Segmentation sémantique d'images aériennes avec améliorations interactives

G. Lenczner[1, 2] B. Le Saux[1] N. Luminari[2] A. Chan-Hon-Tong[1] G. Le Besnerais[1]

[1] ONERA / DTIS, Université Paris-Saclay, F-91123 Palaiseau, France
[2] Delair, FR-31400 Toulouse, France

gaston.lenczner@delair.aero, bls@ieee.org, nicola.luminari@delair.aero,
{adrien.chan_hon_tong, guy.le_besnerais}@onera.fr

Résumé

Nous présentons une approche interactive pour la segmentation multi-classe d'images aériennes. Celle-ci repose sur un réseau de neurones profond qui exploite à la fois des images RGB et des annotations. À partir d'une sortie initiale qui dépend uniquement de l'image, notre réseau affine ensuite de manière interactive ce résultat en utilisant une concaténation de l'image et des annotations de l'utilisateur. Il est important de noter qu'il n'y a pas de réentraînement durant les interactions, ce qui permet un processus rapide et flexible. Dans le contexte d'images extrêmement résolues telles que celles acquises par drone, nous montrons qu'un humain-dans-la-boucle est très gratifiant, quel que soit l'environnement expérimental et l'architecture du réseau utilisée.

Mots Clef

Segmentation Sémantique, Réseaux de neurones profonds, Interactif, Imagerie Aérienne, Humain dans la boucle

Abstract

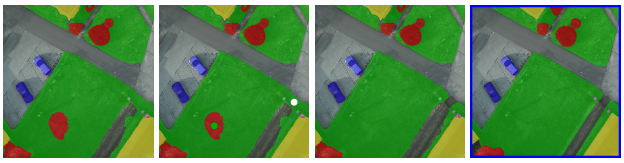
We present an interactive approach for multi-class segmentation of aerial images. It consists in a deep neural network which takes as input RGB images and annotations. Initial segmentation is image-based only. During interactions, user inputs are processed by the network trained on virtual annotations. It is noteworthy there is no retraining during the interactions, enabling a fast and smooth process. In the context of extremely high resolution images such as the ones captured by drone, we show that a human-in-the-loop is extremely rewarding whatever the experimental environment and the network architecture used.

Keywords

Semantic Segmentation, Deep Neural Networks, Interactive, Aerial Images, Human-in-the-loop

1 Introduction

La compréhension de scènes s'est grandement développée ces dernières années en s'appuyant sur des techniques d'apprentissage profond. Des algorithmes d'apprentissage sont d'ailleurs maintenant déployés dans certaines industries. Cependant, puisque de tels algorithmes dépendent par essence des conditions d'apprentissages, il n'est pas garanti que leurs résultats atteignent la précision attendue par les utilisateurs. Ainsi, une supervision humaine est encore souvent nécessaire pour valider la qualité des résultats.



1 - Résultat initial | 2 - Phase d'annotation | 3 - Résultat affiné | Vérité-terrain

Figure 1 – Exemple de l'approche de segmentation sémantique interactive sur le jeu de données ISPRS Potsdam [30]. Un point vert et un point blanc représentent les interactions.

Nous nous concentrons dans cet article sur la segmentation sémantique d'images aériennes à haute résolution, typiquement obtenues par drone. Cette tâche consiste à classer les images au niveau pixelique à des fins de cartographie par exemple. Ce problème est maintenant efficacement abordé en apprentissage supervisé par des réseaux de neurones convolutifs [11]. Sous certaines conditions (par exemple, lorsque suffisamment de données d'apprentissage sont disponibles), ces algorithmes atteignent des résultats proches de la vérité-terrain sur des données nouvelles. Cependant, les erreurs restantes peuvent visuellement faire une grande différence et ne sont pas tolérables dans une majorité d'applications pratiques. En outre, les performances de ces algorithmes se dégradent souvent sur les jeux de données de la vie réelle en raison de divers facteurs : données complexes, absence de vérité-terrain bien annotée, adaptation de domaine, etc.

Pour obtenir des cartes de segmentation précises de façon optimale, nous proposons dans cet article une procédure pour qu'un utilisateur affine itérativement les cartes de segmentation produites par un réseau de neurones selon les étapes décrites en Figure 1. En effet, un humain peut faci-

lement repérer les zones mal classées et ainsi les corriger. La difficulté est alors d'atteindre une classification optimale tout en gardant l'ensemble du processus rapide et suffisamment engageant pour un utilisateur. Pour y parvenir, le réseau de neurones de notre approche est pré-entraîné avec des annotations virtuelles concaténées avec l'image. Ainsi, les annotations réelles de l'utilisateur corrigent efficacement et à la volée la segmentation initiale, permettant de converger vers une carte précise.
Nos contributions sont les suivantes.

