

Détection d'anomalies à bord de satellites pour la protection de l'environnement marin

T. Goudemant¹, C. Szywala¹, B. Francesconi¹, M. Aubrun¹, E. Kervennic¹, I. Grenet^{1,2},
Y. Bobichon¹, M. Bellizzi¹, A. Girard¹

¹ Institut de Recherche Technologique Saint Exupéry

² MyDataModels

30 avril 2024

Résumé

Cet article présente un pipeline d'intelligence artificielle dédié à la détection d'anomalies dans des images maritimes, conçu pour être déployé à bord de satellites d'observation de la Terre. L'approche proposée est innovante en ce qu'elle requiert peu de données annotées, est capable de fonctionner sur des cibles embarquées à faible puissance de calcul, et est générique pour tous types d'anomalies se différenciant de l'état normal de l'eau. Ces qualités font de cette méthode un outil adaptable à diverses missions, ce qui lui a valu d'être respectivement sélectionnée et présélectionnée dans deux concours organisés par l'Agence Spatiale Européenne pour intégrer de l'intelligence artificielle à bord de Φ sat-2 et IMAGIN-e.

Mots-clés

Surveillance de l'environnement marin, intelligence artificielle embarquée, télédétection, détection d'anomalies, apprentissage frugal.

Abstract

This article describes a comprehensive artificial intelligence pipeline for detecting anomaly in maritime images, designed to be run on-board Earth observation satellites. The proposed method is innovative as it requires only weakly labelled data, can run on low-power embedded targets, and is generic for all types of anomalies deviating from the normal water state. These assets make this pipeline easily adaptable to a wide range of missions, resulting in its respective selection and preselection in two European Space Agency competitive challenges aimed at integrating artificial intelligence into Φ sat-2 and IMAGIN-e.

Keywords

Marine environmental monitoring, embedded artificial intelligence, remote sensing, anomaly detection, frugal learning.

1 Introduction

Les océans abritent une large diversité d'espèces marines et constituent une composante vitale de la biodiversité mondiale. Les technologies avancées, telles que l'imagerie satellite et l'intelligence artificielle (IA), peuvent aider les

scientifiques à identifier et à faire face aux menaces environnementales qui pèsent sur cet écosystème.

À titre d'exemple, la détection précoce de proliférations d'algues par des satellites joue un rôle essentiel dans l'atténuation des impacts écologiques, mais aussi dans la préservation des pêcheries qui constituent une source essentielle de nourriture pour des milliards de personnes dans le monde. Une étude [10] a estimé à 82 millions de dollars l'impact annuel moyen des proliférations d'algues nuisibles aux États-Unis. L'identification précoce de la pollution d'origine humaine, comme les marées noires ou les déchets plastiques, revêt également une importance cruciale : elle permet la mise en place d'opérations de nettoyage efficaces, réduit les pertes industrielles et contribue à la compréhension de ces menaces [13].

La plupart des recherches dans le domaine de la protection marine se concentrent sur des menaces spécifiques : évaluation de la qualité de l'eau [19, 23], concentrations de solides en suspension liées aux activités humaines [6] ou non [14], marées noires [2], déchets plastiques flottants [15]. Ces méthodes reposent généralement sur des calculs d'indices à l'aide de méthodes simples, comme combiner plusieurs bandes spectrales pour identifier des matériaux spécifiques. Mais elles rencontrent des difficultés lorsqu'elles sont appliquées à d'autres satellites ou d'autres zones géographiques. L'utilisation de méthodes statistiques ou d'apprentissage automatique classiques [26, 21, 25] font face à cette même difficulté d'adaptation à de nouvelles missions. À l'inverse, les techniques d'apprentissage profond ont produit des résultats de pointe ces dernières années. Les méthodes supervisées [16, 21, 25] offrent une bonne capacité d'adaptation à de nouvelles missions. Elles ne permettent cependant pas de généraliser à d'autres cas d'utilisation aussi bien que les méthodes auto-supervisées [26, 20, 25] et non supervisées basées sur la mesure de l'erreur de reconstruction d'image [26, 20, 21, 25]. L'utilisation de ces techniques d'apprentissage profond implique cependant dans la plupart des cas soit des coûts de calcul élevés, soit une annotation laborieuse des bases de données.

