



**HAL**  
open science

# ZGP : une alternative aux réseaux de neurones pour la segmentation sémantique de nuages dans les images satellites multi-spectrales

I Grenet, Y Bobichon, A Girard, F Férésin

## ► To cite this version:

I Grenet, Y Bobichon, A Girard, F Férésin. ZGP : une alternative aux réseaux de neurones pour la segmentation sémantique de nuages dans les images satellites multi-spectrales. Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle ( APIA2021 ), Jul 2021, Bordeaux, France. hal-03874240

**HAL Id: hal-03874240**

**<https://hal.science/hal-03874240>**

Submitted on 27 Nov 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# ZGP : une alternative aux réseaux de neurones pour la segmentation sémantique de nuages dans les images satellites multi-spectrales

I. Grenet<sup>1,2</sup>, Y. Bobichon<sup>1</sup>, A. Girard<sup>1</sup>, F. Férésin<sup>1</sup>,

<sup>1</sup> Institut de Recherche Technologique Saint Exupéry - Sophia Antipolis, France

<sup>2</sup> MyDataModels - Sophia Antipolis, France

ig@mydatamodels.com

{yves.bobichon, adrien.girard, frederic.feresin}@irt-saintexupery.com

## Résumé

*Cet article présente les résultats de l'algorithme évolutionnaire ZGP appliqué à la segmentation de nuages dans les images satellites multi-spectrales. De nombreuses méthodes ont été développées pour automatiser la détection de nuages mais peuvent parfois s'avérer difficiles à implémenter à bord des satellites. Ici nous comparons les performances de ZGP à celles d'un réseau de neurones (UNET). Les résultats obtenus sur des images du satellite Landsat-8 et les avantages de ZGP en font une bonne alternative aux réseaux de neurones pour le déploiement d'applications de segmentation d'images sur des systèmes embarqués.*

## Mots-clés

*Segmentation de nuages, images satellites multi-spectrales, algorithmes évolutionnaires, ZGP, UNET*

## Abstract

*This paper presents the results of the evolutionary algorithm ZGP applied to cloud segmentation in remote sensing multispectral images. Many methods have been developed in order to automate cloud detection but can sometimes be difficult to implement on-board satellites. Here we compare ZGP to a deep learning algorithm (UNET). The results obtained on the Landsat-8 satellite images and the advantages of ZGP make it a good alternative to neural networks for the deployment of image segmentation applications on embedded systems.*

## Keywords

*Cloud segmentation, multispectral remote sensing images, evolutionary algorithms, ZGP, UNET*

## 1 Introduction

La segmentation des nuages dans les images satellites est un cas d'utilisation bien connu de l'imagerie spatiale. On estime en effet que les nuages recouvrent plus de 54% des terres émergées et 68% de la surface des océans [21] réduisant d'autant l'utilité des images acquises par les satellites d'observation de la Terre pour des utilisateurs généralement

intéressés par des phénomènes se produisant à la surface du globe. Pour faire face à la quantité toujours croissante d'images fournies quotidiennement par les systèmes d'observation de la Terre, de nombreuses méthodes ont été développées ces dernières années pour automatiser la tâche de détection des nuages dans les segments sol des satellites [7, 10, 19, 35]. Ces méthodes ont été testées et validées sur de nombreuses bases de données labellisées dont certaines, comme SPARCS [25] et BIOME [15] développées pour le segment sol du satellite Landsat-8, tendent à s'imposer comme des références sur lesquelles de nombreux algorithmes de détection de nuages ont été entraînés, validés et comparés.

Dans une image satellite les nuages se présentent sous la forme de régions à forte radiométrie parfois délimitées par des contours diffus. De tailles très variables et sans forme géométrique caractéristique, leurs positions changent au cours du temps. Ainsi les méthodes de détection de changements ont été les premières à être utilisées pour détecter efficacement des nuages [17, 36]. Cependant ces approches temporelles sont incompatibles avec la détection en temps réel des nuages sur des acquisitions individuelles pour lesquelles d'autres approches (SVM, arbres de décision, forêts aléatoires) ont été proposées [34]. L'objectif n'est pas ici de fournir un état de l'art exhaustif des méthodes de détection de nuages mais de se concentrer sur quelques-unes reconnues comme étant les plus performantes dans le but de définir une référence de performances pour notre algorithme génétique.

Récemment les réseaux de neurones convolutifs tels que UNET [30], FCN[25], SegNet [2] ou DeepLabV3 [6] ont fait progresser notablement les performances des méthodes classiques de segmentation sémantique d'images dans de nombreux domaines d'applications. Pour la détection de nuages dans les images satellites, les architectures de type UNET comme Rs-Net [20], CS-CNN [9] ou Cloud-Net [26] se sont montrées particulièrement performantes.

Les réseaux de neurones sont connus pour être généralement complexes à mettre en œuvre dans des systèmes embarqués tels que des satellites, dont les capacités de calculs

et de stockage mémoire sont limitées. Dans cette optique, plusieurs approches ont été proposées afin de simplifier les architectures de réseaux de neurones en vue de leur implémentation à bord de satellites pour la détection de nuages en temps réel ou la compression d'image sélective. MobU-Net [33] a été développé pour détecter les nuages dans des images à basse résolution compressées en Jpeg2000-à bord d'un CubeSat ARTSU. Bahl *et al.* [3] proposent une version compacte C-UNET pour la détection de nuages. Feresin *et al.* [14] ont démontré la faisabilité d'une implémentation sur FPGA d'un réseau de neurones compact pour la détection de nuages dans les images du satellite OPS-SAT. Plus récemment, un réseau de neurones pour la détection de nuages dans des images hyperspectrales générées par la camera HySCOUT-2 du nano-sat Phi-Sat a été implémenté dans un processeur ARM.

Toutes ces récentes applications montrent l'intérêt de développer des approches de segmentation d'image adaptées aux contraintes des systèmes embarqués, notamment des satellites, tout en conservant un niveau élevé de performance compatible avec les exigences de fiabilité, de robustesse et de faible complexité algorithmique requises par ce type de systèmes.

Dans cet article nous présentons une application d'un algorithme évolutionnaire propriétaire, le Zoetrope Genetic Programming (ZGP), à la segmentation sémantique. Nous montrons la performance atteinte par cet algorithme dans le cas de la détection de nuages dans les images satellite multi-spectrales. Les résultats sont comparés à ceux obtenus avec les méthodes de deep learning qui s'avèrent être les plus performantes actuellement, basé sur l'état de l'art des réseaux de neurones, le UNET. La section 2 introduit des généralités sur les algorithmes évolutionnaires et présente en particulier l'algorithme ZGP. La section 3 décrit les données et méthodes utilisées dans ces travaux. La section 4 détaille les résultats d'inférence des modèles sur les données SPARCS et BIOME. La section 5 présente les conclusions et les perspectives pour de futurs travaux.