1. Nous proposons un **processus interactif de segmentation pour des images aériennes utilisant l'apprentissage profond**. Une fois le réseau de neurones entraîné, notre algorithme ne nécessite aucun ré-entraînement et peut donc affiner rapidement les cartes de segmentation.
2. S'inspirant de travaux de vision par ordinateur principalement orientés vers la segmentation binaire, notre travail les étend à **la segmentation multi-classe**.
3. Nous montrons à travers une validation expérimentale que notre approche est valide sur des jeux de données aériens et notamment pour les **images acquises par drone** et ce indépendamment de l'architecture sous-jacente du réseau de neurones.

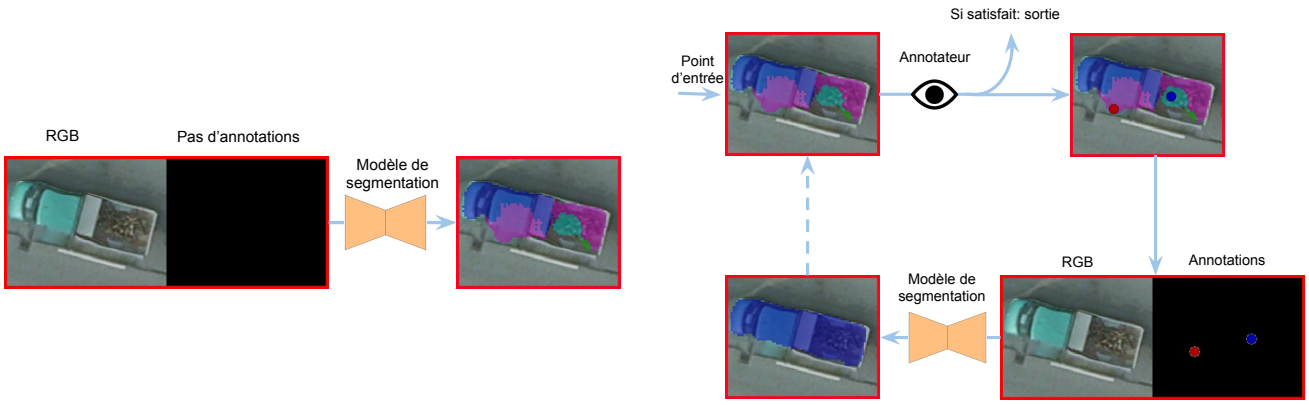
La suite de ce papier est organisée comme suit. Nous décrivons d'abord l'état de l'art dans 2. Nous présentons ensuite notre approche dans la section 3 puis nous discutons de la manière de l'évaluer dans la section 4. Nous détaillons enfin nos expériences dans la section 5.

2 Revue de l'état de l'art

La segmentation interactive a été abordée en vision par ordinateur avec une grande variété de méthodes au cours des deux dernières décennies. Les plus anciennes sont généralement des méthodes de type *graphcut* [6, 29, 13] ou basées sur des forêts aléatoires [31, 33]. Alternativement, [7] aborde la segmentation interactive en se basant sur une représentation morphologique de l'image qui ne nécessite pas d'apprentissage statistique. Plus récemment, les meilleures performances en segmentation sémantique ont été obtenues avec des architectures basées sur réseaux de neurones convolutifs [20]. Plusieurs travaux ont ensuite tenté de les rendre interactifs pour obtenir des résultats plus fins. Ces approches utilisent généralement des points cliqués par des utilisateurs comme annotations. Nous examinons maintenant en détail ces méthodes.
Deep Interactive Object Selection (DIOS) [36] est la première proposition d'un procédé de segmentation interactif basé sur des réseaux de neurones. Celui-ci vise à une classification binaire. Le réseau prend en entrée deux canaux supplémentaires concaténés avec l'image RGB. Le premier contient des points d'annotation du premier plan tandis que l'autre contient ceux d'arrière-plan. Ces points d'annotation représentent l'intérieur des différentes instances et sont encodés avec des cartes de distance euclidiennes. Les annotations sont automatiquement générées pendant l'entraînement à l'aide des cartes de vérité-terrain. Plusieurs travaux étendent DIOS. [18] adopte une stratégie multi-échelles qui affine la prédiction globale en la combinant avec une classification de patchs d'image locaux centrés sur les annotations. [16] suit également une stratégie multi-échelles en concevant un réseau fusionnant deux flux pour traiter les annotations différemment de l'image. Un enjeu particulier du problème consiste à obtenir suffisamment d'annotations utiles. À cette fin, [21] sélectionne lors de l'entraînement des annotations parmi les prédictions erronées. Alternativement, [17] optimise de manière itérative les cartes d'annotation données en entrée en rétro-propageant les erreurs entre les prédictions et les annotations qui sont alors considérées comme de la vérité-terrain. Enfin, DEXTR [22] et [35] demandent tous deux à l'utilisateur de cliquer sur des points sur les bordures et les coins des objets. Récemment, [5] a évalué ces différentes stratégies dans la première étude à grande échelle de segmentation d'instance interactive avec des annotateurs humains. Leurs expériences suggèrent que les clics d'annotation centraux sont plus robustes que ceux sur les bordures et que la transformation de distance pour encoder les points d'annotation peut être remplacée par des disques binaires.