Contrairement à ces approches, nous avons choisi d'exploiter la technologie satellitaire pour détecter tous les événements inattendus, dont diverses menaces environnemen-

tales (sédiments, proliférations d’algues, débris plastiques, marées noires, etc.), y compris ceux qui ne pourraient pas être caractérisés. Nous désignerons dès lors l’ensemble de ces événements par le terme *anomalies*, qui englobe toutes sortes d’éléments se différenciant d’une normalité définie [9], représentée dans notre cas par un ensemble d’images maritimes sans perturbation des écosystèmes.

Notre approche doit également répondre aux contraintes établies par les défis OrbitalAI proposés par l’Agence Spatiale Européenne (*European Space Agency*, ESA). L’objectif de ces défis est de faire progresser l’*edge computing* en orbite en sélectionnant des propositions innovantes d’IA pour les déployer à bord de deux missions de télédétection :

- Le satellite Φ sat-2, un CubeSat 6U en orbite héliosynchrone équipé d’une caméra d’une résolution spatiale de 5 mètres et de 8 bandes spectrales dans le visible et le proche infrarouge. Φ sat-2 embarque un processeur CogniSAT™ d’Ubotica qui intègre un accélérateur Intel® Movidius™ Myriad™ 2.
- IMAGIN-e (*ISS Mounted Accessible Global Imaging Nod-e*), une démonstration de *space edge computing* par Microsoft et Thales Alenia Space. Il s’agit d’installer une caméra hyperspectrale et des capacités de calcul à bord de la Station Spatiale Internationale (*International Space Station*, ISS). Le système comporte une caméra ayant une résolution spatiale de 50 mètres et 50 bandes spectrales, et un processeur Cortex-A72 à 16 cœurs.

L’intérêt de l’*edge computing* est de réduire la réactivité des systèmes d’observation de la Terre en traitant les images directement sur les satellites, plutôt que d’attendre qu’elles soient transmises puis traitées dans les stations au sol. Cela nécessite de modifier des algorithmes conçus pour une infrastructure terrestre afin de les rendre légers et compatibles avec les cibles matérielles embarquées, qui sont limitées en puissance de calcul et en capacité mémoire.

Après un processus compétitif de plus d’un an, notre solution a été sélectionnée par l’ESA pour faire partie des deux nouvelles expériences qui seront embarquées à bord de Φ sat-2 (la compétition continue sur IMAGIN-e).

Notre solution est décrite dans la suite de l’article :

La section 2 détaille le fonctionnement des modèles composant le pipeline et justifie leur sélection en fonction de facteurs identifiés.

La section 3 décrit les données utilisées pour l’entraînement des modèles, ainsi que les étapes de prétraitement entreprises pour garantir la représentation fidèle des caractéristiques des capteurs.

La section 4 couvre l’entraînement, la validation et les résultats algorithmiques. Elle décrit également le déploiement sur la cible Myriad™ 2 et sur des CPU représentatifs des deux missions, ainsi que les performances matérielles obtenues.

2 Méthode

Dans cette section, nous présentons l’architecture de notre solution, puis justifions la sélection des modèles d’IA qui la

composent.

2.1 Description de l’architecture

L’architecture de notre solution, présentée sur la Figure 1, est constituée de trois parties : un *Tiler* pour découper l’image en patches de 32×32 pixels, un *Pipeline IA* pour détecter et caractériser les anomalies, et un *Mosaic’er* pour visualiser les sorties des modèles.