## 2 Les Algorithmes Evolutionnaires

### 2.1 Principe

Les algorithmes évolutionnaires (AE) sont un type d'algorithmes utilisés en intelligence artificielle inspirés des mécanismes biologiques d'évolution tels que la reproduction, la mutation, la recombinaison et la sélection naturelle [11]. Ils permettent de trouver des solutions à des problèmes particulièrement difficiles à optimiser via la génération de nombreuses solutions candidates (les « individus »), au sein d'une population. Leur qualité est évaluée grâce à une fonction d'adaptation aux données (*fitness function*). La population évolue ensuite au cours des générations grâce à l'application d'opérateurs génétiques (mutation, recombinaison) aux individus préalablement sélectionnés, appelés parents, ce qui permet la création de nouveaux individus, les descendants. Les AE sont des algorithmes stochastiques puisque leur apprentissage implique de nombreux processus aléatoires. Le processus général d'un AE, illustré en Figure 1,

est le suivant [8] :

- Initialisation d'une population : création d'un ensemble aléatoire d'individus correspondant aux solutions candidates. Ces individus peuvent être représentés de différentes façons (alphabet fini, valeurs binaires, vecteurs de valeurs réelles, arbres, programmes, etc). La population possède un nombre fini d'individus qui est le plus souvent constant.
- Evaluation : mesure de la qualité de chaque individu par rapport à la meilleure solution. Cette mesure est calculée grâce à une fonction mathématique appelée fonction d'évaluation (ou *fitness function*) et reflète à quel point l'individu correspond à la solution.
- Opérations génétiques : génération de nouveaux individus grâce à deux types d'opérateurs :
  - la mutation : elle applique une transformation aléatoire de certaines caractéristiques d'un individu, indépendamment des autres. La mutation crée de la diversité et permet ainsi l'exploration de l'espace des solutions.
  - la recombinaison (*cross-over*) : mime la reproduction biologique en échangeant de l'information entre deux parents de façon aléatoire, générant ainsi un nouvel individu. La recombinaison a pour but de créer des descendants qui portent les « bonnes » caractéristiques des parents et permet ainsi l'exploitation.
- Sélection : sélection d'individus à partir de l'ensemble des parents et descendants en fonction de leur qualité (*fitness*). Le nouvel ensemble constitue la nouvelle génération.

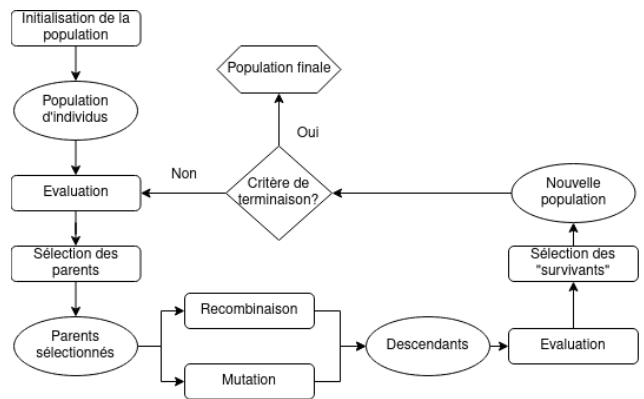


FIGURE 1 – Principe général d'un algorithme évolutionnaire

L'ensemble du processus, depuis la génération d'individus jusqu'à la sélection, est répété un certain nombre d'itérations jusqu'à atteindre une condition de terminaison (*e.g.* : la *fitness* ne s'améliore plus depuis un certain nombre de générations, une limite de temps de calcul est atteinte, etc). Les AE ont plusieurs avantages parmi lesquels leur bonne adaptabilité à une large variété de problèmes. En effet, puisque aucune hypothèse n'est faite sur la distribution des données et la fonction sous-jacente liée à celles-ci,

les AE sont libres d'apprendre n'importe quelle forme de fonction et peuvent adopter différentes stratégies pour trouver de meilleures solutions [16]. Ceci n'est pas le cas d'autres techniques d'optimisation. De plus, ils permettent de trouver des solutions à des problèmes qui ne peuvent être résolus par d'autres types d'algorithmes, en terme de difficulté de calcul, indépendamment de la taille des données. Aussi, contrairement aux réseaux de neurones, les AE sont capables de gérer des fonctions discontinues ou non-différentiables [24].

## 2.2 Applications

Les AE ont été utilisés dans divers domaines et pour une grande variété de problèmes, allant des sciences naturelles et de l'informatique à l'industrie et au management, en passant par la finance, l'économie ou encore les sciences sociales.

Concernant la vision par ordinateur, les AE ont été appliqués avec succès à différentes tâches telles que la classification d'image, la reconnaissance faciale, l'extraction de caractéristiques, la reconnaissance d'objets ou encore la segmentation d'images. En particulier, on distingue deux types d'applications pour la segmentation d'images [13]. La première vise à améliorer les techniques de segmentation existantes en recherchant leurs meilleurs paramètres. La seconde est quant à elle une réelle classification individuelle de chaque pixel. Ces deux types de segmentation utilisant des AE ont surtout été appliquées à des images médicales [4, 22, 12, 31] mais certaines ont aussi été utilisées pour les images satellites. Par exemple, Yiqiang *et al.* ont proposé le « *Chaos Genetic Algorithm* » qui vise à optimiser les clusters initiaux pour la classification non supervisée de pixels d'images Landsat-5 de la ville de Huainan en Chine [32]. Sur le même principe, Awad a développé une méthode non supervisée basée sur un algorithme génétique multi-objectif afin de trouver les pixels à utiliser à l'initialisation d'une méthode de clustering. Pour cela, l'algorithme génétique maximise à la fois le nombre de pixels dans chaque cluster ainsi que leur homogénéité, en fonction de leur signature spectrale [1]. De plus, Mylonas *et al.* ont développé l'algorithme GenESIS qui combine les résultats de classification de pixels faite par un SVM avec un algorithme génétique [29]. Les auteurs ont récemment amélioré leur méthode en proposant une version basée sur la recherche locale [28]. Enfin, Hinojosa *et al.* ont également utilisé un algorithme multi-objectif pour la segmentation d'images satellites multi-spectrales en recherchant les meilleurs seuils de valeurs spectrales pour classifier les pixels [18].

L'ensemble de ces travaux permettent la segmentation par l'utilisation d'AE associés à d'autres méthodes. Dans ce travail, nous montrons qu'un AE, utilisé seul, est capable de générer des modèles de segmentation d'images satellites basée uniquement sur l'information spectrale des pixels avec une bonne précision.