Polygon-RNN++ [1] est une alternative intéressante aux approches de type DIOS. En utilisant une architecture convolutionnelle et récurrente, ils prédisent un polygone qui peut être affiné en déplaçant ses sommets. En utilisant un réseau convolutionnel de graphes (GCN), Curve-GCN [19] étendent ce travail en prédisant une spline qui décrit mieux les objets courbes qu'un polygone. Le point commun de toutes les approches susmentionnées est qu'elles visent à la classification binaire.

La segmentation multi-classe interactive a également été abordée de différentes manières. Plusieurs méthodes [25, 24] tentent de résoudre ce problème en utilisant une approche bayésienne maximale a posteriori (MAP) tandis que [32] s'appuie sur un classificateur Random Forest. [23] aborde ce problème en se basant sur des super-pixels et des SVMs. Récemment, [3] utilise une combinaison de deux Mask-RCNN [14] légèrement modifiés pour calculer plusieurs propositions de segmentation. Ensuite, l'utilisateur choisit lesquelles de ces propositions doivent former la segmentation finale. Enfin, [2] est la première proposition d'une approche d'apprentissage profond qui permet à l'utilisateur de corriger une segmentation sémantique multi-classe proposée. Leur algorithme prend en entrée une concaténation de l'image et des points extrêmes de chaque instance de la scène puis corrige la proposition de segmentation à l'aide de simples tracés fournis par l'utilisateur. A contrario, nous adoptons dans notre travail une approche basée sur DIOS pour affiner une carte de segmentation multi-classe initiale.



Phase d'initialisation

Boucle interactive

Figure 2 – Vue d'ensemble de l'approche proposée : l'information fournie par l'utilisateur modifie seulement les entrées du réseau - pas ses paramètres - ce qui permet une interaction fluide avec l'utilisateur

3 Algorithme proposé

Nous décrivons maintenant en détail l'approche proposée pour la segmentation multi-classe interactive d'images aériennes. En particulier, notre objectif est d'entraîner un réseau de neurones avec deux objectifs :

1. produire une carte de segmentation initiale de la scène de haute qualité sans aucune aide extérieure ;
2. utiliser des annotations fournies par un opérateur pour améliorer rapidement sa prédiction initiale.

Pour y parvenir, nous proposons un réseau de neurones qui conserve sa structure d'origine mais prend en entrée une concaténation des entrées classiques (ex : RGB) et des annotations (N canaux, un par classe). Dans le cas RGB, le réseau traite donc des entrées composées de $3+N$ canaux. Les annotations sont des points cliqués. Il est important de noter que seules les entrées du réseau sont modifiées et non ses poids, ce qui fait la rapidité de l'approche. La Figure 2 présente une vue d'ensemble de notre approche.

3.1 Représentation des annotations

La correction d'une mauvaise segmentation implique de fournir au système des informations supplémentaires sur la sémantique de l'image. Les nouveaux échantillons fournis par les clics peuvent alors représenter soit l'intérieur d'une instance [36], soit sa bordure [22]. Dans notre étude, les annotations représentent l'intérieur des instances car ceci suit les conclusions de [5] et paraît être le plus adapté pour un utilisateur dans notre cas de segmentation multi-classe. De plus, les annotations peuvent être encodées de différentes manières, ce qui peut fournir au système des informations plus ou moins étendues spatialement. Nous encodons nos clics à l'aide de transformées de distances euclidiennes qui permettent de propager localement les informations de localisation.

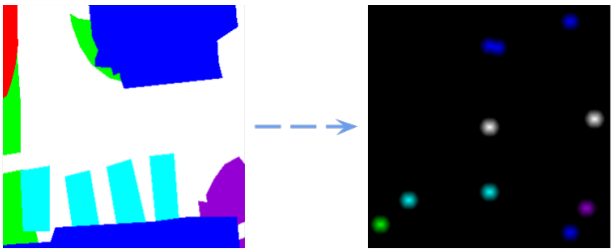


Figure 3 – Échantillonnage d'annotations à partir de la vérité-terrain. En pratique, celles-ci sont encodées dans les N canaux d'annotations

3.2 Stratégie d'entraînement

Dans ce qui suit, nous supposons que nous avons un jeu de données de segmentation sémantique composé de N classes. Les cartes de vérité-terrain sont au cœur de notre stratégie d'entraînement. D'une part, elles sont classiquement utilisées pour calculer et rétro-propager la perte. D'autre part, comme le montre la Figure 3, elles sont également utilisées pour *échantillonner aléatoirement des annotations*. En d'autres termes, des cartes de vérité-terrain parcimonieuses sont utilisées comme annotations. Selon leur classe, ces annotations sont encodées dans les N canaux d'annotation donnés en entrée de l'algorithme. Pour que le réseau soit robuste à différentes dispositions d'annotations, le nombre d'annotations simulées est aléatoire dans chaque exemple d'apprentissage. Étant donné que le réseau doit également pouvoir créer une carte de segmentation précise sans elles, une absence d'annotations est également possible. Concrètement, cette situation signifie que les canaux d'annotation sont remplis uniquement de zéros.

