithme génétique).

4.4 Paramètres expérimentaux

Lors des prochains tests, les topologies issues de méthodes NAS sont analysées en faisant varier les paramètres suivants : l'espace de recherche (AutoDeepLab ou FastNAS), l'algorithme d'optimisation (aléatoire, génétique ou génétique modifié), le critère à optimiser (le score de segmentation, ici l'Intersection moyenne sur l'Union ou mIoU de test, la latence sous Pytorch 1.12 sur un GPU Nvidia V100 avec CUDA 10.2, le nombre de paramètres du réseau), et enfin le nombre d'itérations (réseaux évalués par NAS).

En particulier, les meilleures topologies obtenues seront comparées avec des réseaux de l'état de l'art sur le score de segmentation. Néanmoins, il est à prendre en compte que la plupart des réseaux de l'état de l'art n'utilisent pas uniquement les canaux de couleurs en données d'entrée contrairement à notre méthode.

5 Résultats et analyse

Cette section détaille l'influence des paramètres d'optimisation des méthodes NAS et effectue une comparaison avec les réseaux manuels ou issus de NAS de l'état de l'art.

5.1 Impact de l'optimisation pour les NAS

Afin d'explorer efficacement l'espace de recherche par les méthodes FastNAS (10^{12} combinaisons) et AutoDeepLab (10^{19}), il est possible de jouer sur l'algorithme d'optimisation, le critère à optimiser, et le jeu de données.

5.1.1 Convergence selon l'algorithme d'optimisation

La convexité de l'espace combinatoire des architectures pour NAS n'est pas toujours garantie. Ainsi, pour mieux éviter les problèmes de minima locaux, nous avons implémenté 3 méthodes d'optimisation, dites *boîte noire*, à la place des algorithmes de gradient et de renforcement initiaux. Les deux premiers algorithmes sont un algorithme aléatoire et un algorithme génétique basique faisant intervenir des mutations. Pour le troisième algorithme, nous avons ajouté à l'algorithme génétique classique une contrainte de population initiale favorable pour obtenir de meilleurs résultats. Généralement, les algorithmes aléatoires sont utilisés comme base de comparaison, et les algorithmes génétiques sont capables de proposer de meilleurs résultats en exploitant les caractéristiques des meilleurs candidats par des mécanismes d'évolution (mutations, éliminations, ...).

Nos résultats d'expérimentation montrent néanmoins que les 3 algorithmes mis en œuvre peuvent chacun conduire à une meilleure solution selon le jeu de données et la méthode utilisés. Ceci est illustré en Table 1, où l'on constate notamment que l'algorithme aléatoire est une base de comparaison solide. Par ailleurs, nos expérimentations sur 6 autres tests de 500 itérations ont montré que les algorithmes génétiques stagnent au-delà de 75 itérations. L'espace des combinaisons possibles pour AutoDeepLab et FastNAS semble donc particulièrement non-convexe. Néanmoins, l'espace de FastNAS est plus expressif, car il reste plus compact tout en offrant de meilleurs scores qu'AutoDeepLab.

TABLE 1 – Meilleures mIoU obtenues avec les NAS

ConfigurationNAS		Meilleures performances		Meilleurs algorithmes	
Dataset	Espace	mIoU test	Latence (ms)	Nom	Iter.
Vaihingen	AutoDeepLab	0.703	21.8	Génétique *	50
	FastNAS	0.791	27	Aléatoire	50
Potsdam	AutoDeepLab	0.753	24.8	Génétique *	500
	FastNAS	0.838	18	Génétique	50

* Initialisation avec des architectures à score correct

5.1.2 Analyse topologique selon le critère optimisé

Cette section analyse les tendances topologiques explorées par les méthodes NAS pour un unique critère parmi le nombre de paramètres, la latence ou le score sémantique. Pour cela, nous nous appuyons sur les différentes architectures générées précédemment lors des tests de convergence des algorithmes.

La base de connaissance produite est constituée de 1670 architectures différentes. 660 architectures sont optimisées sur le nombre de paramètres, 350 sur la latence, et 660 sur le score sémantique.

Critère de latence : En optimisant la latence, les deux espaces de recherche de FastNAS et AutoDeepLab favorisent des topologies séquentielles et peu profondes. Les latences peuvent être plus importantes pour AutoDeepLab (de 16 à 390 ms) que pour FastNAS (13,3 et 35,5 ms). Cette différence s'explique par la présence d'opérations plus complexes autorisées par AutoDeepLab, lorsque le nombre de filtres devient grand.

Critère du nombre de paramètres : Comme pour la latence, les méthodes NAS ont tendance à converger vers des topologies peu profondes (nombre de filtres moyen ou petit) pour ne pas inclure d'opérations lourdes en paramètres. Pour les deux espaces de recherche, il existe une variation de 1 million de paramètres entre les topologies les plus lourdes et les plus légères (de 1,9 à 2,9 M).

Critère du score (mIoU de validation) : Contrairement aux critères précédents, optimiser le score sémantique apporte une plus grande diversité de topologies performantes. Nos expérimentations montrent que les topologies combinant fortement des informations provenant de différentes échelles obtiennent les meilleurs résultats. En particulier, FastNAS, qui privilégie des données d'entrées multi-échelles issues de son encodeur, génère de nombreuses topologies performantes.

Pour conclure, nous observons que la séquentialité des blocs et des opérations peut satisfaire les trois critères à la fois. Par ailleurs, le traitement d'une information contextuelle, utile pour les images aériennes, se dégage de la structure macroscopique des topologies.