## 2.3 Zoetrope Genetic Programming

Nous utilisons ici un algorithme évolutionnaire propriétaire et original appelé, Zoetrope Genetic Programming (ZGP) [5]. Cet algorithme est au coeur de l'application web de machine learning automatique appelée TADA, développée par la startup française MyDataModels<sup>1</sup>. ZGP est considéré comme un algorithme de régression symbolique, dans lequel les individus de la population pourraient être assimilés à des arbres d'expressions. En effet, même si la façon de générer les individus est unique et non standard, au final ceux-ci sont représentés par une combinaison mathématique de constantes et de variables qui évolue durant la phase d'apprentissage afin de trouver celle qui s'adapte le mieux aux données. Plus précisément, le mécanisme de construction des formules est le suivant : pour commencer,  $m_e$  éléments ( $E_1, \dots, E_{m_e}$ ) sont sélectionnés aléatoirement parmi les variables d'entrée (avec une probabilité de 90%) ou des constantes aléatoires (avec une probabilité de 10%). Ces éléments subissent ensuite  $m_m$  étapes de maturation qui consistent à appliquer l'opération de fusion représentée par l'équation (1) aux couples d'éléments  $E_i$  et  $E_j$  :

$$f(E_i, E_j) = r \cdot \text{op}_1(E_i, E_j) + (1 - r) \cdot \text{op}_2(E_i, E_j) \quad (1)$$

Où  $\text{op}_i$ ,  $i = 1, 2$  sont des opérateurs mathématiques uniformément choisis dans un ensemble prédéfini  $\mathcal{O}$ , et  $r = U[0, 1]$ ; le résultat de  $f(E_i, E_j)$  remplace soit  $E_i$  soit  $E_j$ . A la fin des étapes de maturation, les éléments matures sont appelés "zootropes" et sont donc le résultat de plusieurs opérations de fusions, représentés par une formule mathématique composée de variables et constantes. Ils sont ensuite combinés et pondérés de façon linéaire grâce à une fonction mathématique qui vise à minimiser les erreurs de prédiction sur une partie des données. Cette fonction diffère avec la tâche apprise par l'algorithme (régression, classification binaire ou multi-classes).

Finalement, bien que les zootropes aient été construits de façon naïve par un processus de sélection aléatoire de variables, ceux sélectionnés par le modèle final sont ceux qui possèdent les variables explicatives les plus pertinentes. Ensuite, l'algorithme de programmation génétique considère les zootropes en tant que modèles et les fait évoluer grâce aux opérateurs génétiques. Là encore, ZGP possède des méthodes de mutation et de recombinaison non standards. En effet, lors de la mutation, l'algorithme sélectionne des couples de modèles et remplace le plus mauvais par un "mutant" du meilleur (évalués grâce à la *fitness function*). Durant la recombinaison, l'algorithme sélectionne le meilleur et le plus mauvais modèle parmi un ensemble de modèles; il propage ensuite un certain nombre d'éléments et de fusions du meilleur modèle vers le plus mauvais. La mutation et la recombinaison sont répétées plusieurs fois au cours d'une même génération. A la fin de chaque génération, tous les modèles sont évalués grâce à la *fitness function* en utilisant un ensemble de données de validation et le meilleur modèle est gardé en mémoire. Une nouvelle

1. [www.mydatamodels.com](http://www.mydatamodels.com)

génération débute alors à partir de tous les modèles obtenus à la précédente génération. Au terme de l'apprentissage, le meilleur modèle, toutes générations confondues, sera sélectionné comme modèle final.

À l'issue de la phase d'apprentissage, l'algorithme sélectionne finalement le sous-ensemble de variables les plus pertinentes pour expliquer le phénomène considéré. Cela se traduit par une expression mathématique relativement simple qui permet de prédire (régression) ou de classer (classification) de nouvelles données. Grâce à son format compréhensible par l'utilisateur, cette formule est facilement interprétable et permet d'identifier les variables les plus importantes pour prédire le phénomène sous-jacent. De plus, ce type d'expression peut être programmé dans différents langages (Python, JavaScript, C++) et être aisément implémenté sur des microcontrôleurs dans des systèmes embarqués. En effet, le modèle généré possède une très faible taille mémoire (de l'ordre de quelques kilo-octets).

Par rapport aux autres types d'AE, ZGP possède plusieurs avantages. D'abord, la façon spécifique de représenter les individus évite l'effet de congestion, c'est-à-dire la construction de longues formules, alors que la plupart des AE doivent d'abord être paramétrés pour cela. En effet, alors que pour les autres AE, des arbres à expressions finis sont générés aléatoirement, ZGP construit d'abord des formules aléatoires de façon naïve correspondant aux branches d'un arbre de taille finie et seulement les plus informatives sont gardées pour donner l'arbre final. Cela permet une recherche plus rapide des solutions. De plus, la formule est ainsi plus facilement interprétable et compréhensible par un expert métier puisque les dépendances entre les variables explicatives et la variable d'intérêt sont explicites et peu nombreuses. Aussi, tel un générateur de variables, ZGP a une capacité intrinsèque à faire de la "feature engineering", contrairement aux algorithmes classiques de machine learning pour lesquels ce travail doit être fait en amont durant la phase de préparation des données (pré-processing). Cette capacité est commune aux algorithmes de régression symbolique.

## 3 Méthode

### 3.1 Données

Dans cette étude nous avons considéré deux jeux de données labélisées, SPARCS et BIOME, par ailleurs déjà largement utilisés dans des travaux similaires, notamment [20], pour établir la performance d'algorithmes de détection de nuages. Ces données sont constituées d'images multi-spectrales Landsat-8 avec 10 bandes spectrales allant du spectre visible (bandes de l'instrument OLI) à l'infra-rouge thermique (Bandes TIRS) (Table 1).

La base de données SPARCS [25] contient 80 images de 1000x1000 pixels calibrées radiométriquement dont les valeurs représentent pour chaque bande spectrale la réflectance codée sur une échelle de 16-bits. Les images sont labélisées et fournies avec un masque pour 7 classes de pixels : *cloud shadow*, *cloud*, *shadow over water*, *water*, *snow*, *land*, *flooded*.

Bande	Longueur d'onde ( $\mu\text{m}$ )
Coastal	0.43-0.45
Blue	0.45-0.51
Green	0.53-0.59
Red	0.64-0.67
NIR	0.85-0.88
SWIR1	1.57-1.65
SWIR2	2.11-2.29
Cirrus	1.36-1.38
TIRS1	10.6-11.19
TIRS2	11.50-12.51

TABLE 1 – Bandes spectrales des images du satellite Landsat-8 utilisées dans les jeux de données SPARCS et BIOME

La base de données BIOME [15] est constituée de 96 images Landsat-8 réparties en 8 types de terrains différents appelés biomes : *barren*, *forest*, *grass crops*, *shrubland*, *urban*, *water*, and *wetlands*. Un masque de pixels est fourni pour les 4 classes suivantes : *cloud shadow*, *clear*, *thin cloud*, *cloud*. Comme pour les données SPARCS, la radiométrie des images BIOME représente la réflectance spectrale sur une échelle de 16-bits.

Cette étude se focalise sur la détection de nuages et la distinction entre nuages et neige. Dans cet objectif, tous les pixels ne représentant pas des nuages sont regroupés dans une même méta-classe *no cloud*, sauf lors de l'utilisation de modèles multi-classes pour lesquels les pixels labélisés *snow* sont considérés séparément des pixels non nuageux dans une classe appelée *snow*.

Les données SPARCS ont été choisies pour l'entraînement des modèles en raison notamment de la présence d'annotations de pixels de la classe *snow* qui permet de réaliser l'entraînement de modèles sur des pixels de neige pour les modèles ZGP multi-binaire et multi-classes. Les images BIOME sont utilisées comme images de test pour évaluer et comparer la capacité de généralisation des différents modèles sur de nouvelles images jamais vues lors de l'entraînement.