4 Stratégie d'évaluation

Pour évaluer des approches interactives, deux situations sont possibles. Soit la vérité-terrain est disponible et peut

être utilisée pour simuler des annotations, soit elle ne l'est pas et un opérateur humain doit effectuer les annotations. Par conséquent, nous avons évalué notre approche à la fois manuellement et automatiquement. Nous utilisons l'Intersection sur l'Union (IoU) moyennée sur toutes les classes comme métrique d'évaluation.

Évaluation automatique Pour chaque image, le réseau de neurones fait une première inférence sans annotations. Un clic est alors simulé et le réseau fait une nouvelle inférence. Ce processus est répété de manière itérative pour un nombre fixe de clics. En utilisant une comparaison entre la prédiction et la vérité-terrain, les clics sont simulés vers le centre de l'une des plus grandes zones mal classifiée de l'image. De l'aléatoire est ajouté dans le choix de la zone et dans la localisation du clic à l'intérieur de celle-ci pour mieux simuler un comportement humain. Si des classes sont sous-représentées par rapport à d'autres dans le jeu de données, nous contraignons les clics à être répartis dans les différentes classes pour s'assurer que les corrections bénéficient à toutes les classes présentes dans le jeu de données.

Évaluation manuelle Dans ce cas, les clics sont désormais effectués par un opérateur humain. Cet opérateur vise également à corriger les plus grosses zones d'erreur mais la localisation des clics est alors intrinsèquement subjective. Pour effectuer cette évaluation manuelle, nous avons construit un plugin QGIS [26]. L'interaction utilisateur est gérée par QGIS tandis que les calculs pour la segmentation sémantique sont effectués dans un serveur séparé qui peut être local ou distant. Une fois le serveur lancé, le transfert de données est transparent pour l'utilisateur.

5 Expériences

Dans cette section, nous visons tout d'abord à montrer que notre méthode fonctionne, ce que nous faisons avec notre procédé d'évaluation automatique décrit dans la section 4. Ensuite, nous comparons différentes architectures de réseaux de neurones pour évaluer si cela a un impact significatif sur les performances. De plus, ces différentes architectures produisant des cartes de segmentation initiales différentes, cette comparaison nous permet également d'étudier si la qualité initiale des cartes de segmentation a une influence sur les avantages apportés par les annotations.Enfin, nous analysons la portée ainsi que les limites de notre approche avec un humain dans la boucle.

5.1 Protocole expérimental

Jeux de données. Nous avons testé notre approche sur deux jeux de données d'images aériennes génériques et un jeu dédié à l'imagerie par drone. Nous avons divisé les jeux d'entraînement initiaux en sous-jeu d'entraînement et de validation avec un rapport de 80%-20%. Ayant ainsi accès aux cartes de vérité-terrain pour simuler les annotations, nous utilisons ces jeux de validation pour nos expériences.

1. Le jeu de données Aerial Imagery for Roof Segmentation (AIRS) [12] est composé de deux classes (*bâtiments* et *non bâtiments*) et couvre 457 km^2 sur Christchurch en Nouvelle Zélande. La résolution spatiale est de 0.075 m et la taille de chaque image est de 10000×10000. Le jeu d'entraînement initial est composé de 951 images. La classe *bâtiment* est sous-représentée : 148 images ne contiennent aucun bâtiment et les pourcentages moyens et medians de pixels *bâtiment* par image sont respectivement de 7.6% et 2.1%.
2. Le jeu de données ISPRS Potsdam [30] est composé de 6 classes (*route, bâtiment, végétation basse, arbre, voiture* et *inclassable*). La classe *voiture* est sous-représentée par rapport aux autres classes. Ce jeu de données couvre environ 3 km^2 avec une résolution spatiale de 0.05 m. La taille de chaque image est de 6000×6000 pixels. Le jeu d'entraînement initial est composé de 24 images.
3. Le jeu de données DroneDeploy Segmentation Dataset (DDSD) [1] est composé 6 classes (*bâtiment, inclassable, végétation, eau, sol* et *voiture*). La classe *voiture* est aussi sous-représentée dans ce jeu de données. Il contient 55 images de taille variable mais de résolution spatiale constante de 0.01m et contenant diverses scènes capturées par drones. Les images de ce jeu de données sont variées (milieux urbains, agricoles, industriels, ...) ce qui rend l'apprentissage de sa sémantique plus complexe que celle des autres jeux de données.