5.1.3 Analyse topologique selon le jeu de données

Cette section compare les différences de topologies entre nos réseaux les plus performants, obtenues avec le jeu de données Vaihingen ou Postdam, et ceux obtenus dans les publications originales. Les meilleures topologies pour le jeu de données Postdam et Vaihingen sont représentées sur la Figure 2.

Dans [6], les auteurs d'AutoDeepLab utilisent le jeu de données de rue urbaine Cityscapes. Entre leur solution et celle présentée en Figure 2.a, notre topologie montre une meilleure utilisation de l'information multi-niveaux au sein d'un bloc avec des connexions séquentielles et parallèles. Aussi plus d'opérations de pooling sont utilisées, possiblement pour favoriser l'information multi-échelle. Leurs allures macroscopiques restent cependant similaires.

Les auteurs de FastNAS, quant à eux, ont utilisé les jeux de données d'objets proches COCO et PASCAL VOC. Contrairement à leur topologie, notre architecture (Figure 2.b) utilise plus de sorties intermédiaires de l'encodeur MobileNetV2 plutôt que les toutes dernières comme données d'entrée des blocs. Les blocs et les opérations sont agencés de manière plus séquentielle, et les convolutions diluées sont davantage utilisées pour favoriser l'information multi-échelle comme pour AutoDeepLab.

5.2 Comparaison avec l'état de l'art

Comparons maintenant la performance des méthodes FastNAS et AutoDeepLab avec d'autres réseaux manuels ou issus de NAS de l'état de l'art. Comme le montre la Figure 3, AutoDeepLab et FastNAS représentent un compromis intéressant et arrivent même à mieux se positionner que d'autres RdN pourtant optimisés manuellement pour le traitement d'images aériennes. Par exemple, FastNAS atteint un mIoU de 0,838 pour une complexité calculatoire de 4,6 GMAC.

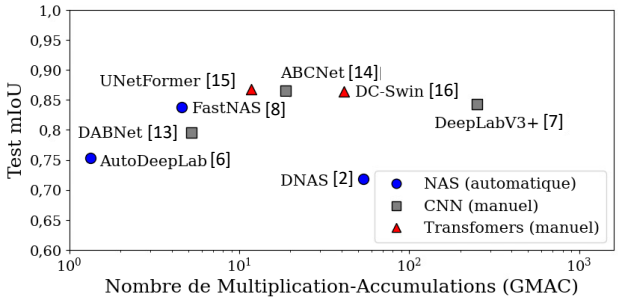

FIGURE 3 – Comparaison des méthodes NAS avec l'état de l'art sur Potsdam

6 Conclusion

Deux méthodes de recherche automatique d'architectures neuronales (NAS), initialement prévues pour de la segmentation sémantique de rue, ont été adaptées puis optimisées sur les jeux de données Potsdam et Vaihingen. Après analyse des paramètres d'optimisation, nous avons constaté que leurs topologies générées s'adaptent réellement pour mieux prendre en compte l'information contextuelle recherchée. Les espaces de recherche de FastNAS et AutoDeepLab parviennent à s'aligner sur les autres modèles de l'état de l'art en offrant un très bon compromis entre le score sémantique et la complexité calculatoire. Un effort doit cependant être réalisé au niveau de l'algorithme d'optimisation pour prévenir des minimums locaux.

Références

[1] Long et COL. : Fully convolutional networks for semantic segmentation. *In CVPR*. IEEE, 2015.

[2] Wang et COL. : DNAS : Decoupling neural architecture search for high-resolution remote sensing image semantic segmentation. *Remote Sens.*, 14, 2022.

[3] Zhao et COL. : Pyramid scene parsing network. *In Proc. CVPR*. IEEE, 2017.

[4] Ding et COL. : LANet : Local attention embedding to improve the semantic segmentation of remote sensing images. *TGRS*, 2021.

[5] Liu et COL. : Self-constructing graph convolutional networks for semantic labeling. *In IGARSS*. IEEE, 2020.

[6] Liu et COL. : Auto-DeepLab : Hierarchical neural architecture search for semantic image segmentation. *In Proc. CVPR*. IEEE, 2019.

[7] Liang-Chieh Chen et COL. : Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv :1706.05587*, 2017.

[8] Nekrasov et COL. : Fast neural architecture search of compact semantic segmentation models via auxiliary cells. *In Proc. CVPR*. IEEE, 2019.

[9] Zhang et COL. : DCNAS : Densely connected neural architecture search for semantic image segmentation. *In Proc. CVPR*. IEEE, 2021.

[10] Chen et COL. : FasterSeg : Searching for faster real-time semantic segmentation. *arXiv preprint arXiv :1912.10917*, 2019.

[11] Rottensteiner et COL. : ISPRS semantic labeling contest.

[12] Wang et COL. : GeoSeg open-source semantic segmentation toolbox for pytorch.

[13] Li et COL. : DABNet : Depth-wise asymmetric bottleneck for real-time semantic segmentation. *In BMVC*, 2019.

[14] Li et COL. : ABCNet : Attentive bilateral contextual network for efficient semantic segmentation of fine-resolution remotely sensed imagery. *ISPRS J. Photogramm. Remote Sens*, 2021.

[15] Wang et COL. : UNetFormer : A unet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery. *ISPRS J. Photogramm. Remote Sens*, 2022.

[16] Wang et COL. : A novel transformer based semantic segmentation scheme for fine-resolution remote sensing images. *GRSL*, 2022.

Ces travaux ont bénéficié d'un accès aux moyens de calcul de l'IDRIS au travers de l'allocation de ressources 2021-AD011013191 attribuée par GENCI