### 3.2 Algorithme de référence

Le modèle utilisé pour établir les performances de référence est un UNET tel que décrit dans [30] et représenté Figure 2. Ce type d'architecture n'est pas parmi les plus récentes développées pour la segmentation (SegNet [2] ou DeepLabV3 [6] sont par exemple plus récentes). Cependant, UNET a été sélectionné ici comme méthode de référence car cette approche a été très utilisée pour la segmentation de nuages dans les images satellite dans plusieurs applications récentes [9, 26, 20], notamment sur les bases de données SPARCS et BIOME qui font référence dans ce domaine de la télédétection.

L'architecture d'un réseau UNET standard est constituée d'un encodeur (partie gauche) et d'un décodeur (partie droite). L'encodeur est un réseau de neurones convolutif formé d'une alternance de couches de convolution

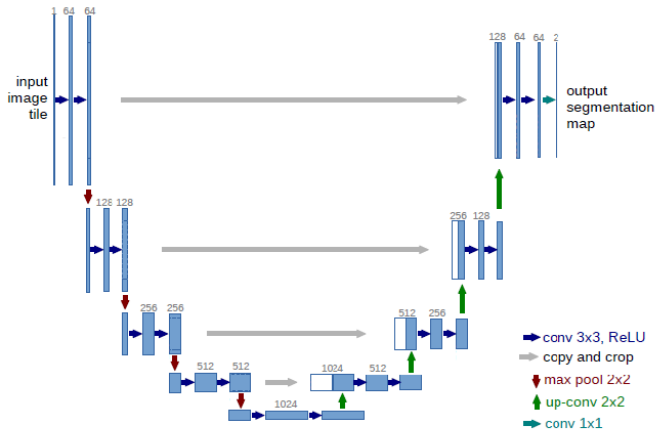


FIGURE 2 – Architecture UNET d’après [30]

par des filtres 3x3 entraînaibles, des fonctions d’activation ReLU et des couches de max-pooling 2x2 assurant le sous échantillonnage d’un facteur 2 à chaque niveau. Le nombre de filtres de convolution de la couche d’entrée est de 64 et double à chaque niveau pour atteindre 512 au dernier niveau. Chaque étape du décodeur réalise le sur-échantillonnage d’un facteur 2 des couches précédentes dont le nombre est réduit d’un facteur 2 à la sortie de chaque niveau, la concaténation avec les couches de l’encodeur correspondant au même niveau, et la convolution par des filtres 2x2 (noté up-conv sur la Figure 2) suivie par des fonctions d’activation ReLU. A la sortie du réseau, une convolution 1x1 est utilisée pour projeter chacune des composantes du tenseur sur le nombre de classes. Une fonction softmax détermine ensuite la classe des pixels. Pour éviter le sur-apprentissage, un dropout de 50% a été ajouté en sortie de l’encodeur et une *batch normalization* (non représentée sur la Figure 2) après chaque couche de convolution comme pour le Res-Net dans [20].

L’apprentissage a été réalisé sur des patches de taille 256x256 extraits des données SPARCS après division par 65 535 de chaque bande spectrale de manière à réduire la dynamique des entrées dans l’intervalle [0,1]. Les paramètres d’apprentissage utilisés sont les suivants :

- Learning rate :  $0.5 * 10^{-3}$
- Taille des sous-ensembles : 90% apprentissage (soit 72 images SPARCS) et 10% validation (8 images SPARCS)
- 150 époques d’apprentissage, une validation par époque

Les données ont été traitées par batches de 96 patches 256x256 représentant plus de 75 millions d’échantillons d’entraînement (*i.e.* pixels spectraux). La taille de ces batches a été optimisée en fonction du nombre et de la capacité mémoire des 6 GPU Geforce GTX 1080 utilisés en parallèle lors de l’apprentissage. Il en résulte, malgré la complexité d’UNET, une durée d’apprentissage relativement faible d’environ 90 minutes pour 150 époques sur la

totalité des 10 bandes spectrales et des 80 images de la base de données SPARCS.

Le modèle de référence a été entraîné sur deux combinaisons spectrales différentes. Une première combinaison multi-spectrale (MS) comprenant la totalité des 10 bandes spectrales disponibles dans les données Landsat-8 et une combinaison avec uniquement les bandes RGB classiquement utilisées sur des nano-satellites comme dans l’expérience OPS-SAT [14]. Ces mêmes combinaisons de bandes ont également été utilisées pour entraîner les modèles ZGP MS et RGB décrits ci-après.

### 3.3 Algorithme ZGP

Dans ces travaux nous entraînons ZGP pour générer des modèles de classification permettant de prédire la classe de chaque pixel d’une image à partir de leur information spectrale uniquement. Plus précisément, nous entraînons 3 modèles basés sur les bandes spectrales RGB (modèles RGB) et 3 modèles basés sur les 10 bandes spectrales des capteurs OLI et TIR du satellite Landsat-8 (modèles multi-spectraux ou MS). Dans les deux cas, les 3 modèles sont les suivants :

- un modèle binaire (*cloud, no cloud*)
- un modèle multi-binaire : le modèle binaire (*cloud, no cloud*) précédent suivi d’un second modèle binaire (*cloud, snow*) appliqué uniquement sur les pixels prédits comme *cloud* par le premier modèle
- un modèle multi-classes (*cloud, land, snow*) dont la classe *land* contient tous les pixels *no cloud* excepté ceux de la classe *snow*.

Pour l’apprentissage, ZGP utilise uniquement 10 000 observations. Chaque observation correspond à un vecteur de pixels spectraux avec respectivement 3 ou 10 variables pour les modèles RGB ou MS. Les 10 000 pixels sont choisis aléatoirement parmi les 80 scènes de la base de données SPARCS de telle sorte que le jeu de données final soit équilibré (nombre égal de pixel dans chaque classe). Les variables d’entrée sont les valeurs des pixels sur 16 bits entre 0 et 65 535. Concernant la variable de sortie (la classe), nous considérons les méta-classes *no cloud, cloud* et *snow* définies précédemment. Pour l’apprentissage, ZGP divise le jeu de données de façon automatique et aléatoire en trois sous-ensembles : un ensemble d’apprentissage, un ensemble de validation et un ensemble de test. Les paramètres utilisés pour entraîner l’algorithme évolutionnaire sont les suivants :

- taille des sous-ensembles : 40% apprentissage, 30% validation, 30% test
- taille de la population : 500 individus
- nombre de générations : 100
- critère de terminaison : fin des générations

### 3.4 Métriques

Les performances sont évaluées de manière globale sur un ensemble de données par les métriques standard suivantes.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Rappel = \frac{TP}{TP + FN} \quad (3)$$

$$F - score = \frac{2 \cdot precision \cdot rappel}{precision + rappel} \quad (4)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (5)$$

Quel que soit le modèle, les métriques sont calculées pour la classe *cloud*. Ainsi, les vrais positifs (TP) correspondent au nombre de pixels *cloud* correctement identifiés en tant que nuage alors que les vrais négatifs (TN) correspondent au nombre de pixels *no cloud* correctement identifiés en tant que tels. Les faux positifs (FP) sont le nombre de pixels *no cloud* mal classifiés en *cloud* et inversement pour les faux négatifs (FN). Par définition, l'*accuracy* globale mesure le pourcentage de bonnes prédictions pour toutes les classes. Pour comparer les différents modèles entre eux, on s'intéressera essentiellement à la valeur du F-score qui représente la moyenne de la précision et du rappel pour une classe donnée et permet ainsi de rendre compte à la fois de la complétude des prédictions (rappel) et de leur exactitude (précision).