Réseau de neurones. Sauf dans la comparaison des architectures, nous utilisons une architecture LinkNet [9]. Il s'agit d'une architecture de type encodeur/décodeur classique reposant sur un encodeur ResNet [15]. Les réseaux sont entraînés à l'aide de descentes de gradient stochastique sur 50 époques avec des *batchs* de taille 8. Ils voient à chaque époque 10000 échantillons choisis aléatoirement parmi le jeu d'entraînement et recadrés en patchs d'images de taille 512×512. Le taux d'apprentissage initial est fixé à 0.05 et est divisé par 10 après 15, 30 et 45 époques. Seule une augmentation basique des données est effectuée : inversions horizontales et verticales. L'implémentation est faite à l'aide de Pytorch.

Annotations. Au cours de nos différentes évaluations, nous simulons 120 clics par image - correspondant à un processus de 6 minutes avec 20 clics par minute - et mesurons le gain en IoU pour chaque classe. Mis à part dans la section 5.4, les annotations sont simulées automatiquement comme décrit dans la section 4.

5.2 Validation de l'approche

Les résultats du Tableau 1 valident l'efficacité de notre approche sur les trois jeux de données. En effet, les performances de segmentation sont améliorées pour chaque classe : en moyenne, l'IoU moyenne est augmenté de 4,2% sur ISPRS Potsdam, de 9.2% sur AIRS et de 13% sur

1. https ://github.com/dronedeploy/dd-ml-segmentation-benchmark

	non bâtiment	bâtiment	moyenne
Avant	98.4	72.2	85.3
Après	98.9	90.1	94.5

(a) AIRS

	rou.	bât.	vég.	arb.	voi.	inc.	moy.
Avant	80.9	83.4	68.2	66.7	80	44.5	70.6
Après	84.5	86.5	72	70.5	82.8	52.5	74.8

(b) ISPRS

	bât.	inc.	vég.	eau	sol	voi.	moy.
Avant	44.3	17.7	33.3	34.3	69.2	39	39.6
Après	67.7	25	52.9	49.2	77.8	43.1	52.6

(c) DDSD

Tableau 1 – IoU obtenue par classe et en moyenne avant et après le processus interactif

Jeu de données	Pixels corrigés
ISPRS	7219
DDSD	13876
AIRS	4056

Tableau 2 – Nombre moyen de pixels corrigés par clic

DDSD. De plus, comme on peut le voir sur le Tableau 2, chaque clic permet de corriger environ 8000 pixels en moyenne.

Les écarts entre les différents résultats peuvent être expliqués par la nature des différents jeux de données. En effet, dans AIRS, le réseau peut initialement ne pas détecter les rares bâtiments d'une image ou, au contraire, prédire des bâtiments dans une image n'en contenant aucun. Quelques clics suffisent alors pour corriger précisément ces erreurs. Ainsi, l'IoU augmente considérablement sans modifier un grand nombre de pixels dans les cartes de prédiction. Ensuite, pour les jeux de données ISPRS et DDSD qui contiennent plus de classes qu'AIRS, DDSD est plus complexe à segmenter qu'ISPRS comme le montrent les résultats initiaux du Tableau 2. Les clics peuvent donc corriger de plus larges régions d'erreur et ainsi avoir plus d'impact sur l'IoU.

5.3 Influence de l'architecture du réseau

Nous comparons LinkNet à SegNet [4], UNet [28] et DeepLabv3 [10] qui sont des réseaux de segmentation standards de complexité croissante et également aux architectures plus légères suivantes : LEDNet [34], ERFNet [27] et D3Net [8]. La Figure 4 montre les résultats obtenus avec les différentes architectures sous les mêmes conditions d'apprentissage et d'évaluation. Comme attendu puisque ce processus est censé être indépendant de l'architecture du réseau, les gains sont du même ordre de grandeur. En effet, l'IoU initiale moyenne est de 68,8% avec un écart-type de 2.13 tandis que la moyenne du gain IoU est de 3.9% avec

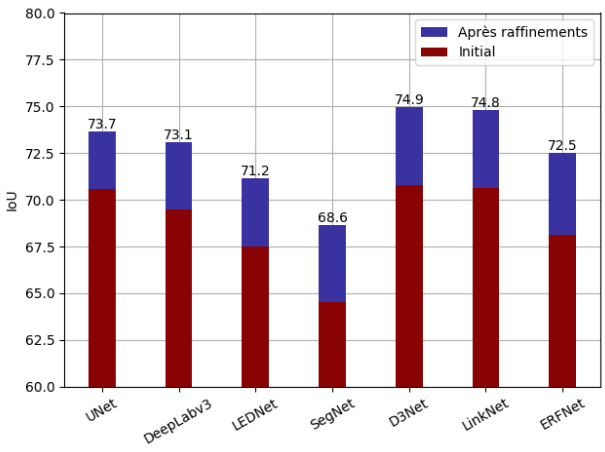


Figure 4 – Étude de l'impact du choix de l'architecture sur le jeu de données de Potsdam. Les résultats sont triés par gain d'IoU.

un écart-type de 0.4. La Figure 4 montre également que, sur un même jeu de données, le gain de précision apporté par la correction ne semble pas être corrélé à la précision de la carte de segmentation initiale. Par exemple, l'architecture présentement la moins précise initialement - SegNet - est dans la moyenne en ce qui concerne le gain d'IoU.