## 4 Résultats

### 4.1 Modèles générés par ZGP

Pour chacun des 6 modèles entraînés, ZGP sélectionne automatiquement les bandes spectrales les plus pertinentes pour la classification à effectuer et n'utilise pas forcément toutes les bandes disponibles, comme illustré dans la Table 2.

Modèle	Classes	Bandes utilisées
RGB binaire	<i>cloud</i> / <i>no cloud</i>	Red, Blue
RGB multi-binaire	<i>cloud</i> / <i>no cloud</i> + <i>cloud</i> / <i>snow</i>	Red, Blue + Red, Green
RGB multi-classes	<i>cloud</i> / <i>land</i> / <i>snow</i>	Red, Green, Blue
MS binaire	<i>cloud</i> / <i>no cloud</i>	Coastal, Blue, SWIR1, TIRS2
MS multi-binaire	<i>cloud</i> / <i>no cloud</i> + <i>cloud</i> / <i>snow</i>	Coastal, Blue, SWIR1, TIRS2 + Coastal, Red, NIR, SWIR2, TIRS2
MS multi-classes	<i>cloud</i> / <i>land</i> / <i>snow</i>	Coastal, Blue, Red, SWIR2, TIRS1, TIRS2

TABLE 2 – Résumé des 6 modèles ZGP avec les bandes sélectionnées par l'algorithme

La formule du modèle MS binaire générée par l'algorithme est donnée à titre d'exemple en Figure 3. On voit que pour ce modèle, ZGP n'a retenu que 4 bandes, ce qui permet une réduction de plus d'un facteur 2 du nombre de bandes à traiter par rapport au modèle de référence UNET MS qui utilise les 10 bandes. Les valeurs des deux formules correspondant aux deux classes (*cloud*, *no cloud*) sont calculées pour chaque pixel. Ensuite, la classe correspondant à la plus grande valeur est attribuée au pixel.

CLOUD :

$$-0.339 \times TIRS2 \times SWIR1 + 0.339 \times SWIR1 \times Coastal + 0.433 \times SWIR1 \times |Coastal| + 0.227 \times [0.439 \times (-TIRS2 + Coastal) + 0.5601 \times |Coastal|]$$

NO CLOUD :

$$0.855 \times Blue - 0.855 \times Coastal + 0.145 \times Blue^2$$

FIGURE 3 – Formule du modèle binaire ZGP multispectral. [...] correspond à l'arrondi à l'entier inférieur.

En terme de temps de calcul, l'entraînement de ZGP est de respectivement 5 et 15 minutes pour générer les modèles binaires et multi-classes présentés ici.

### 4.2 Performance des modèles

Une première évaluation des performances des différents modèles ZGP a été réalisée sur les données SPARCS. Seulement 1% des données SPARCS sont utilisées lors de l'apprentissage des modèles ZGP contre 90% pour celui des modèles UNET. Ainsi lors de l'inférence sur l'ensemble des données SPARCS, les modèles ZGP sont en situation de généralisation pour 99% des pixels contre seulement 10% pour les UNET. Les conditions d'inférence sur les données SPARCS sont donc plutôt favorables aux modèles UNET. Cependant cette première évaluation permet de comparer les modèles ZGP entre eux afin de mettre en évidence notamment l'apport des modèles MS par rapport aux modèles RGB.

Les données BIOME sont ensuite utilisées comme données de test afin de pouvoir comparer les performances de généralisation des différents modèles dans des conditions identiques (*i.e.* nouvelles données pour tous les modèles).

La Figure 4 indique les valeurs de F-score mesurées sur les bases de données SPARCS (barres pleines) et BIOME (barres hachurées) des modèles ZGP (en vert) et UNET (en bleu) pour les bandes RGB. La Figure 5 fourni les résultats de F-Score pour les modèles MS.

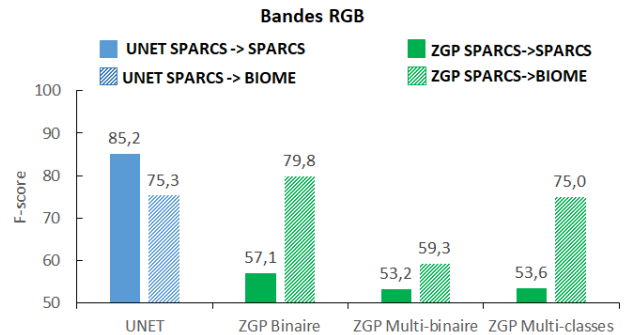


FIGURE 4 – F-score des modèles UNET et ZGP RGB entraînés sur SPARCS et inférés sur SPARCS et BIOME

On observe sur la Figure 4 que les modèles RGB ZGP inférés sur SPARCS sont moins performants que ceux issus de UNET mais que cette tendance s'inverse sur les données BIOME.

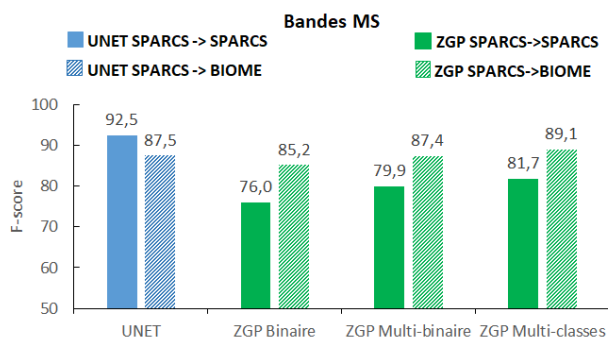


FIGURE 5 – F-score des modèles UNET et ZGP MS entraînés sur SPARCS et inférés sur SPARCS et BIOME

On note une tendance similaire en multi-spectral sur la Figure 5 pour laquelle le F-score UNET MS sur BIOME est seulement de 87,5% alors que celui du modèle ZGP multi-classes MS est de 89,1% (meilleur F-score de tous les modèles UNET et ZGP toutes bandes confondues sur la base de données de test BIOME).

Les Table 3 et Table 4 fournissent le détail des métriques mesurées sur l'ensemble des données SPARCS et BIOME pour les différents modèles UNET et ZGP. Les valeurs en gras indiquent le modèle qui obtient la meilleure métrique.