5.4 Analyse avec un utilisateur

Pour cette expérience, les images du jeu de données de validation de Potsdam ont été raffinées manuellement par un annotateur humain. Si le nombre de clics dépasse 120, nous le seuillons à 120 afin de faire une comparaison équitable avec le processus automatique.

Analyse locale. D'une part, comme le montre la Figure 5, les raffinements peuvent être très intuitifs et efficaces sur des domaines sémantiquement similaires à ceux observés lors de l'entraînement. En revanche, si la sémantique est nouvelle par rapport à ce qui se trouve dans le jeu d'apprentissage, les réseaux de neurones ont du mal à utiliser efficacement les annotations.

Par exemple, dans le jeu de données de Potsdam, il n'y a qu'un seul parking extérieur sur le toit d'un bâtiment. Ceci signifie qu'il y a un seul endroit avec la sémantique *voiture* entourée par *bâtiment* dans le jeu de données. Nous avons conservé l'image associée à ce parking dans le jeu de validation pour étudier l'impact des annotations dans ce scénario. La Figure 6 montre le résultat de notre approche sur ce parking. Puisque cette zone ressemble également à une route, qui est aussi une classe possible, il est initialement difficile pour le réseau de la segmenter correctement. Néanmoins, il réussit à y reconnaître les voitures garées. Puis, avec des annotations *bâtiment*, le réseau reconnaît avec succès un bâtiment. Cependant, il considère désormais que les véhicules qui y sont stationnés font partie du bâtiment car il n'a jamais vu la classe *voiture* entourée

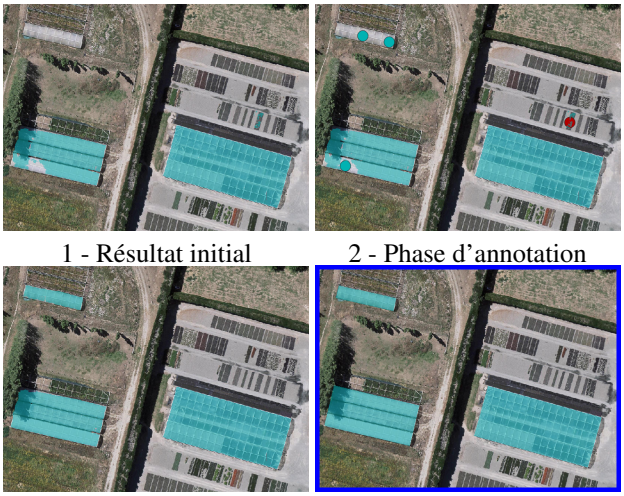


Figure 5 – Correction de segmentation de bâtiments. Trois points bleus *bâtiments* et un point rouge *non-bâtiment* représentent les interactions.

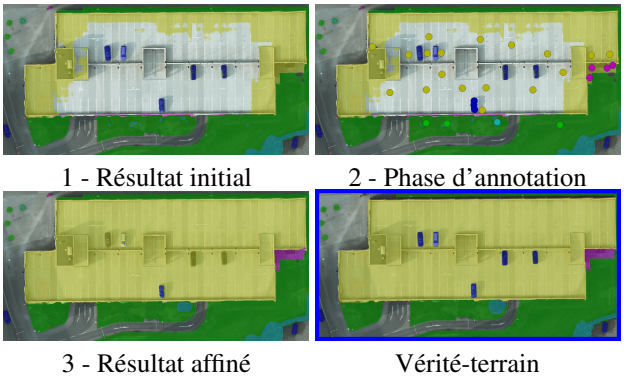


Figure 6 – Segmentation difficile d'un parking extérieur puisque le réseau n'a pas appris la sémantique *voiture sur bâtiment*. Seule la voiture en bas des images est annotée et reconnue comme telle.

de la classe *bâtiment* pendant l'entraînement. Comme nous pouvons le voir sur la Figure 6, des annotations supplémentaires *voiture* permettent au réseau de reconnaître la sémantique correcte de la scène. Cependant, le processus n'est alors pas fluide et intuitif car les voitures qui étaient principalement bien reconnues doivent néanmoins être annotées. Cet exemple montre donc que notre procédé ne fonctionne pas de manière optimale lorsqu'il est confronté à des zones avec une sémantique trop différente de celle présente dans le jeu de données d'entraînement.

Analyse globale. En ce qui concerne la distribution des clics, comme le montre la Figure 7, un opérateur humain a tendance à concentrer ses clics sur des zones spécifiques tandis que l'évaluation automatique répartit plutôt les annotations sur toute l'image. Cependant, comme le montre la Figure 8, ces clics groupés semblent augmenter efficacement la métrique. En effet, avec l'évaluation manuelle, 4

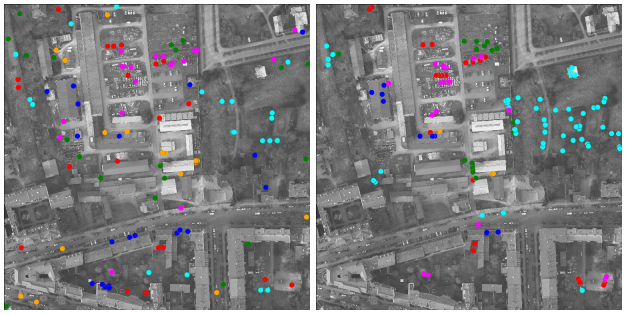


Figure 7 – Distribution des clics sur une image de Potsdam en évaluation automatique (gauche) et manuelle (droite). Les couleurs représentent les différentes classes.

classes sur 6 sont plus améliorées qu'en automatique et le gain d'IoU moyen est globalement meilleur. Cela montre l'efficacité de notre approche avec un utilisateur humain dans la boucle et suggère que les résultats automatiques sont une borne inférieure du gain potentiel.

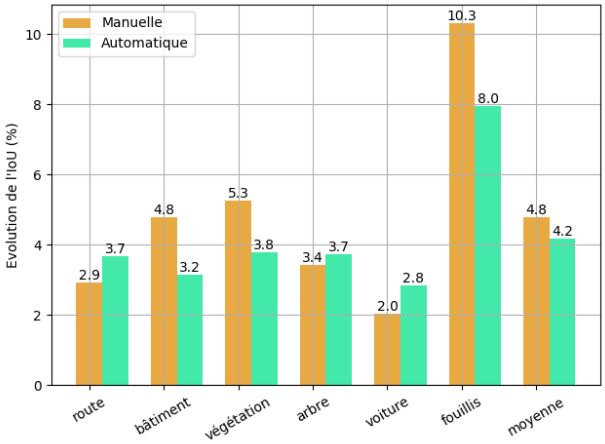


Figure 8 – Comparaison de l'évolution d'IoU entre les évaluations manuelles et automatiques sur le jeu de données de Potsdam.

6 Conclusion

Nous avons proposé dans cet article un procédé de segmentation multi-classe interactif pour les images aériennes. Partant d'un réseau de neurones conçu pour de la segmentation sémantique, il consiste à entraîner ce réseau à exploiter des annotations utilisateur. Au moment du test, les annotations utilisateur sont entrées dans le réseau de neurones sans changer les paramètres du modèle, ce qui rend le processus rapide et efficace. Grâce à des expériences sur trois jeux de données aériens publics, nous avons montré que le raffinement interactif est efficace pour toutes les classes. Il améliore les résultats de 4 % en moyenne pour 120 clics et produit des cartes de segmentation visuellement plus élégantes. Nous avons montré que notre processus interactif

est efficace quel que soit l'architecture de réseau de neurones sous-jacent. À l'avenir, nous étudierons un encodage des annotations adaptatif dépendant de la classe.

Références

[1] D. Acuna, H. Ling, A. Kar, and S. Fidler. Efficient interactive annotation of segmentation datasets with Polygon-RNN++. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 859–868. IEEE, 2018. 2

[2] E. Agustsson, J. R. Uijlings, and V. Ferrari. Interactive full image segmentation by considering all regions jointly. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11622–11631. IEEE, 2019. 2

[3] M. Andriluka, J. R. Uijlings, and V. Ferrari. Fluid annotation : a human-machine collaboration interface for full image annotation. In *MultiMedia (MM)*, pages 1957–1966. ACM, 2018. 2

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet : A deep convolutional encoder-decoder architecture for image segmentation. In *Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, volume 39, no. 12, pages 2481–2495. IEEE, 2017. 5

[5] R. Benenson, S. Popov, and V. Ferrari. Large-scale interactive object segmentation with human annotators. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11700–11709. IEEE, 2019. 2, 3

[6] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *International Conference on Computer Vision (ICCV)*, pages 105–112. IEEE, 2001. 2

[7] E. Carlinet and T. Géraud. Une approche morphologique de segmentation interactive avec l'arbre des formes couleur. In *Groupement de Recherche en Traitement du Signal et de l'Image (GRETSI)*, 2015. 2

[8] M. Carvalho, B. Le Saux, P. Trouvé-Peloux, A. Almansa, and F. Champagnat. On regression losses for deep depth estimation. In *International Conference on Image Processing (ICIP)*, pages 2915–2919. IEEE, 2018. 5