Métriques sur SPARCS des modèles entraînés sur SPARCS					
Bandes	Modèles	Accuracy	Précision	Rappel	F-score
RGB	UNET	94.86	<b>96.19</b>	76.5	85.22
	ZGP Binaire	76.03	43.69	85.22	57.06
	ZGP multi-binaire	83.43	58.73	48.66	53.22
	ZGP multi-classes	85.22	68.44	44.02	53.57
MS	UNET	<b>97.12</b>	93.77	<b>91.2</b>	<b>92.46</b>
	ZGP Binaire	89.16	66.53	88.68	76.02
	ZGP multi-binaire	91.52	73.88	86.97	79.89
	ZGP multi-classes	93.71	93.4	72.67	81.74

TABLE 3 – Métriques de modèles UNET et ZGP mesurées sur les données SPARCS

Métriques sur BIOME des modèles entraînés sur SPARCS					
Bandes	Modèles	Accuracy	Précision	Rappel	F-score
RGB	UNET	79.98	<b>90.88</b>	64.26	75.29
	ZGP Binaire	77.61	69.83	92.96	79.75
	ZGP multi-binaire	67.04	71.63	50.54	59.26
	ZGP multi-classes	77.53	79.38	71.12	75.02
MS	UNET	87.52	83.27	92.22	87.51
	ZGP Binaire	83.67	74.67	<b>99.25</b>	85.22
	ZGP multi-binaire	86.79	79.91	96.4	87.38
	ZGP multi-classes	<b>89.26</b>	85.92	92.54	<b>89.1</b>

TABLE 4 – Métriques de modèles UNET et ZGP mesurées sur les données BIOME

Concernant l'utilisation des modèles, sur un CPU Xeon multi-coeurs Silver 4114 x86 64 bits à 2.20GHz, le débit d'inférence des modèles ZGP binaires est de 24 Mpixels/s en RGB et 45 Mpixels/s en multi-spectral. Ces débits sont à comparer avec ceux des modèles binaires UNET qui sont de 2,3 Mpixel/s en MS et 2,7 Mpixels/s en RGB. A noter que le débit d'inférence des modèles ZGP dépend essentielle-

ment du nombre et de la complexité des opérateurs mathématiques à appliquer entre les bandes et peu du nombre de bandes spectrales. Ainsi pour effectuer une séparation pertinente des classes, ZGP conduit à un modèle RGB avec plus d'opérations et des opérateurs plus complexes qu'en MS où les bandes sont intrinsèquement plus discriminantes. C'est l'inverse avec les modèles UNET dont le débit d'inférence diminue logiquement lorsque le nombre de bandes à traiter par la couche d'entrée du réseau augmente.

### 4.3 Interprétation des modèles RGB

Le modèle multi-binaire ZGP n'améliore pas la performance du modèle binaire en RGB. En effet, ce modèle permet de supprimer certains faux positifs dus à des pixels de neige classés *cloud* par le modèle binaire mais il introduit des faux négatifs en déclassant en *snow* des pixels de nuage précédemment bien prédits. Cela est bien visible sur la Figure 6 si l'on compare les faux positifs (en rouge) et les faux négatifs (en bleu) des modèles ZGB binaire et multi-binaire en RGB. Ceci montre que les bandes RGB ne sont pas suffisamment discriminantes pour séparer correctement certains pixels de nuage et de neige malgré un entraînement spécifique sur ces deux classes. Ce phénomène est confirmé par le modèle multi-classes dont les résultats sont très similaires à ceux du multi-binaire. UNET s'en sort en général mieux en RGB que les modèles ZGP RGB. En effet, les filtres de convolution, en réalisant une analyse du contexte autour de chaque pixel, permettent au réseau de neurones de discriminer la neige des nuages en l'absence d'information spectrale peu pertinente. Le UNET échoue parfois à détecter les nuages à partir des seules bandes RGB en produisant des faux négatifs sur les nuages peu texturés ou proches de la saturation. Le modèle ZGP binaire tend plutôt à créer des faux positifs sur les régions à forte radiométrie (neige sur les reliefs de l'image SPARCS ou roches claires sur l'image BIOME).

### 4.4 Interprétation des modèles MS

Toutes les métriques des modèles ZGP MS sont améliorées par rapport à celles des modèles ZGP RGB, notamment le F-score, comme illustré sur les Figure 4 et Figure 5. Il est cependant intéressant de noter que l'ajout des bandes MS améliore le rappel du UNET mais dégrade fortement la qualité des prédictions (précision du UNET MS inférieure à celle du UNET en RGB). A la vue de ces résultats, on peut s'interroger sur la capacité d'un réseau de neurones comme le UNET à correctement exploiter l'information spectrale lorsque le nombre de bandes devient grand. A l'inverse, on remarque sur les images de la Figure 6, la très bonne capacité des modèles ZGP MS à détecter les nuages dans des conditions difficiles, notamment en présence de neige grâce à une information spectrale beaucoup plus riche qu'en RGB. Le modèle ZGP binaire MS tend cependant à produire encore quelques faux positifs sur les bords des zones enneigées des reliefs de l'image SPARCS. Ces faux positifs sont en partie corrigés par le modèle multi-binaire qui ne rajoute pas de faux négatifs lors de cette correction en exploitant l'information portée par les