[9] A. Chaurasia and E. Culurciello. LinkNet : Exploiting encoder representations for efficient semantic segmentation. In *Visual Communications and Image Processing (VCIP)*. IEEE, 2017. 4

[10] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. In *arXiv preprint arXiv :1706.05587*, 2017. 5

[11] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*, pages 801–818. Springer, 2018. 1

[12] Q. Chen, L. Wang, Y. Wu, G. Wu, Z. Guo, and S. L. Waslander. Aerial imagery for roof segmentation : A large-scale dataset towards automatic mapping of buildings. 2018. 4

[13] L. Grady. Random walks for image segmentation. In *Trans. on Pattern Analysis & Machine Intelligence (TPAMI)*, volume 28, no. 11, pages 1768–1783. IEEE, 2006. 2

[14] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. In *International Conference on Computer Vision (ICCV)*, pages 2961–2969. IEEE, 2017. 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016. 4

[16] Y. Hu, A. Soltoggio, R. Lock, and S. Carter. A fully convolutional two-stream fusion network for interactive image segmentation. In *Neural Networks*, volume 109, pages 31–42. Elsevier, 2019. 2

[17] W.-D. Jang and C.-S. Kim. Interactive image segmentation via backpropagating refinement scheme. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5306. IEEE, 2019. 2

[18] J. Liew, Y. Wei, W. Xiong, S.-H. Ong, and J. Feng. Regional interactive image segmentation networks. In *International Conference on Computer Vision (ICCV)*, pages 2746–2754. IEEE, 2017. 2

[19] H. Ling, J. Gao, A. Kar, W. Chen, and S. Fidler. Fast interactive object annotation with Curve-GCN. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5257–5266. IEEE, 2019. 2

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440. IEEE, 2015. 2

[21] S. Mahadevan, P. Voigtlaender, and B. Leibe. Iteratively trained interactive segmentation. In *arXiv preprint arXiv :1805.04398*, 2018. 2

[22] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. Deep Extreme Cut : from extreme points to object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 616–625. IEEE, 2018. 2, 3

[23] B. Mathieu, A. Crouzil, and J.-B. Puel. Segmentation interactive pour l'annotation de photographies de paysages. In *Congres national sur la Reconnaissance des Formes et Intelligence Artificielle (RFIA)*, 2016. 2

[24] C. Nieuwenhuis and D. Cremers. Spatially varying color distributions for interactive multilabel segmentation. In *Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, volume 35, no. 5, pages 1234–1247. IEEE, 2012. 2

[25] C. Nieuwenhuis, S. Hawe, M. Kleinsteuber, and D. Cremers. Co-sparse textural similarity for interactive segmentation. In *European Conference on Com-*

puter Vision (ECCV), pages 285–301. Springer, 2014. 2

[26] QGIS Development Team. *QGIS Geographic Information System*. Open Source Geospatial Foundation, 2009. 4

[27] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo. ERFNet : Efficient residual factorized convnet for real-time semantic segmentation. In *Trans. on Intelligent Transportation Systems (ITS)*, volume 19, no. 1, pages 263–272. IEEE, 2017. 5

[28] O. Ronneberger, P. Fischer, and T. Brox. U-Net : Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015. 5

[29] C. Rother, V. Kolmogorov, and A. Blake. GrabCut : Interactive foreground extraction using iterated graph cuts. In *Trans. On Graphics (TOG)*, volume 23, no. 3, pages 309–314. ACM, 2004. 2

[30] F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf. The ISPRS benchmark on urban object classification and 3D building reconstruction. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS Annals)*, volume 1, no. 1, pages 293–298. Göttingen : Copernicus GmbH, 2012. 1, 4

[31] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In *International Conference on Computer Vision (ICCV)*, pages 1393–1400. IEEE, 2009. 2

[32] J. Santner, T. Pock, and H. Bischof. Interactive multi-label segmentation. In *Asian Conference on Computer Vision (ACCV)*, pages 397–410. Springer, 2010. 2

[33] J. Santner, M. Unger, T. Pock, C. Leistner, A. Saffari, and H. Bischof. Interactive texture segmentation using random forests and total variation. In *British Machine Vision Conference (BMVC)*, pages 1–12. BMVA, 2009. 2

[34] Y. Wang, Q. Zhou, J. Liu, J. Xiong, G. Gao, X. Wu, and L. J. Latecki. LEDNet : A lightweight encoder-decoder network for real-time semantic segmentation. In *arXiv preprint arXiv :1905.02423*, 2019. 5

[35] Z. Wang, D. Acuna, H. Ling, A. Kar, and S. Fidler. Object instance annotation with deep extreme level set evolution. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7500–7508. IEEE, 2019. 2

[36] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 373–381. IEEE, 2016. 2, 3