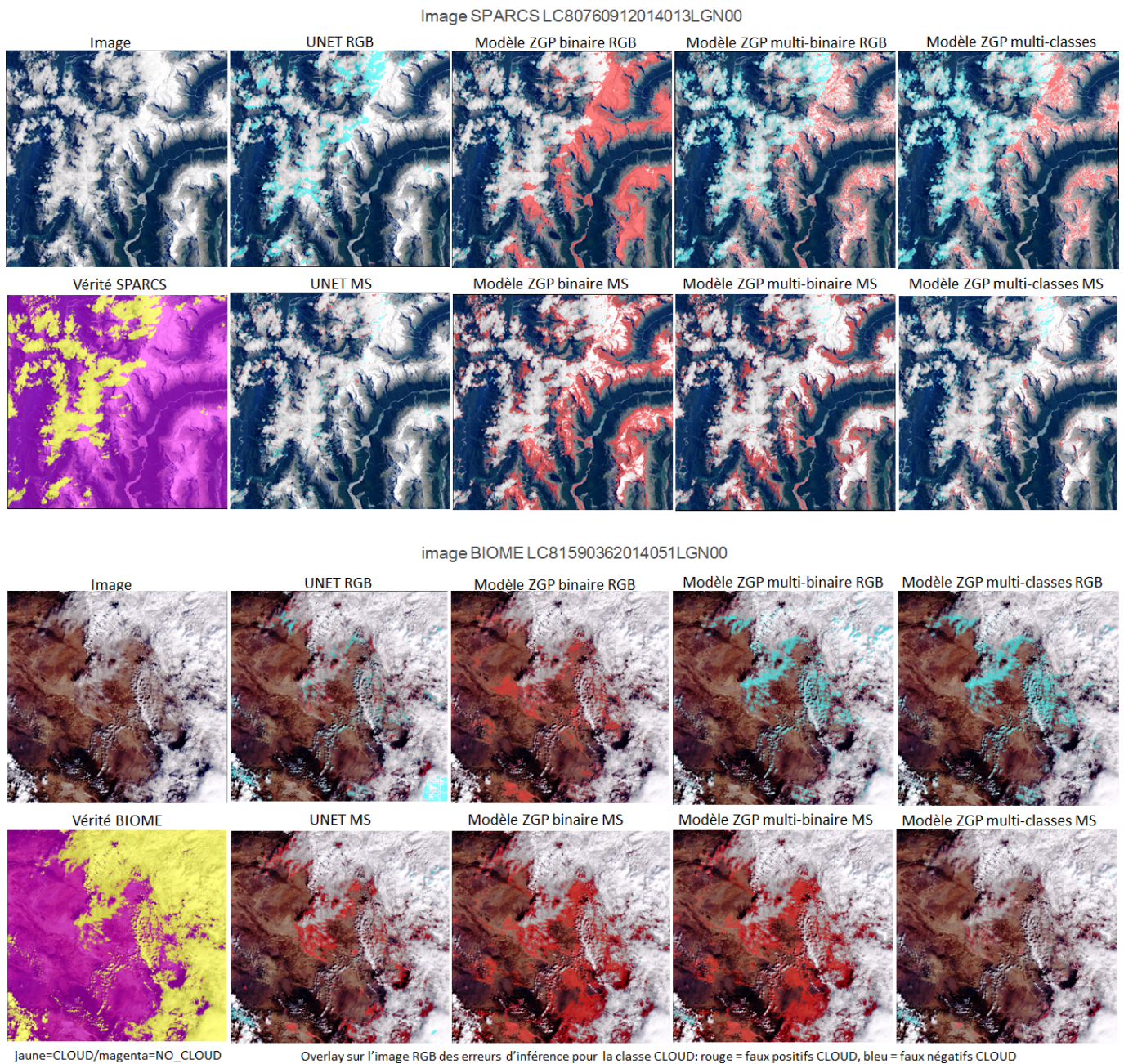


FIGURE 6 – Résultats d'inférence des modèles UNET et ZGP en RGB et MS sur une image extraite de la base de données SPARCS (en haut) et une image extraite de la base de données BIOME (en bas). Pour chaque modèle, les erreurs d'inférence sont superposées à l'image avec en rouge les faux positifs et en bleu les faux négatifs pour la classe *cloud* de la vérité.

bandes infrarouge SWIR et TIR sensibles à la température de surface. Sur l'image BIOME, les faux positifs du modèle ZGP binaire MS sont causés par des roches claires, ils ne sont par conséquent pas corrigés par la post-classification *cloudsnow* du modèle multi-binaire puisqu'aucun de ces faux positifs *cloud* ne correspond en réalité à de la neige. Cet exemple illustre les limites des modèles multi-binaires que n'ont pas les modèles multi-classes MS. Le modèle ZGP multi-classes MS est le plus performant de tous les modèles et rivalise avec les résultats du modèle de réf-

rence UNET MS d'un point de vue qualitatif (voir images Figure 6) et quantitatif (voir métriques Table 3 et Table 4), le dépassant notamment sur les images de test BIOME en terme d'*accuracy* globale et de F-score.

Le fait que les résultats des modèles ZGP soient meilleurs sur BIOME que sur SPARCS au contraire des modèles UNET, ne signifie pas pour autant que les modèles ZGP ont une capacité de généralisation supérieure. En effet, comme déjà évoqué précédemment les modèles ZGP sont en situation de généralisation sur 99% des données SPARCS en rai-

son du faible nombre d'échantillons nécessaire à leur entraînement. La différence de performance constatée entre les deux bases de données est donc liée à leur contenu respectif. La mesure des performances hors images contenant de la neige permet d'écarter un possible effet dû à une prépondérance de ce type de zones dans les images SPARCS. L'analyse visuelle des erreurs de prédictions montre que la majorité de ces erreurs est localisée sur les bords de nuages où la labélisation est souvent ambiguë. On observe par ailleurs que les images SPARCS présentent une majorité de nombreux petits nuages fragmentés plus propices à générer ce type d'erreurs de bord que les images BIOME peuplées essentiellement de zones nuageuses étendues.

## 5 Conclusion

Cette étude présente les performances de l'algorithme évolutionnaire ZGP utilisé dans la WebApp TADA appliqué à un cas d'usage dans le domaine de l'imagerie satellite : la segmentation de nuages. Les performances de différents modèles ZGP ont été comparées à celles de modèles issus d'un réseau de neurones profond de l'état de l'art, sur des images satellite publiques. Nous avons pu montrer que les performances des modèles ZGP pouvaient atteindre voire dépasser celles du réseau UNET, en particulier dans le cas de l'utilisation des images multi-spectrales. Cela démontre ainsi que l'information spectrale à elle seule permet la distinction des différentes classes de pixel, notamment entre les nuages et la neige. ZGP possède plusieurs avantages parmi lesquels sa frugalité en données d'apprentissage, l'interprétabilité des modèles fournis, la rapidité d'entraînement et d'inférence ainsi que sa facilité de déploiement et de portage sur des systèmes embarqués. En fonction des applications et objectifs considérés, ces éléments font de ZGP une bonne alternative aux réseaux de neurones, qui sont à l'inverse gourmands en données, en temps d'apprentissage, et difficiles à porter sur des systèmes embarqués.

Dans de futurs travaux, les performances des modèles générés par ZGP sur des images seront encore améliorées par la mise en oeuvre d'une stratégie permettant de sélectionner un nombre minimum d'échantillons d'apprentissage tout en maximisant la représentativité de cette sélection vis-à-vis des données d'entraînement. Nous envisageons aussi d'ajouter de l'information spatiale à l'aide de filtres de contexte comme ceux utilisés par les réseaux de neurones. Aussi, une technique hybride « réseau de neurones / ZGP » permettrait de tirer profit des avantages des deux types d'approches. Par exemple, la formule mathématique des modèles ZGP pourrait remplacer la fonction de classification utilisée dans la dernière couche des réseaux de neurones. Enfin, ZGP permettrait d'optimiser l'architecture et les hyper-paramètres d'un réseau de neurones, comme dans les travaux de Leung *et al.* [23] ou Mondal [27].

## Remerciements

Ces travaux ont été menés dans le cadre du projet CIAR ("Chaîne Image Autonome et Réactive") de l'Institut de Recherche Technologique Saint-Exupéry ([www.irt-](http://www.irt-saintexupery.com)

[saintexupery.com](http://www.irt-saintexupery.com)). Les auteurs remercient les partenaires industriels et académiques du projet : ActiveEon, Avisto, Elsys Design, GEO4i, Inria, LEAT/CNRS, MyDataModels, Thales Alenia Space et TwinswHeel.

## Références

- [1] Mohamad Awad. Improving satellite image segmentation using evolutionary computation. *American Journal of Remote Sensing*, 1 :13–20, 01 2013.
- [2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet : A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39 :2481–2495, 2017.
- [3] Gaetan Bahl, Lionel Daniel, Matthieu Moretti, and Florent Lafarge. Low-power neural networks for semantic segmentation of satellite images. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2469–2476, 2019.
- [4] Leonardo Bocchi, Lucia Ballerini, and Signe Hässler. A new evolutionary algorithm for image segmentation. In Franz Rothlauf, Jürgen Branke, Stefano Cagnoni, David Wolfe Corne, Rolf Drechsler, Yaochu Jin, Penousal Machado, Elena Marchiori, Juan Romero, George D. Smith, and Giovanni Squillero, editors, *Applications of Evolutionary Computing*, pages 264–273, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [5] Aurélie Boisbunon, Carlo Fanara, Ingrid Grenet, Jonathan Daeden, Alexis Vighi, and Marc Schoenauer. Zoetrope genetic programming for regression. In *2021 Genetic and Evolutionary Computation Conference (GECCO'21)*, 2021.
- [6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv :1706.05587v3*, 2017.
- [7] Josef Cihlar and J. Howarth. Detection and removal of cloud contamination from avhrr images. *IEEE Transactions on Geoscience and Remote Sensing*, 32(3) :583–589, 1994.
- [8] Kenneth De Jong. *Evolutionary Computation – A Unified Approach*. 01 2006.
- [9] Johannes Dröchner, Nikolaus Korfhage, Sebastian Egli, Markus Mühlhling, Boris Thies, Jörg Bendix, Bernd Freisleben, and Bernhard Seeger. Fast cloud segmentation using convolutional neural networks. *Remote Sensing*, 10(11), 2018.
- [10] Matthias Drusch, Umberto Del Bello, Stefane Carlier, Olivier Colin, Valerie Fernandez, Ferran Gascon, Bianca Hoersch, Claudia Isola, Paolo Laberinti, Philippe Martimort, Aimé Meygret, François Spoto, Omar Sy, Franco Marchese, and Pier Bargellini. Sentinel-2 : Esa's optical high-resolution mission for gmes operational services. *Remote Sensing of Environment*, 120 :25 – 36, 2012. The Sentinel Missions - New Opportunities for Science.
- [11] Agoston Eiben and Jim Smith. *Introduction To Evolutionary Computing*, volume 45. 01 2003.
- [12] Zhun Fan, Jiahong Wei, Guijie Zhu, Jiajie Mo, and Wenji Li. Evolutionary neural architecture search for retinal vessel segmentation. *arXiv preprint arXiv :2001.06678*, 2020.
- [13] Michael. E. Farmer and David. Shugars. Application of genetic algorithms for wrapper-based image segmentation and classification. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1300–1307, 2006.

- [14] Frederic Feresin, Michael Benguigui, Yves Bobichon, Edgard Lemaire, Matthieu Moretti, and Gaetan Bahl. On board images processing using ia to reduce data transmission : example of opssat cloud detection. In *7th On-Board Payload Data Compression Workshop, OBPDC*, 2020.
- [15] Steve Foga, Pat L. Scaramuzza, Song Guo, Zhe Zhu, Ronald D. Dilley, Tim Beckmann, Gail L. Schmidt, John L. Dwyer, M. Joseph Hughes, and Brady Laue. Cloud detection algorithm comparison and validation for operational landsat data products. *Remote Sensing of Environment*, 194 :379 – 390, 2017.
- [16] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition, 1989.
- [17] Olivier Hagolle, Mireille Huc, David Villa Pascual, and Gerard Dedieu. A multi-temporal method for cloud detection, applied to formosat-2, venus, landsat and sentinel-2 images. *Remote Sensing of Environment*, 114(8) :1747 – 1755, 2010.
- [18] Salvador Hinojosa, Omar Avalos, Jorge Gálvez, Diego Oliva, Eric Cuevas, and Marco A. Pérez-Cisneros. Remote sensing imagery segmentation based on multi-objective optimization algorithms. In *2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–6, 2018.
- [19] Ben V. Hollingsworth, Liqiang Chen, Stephen E. Reichenbach, and Richard R. Irish. Automated cloud cover assessment for Landsat TM images. In Michael R. Descour and Jonathan Martin Mooney, editors, *Imaging Spectrometry II*, volume 2819, pages 170 – 179. International Society for Optics and Photonics, SPIE, 1996.
- [20] Jacob Høxbroe Jeppesen, Rune Hylsberg Jacobsen, Fadil Inceoglu, and Thomas Skjødeberg Toftegaard. A cloud detection algorithm for satellite imagery based on deep learning. *Remote Sensing of Environment*, 229 :247 – 259, 2019.
- [21] Michael. D. King, Steven Platnick, W. Paul Menzel, Steven A. Ackerman, and Paul A. Hubanks. Spatial and temporal distribution of clouds observed by modis onboard the terra and aqua satellites. *IEEE Transactions on Geoscience and Remote Sensing*, 51(7) :3826–3852, 2013.
- [22] Chih-Chin Lai and Chuan-Yu Chang. A hierarchical evolutionary algorithm for automatic medical image segmentation. *Expert Systems with Applications*, 36(1) :248 – 259, 2009.
- [23] Frank. H. F. Leung, Hak-Keung Lam, Sai-Ho Ling, and Peter. K. S. Tam. Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *IEEE Transactions on Neural Networks*, 14(1) :79–88, 2003.
- [24] Bailin Li, Jianguo Lin, and Xiuming Yao. A novel evolutionary algorithm for determining unified creep damage constitutive equations. *International Journal of Mechanical Sciences*, 44(5) :987 – 1002, 2002.
- [25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.
- [26] Sorour Mohajerani and Parvaneh Saeedi. Cloud-net : An end-to-end cloud detection algorithm for landsat 8 imagery. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 1029–1032, 2019.
- [27] A. S. Mondal. Evolution of convolution neural network architectures using genetic algorithm. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2020.
- [28] Stelios K. Mylonas, Dimitris G. Stavrakoudis, John B. Theocharis, Georges C. Zalidis, and Ioannis Z. Gitas. A local search-based genesis algorithm for the segmentation and classification of remote-sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(4) :1470–1492, 2016.
- [29] Stelios K. Mylonas, Dimitris G. Stavrakoudis, and John B. Theocharis. Genesis : A ga-based fuzzy segmentation algorithm for remote sensing images. *Knowledge-Based Systems*, 54 :86 – 102, 2013.
- [30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science*, volume 9351, pages 234–241. Springer, 10 2015.
- [31] Christopher Walsh and Nick Taylor. Evolution of convolutional neural networks for lymphoma classification. In *24th International Conference on Image Processing, Computer Vision, and Pattern Recognition 2020, IPCV'20*, 2020.
- [32] Guo Yiqiang, Wu Yanbin, Ju Zhengshan, Wang Jun, and Zhao Luyan. Remote sensing image classification by the chaos genetic algorithm in monitoring land use changes. *Mathematical and Computer Modelling*, 51(11) :1408 – 1416, 2010. Mathematical and Computer Modelling in Agriculture.
- [33] Zhaoxiang Zhang, Guodong Xu, and Jianing Song. Cubesat cloud detection based on jpeg2000 compression and deep learning. *Advances in Mechanical Engineering*, 10, 2018.
- [34] Zhe Zhu, Shixiong Wang, and Curtis E. Woodcock. Improvement and expansion of the fmask algorithm : cloud, cloud shadow, and snow detection for landsats 4–7, 8, and sentinel 2 images. *Remote Sensing of Environment*, 159 :269 – 277, 2015.
- [35] Zhe Zhu and Curtis E. Woodcock. Object-based cloud and cloud shadow detection in landsat imagery. *Remote Sensing of Environment*, 118 :83 – 94, 2012.
- [36] Zhe Zhu and Curtis E. Woodcock. Automated cloud, cloud shadow, and snow detection in multitemporal landsat data : An algorithm designed specifically for monitoring land cover change. *Remote Sensing of Environment*, 152 :217 – 234, 2014.