Le *Pipeline IA*, au cœur de notre solution, est composé de quatre modèles d’IA successifs :

1. Un modèle qui filtre uniquement les patches marins pour traitement ultérieur. Pour cette fonction, nous avons choisi d’utiliser un modèle de segmentation de la mer obtenu par l’algorithme génétique et frugal ZGP (*Zoetrope Genetic Programming*) [4].
2. Un modèle qui détecte les menaces potentielles sur les zones maritimes en mesurant la distance entre chaque zone et une normalité apprise. Ce modèle est constitué d’un réseau de neurones entraîné à l’aide de la méthode d’apprentissage auto-supervisé SimCLR [5], qui projette les patches dans un espace latent plus robuste et...
3. ... d’un *Gaussian Mixture Model* (GMM) [27] qui compare les caractéristiques des patches à une distribution définissant l’état dit *normal* de l’écosystème marin. Il génère un score d’anomalie allant de zéro à un pour chaque patch et conserve ceux avec un score significatif pour l’étape suivante.
4. Un autre modèle ZGP [4] qui caractérise précisément certains types d’anomalies en cas de dépassement d’un seuil. Il segmente les pixels des patches en trois classes : marée noire, prolifération d’algues et sédiments. Une classe *Autres* est également incluse pour les pixels normaux d’un patch, ou ceux présentant un type d’anomalie inconnu.

La description détaillée et la justification du choix de chaque modèle sont présentées dans la section 2.2.

Le *Mosaic’er*, quant à lui, recombine les patches de sortie des trois modèles et génère :

- une carte de segmentation terre/mer/nuages après le premier ZGP ;
- une carte de chaleur illustrant l’amplitude de l’anomalie pour chaque patch après le GMM ;
- une carte de segmentation du type d’anomalie au niveau pixel après le deuxième ZGP.

Cette solution modulaire offre une approche globale et peut être utilisée de manière polyvalente pour diverses applications opérationnelles, telles que :

- La priorisation d’images : les images dont les scores d’anomalie sont les plus élevés sont téléversées en priorité par le satellite afin de permettre aux opérateurs de se concentrer uniquement sur les images renfermant des informations cruciales lors de la surveillance de vastes étendues marines.
- L’envoi d’alertes : en cas de détection d’un incident majeur en mer, tel qu’une marée noire, une alerte est immédiatement déclenchée afin de favoriser une

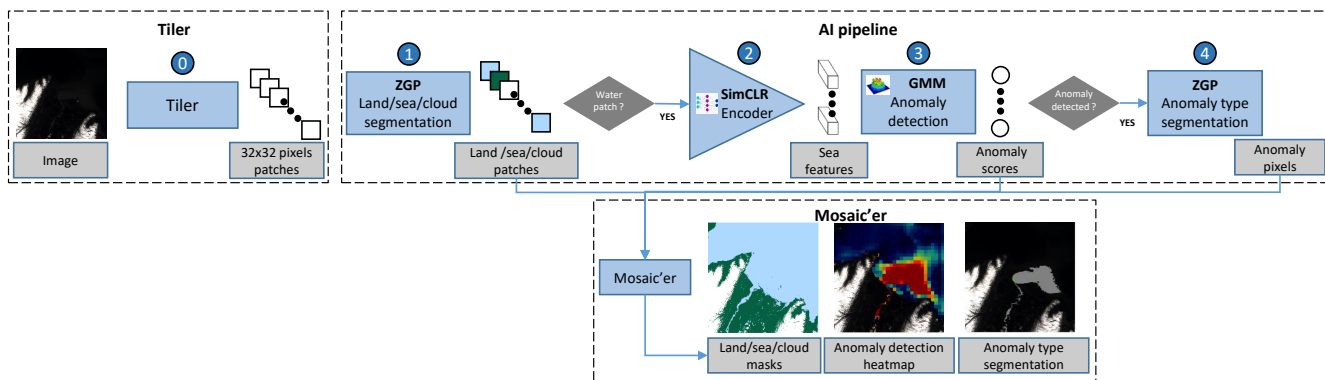


FIGURE 1 – Architecture du pipeline de détection d’anomalies. Elle est composée d’un *Tiler* (0) qui transforme l’image en patches. Les patches marins sont ensuite filtrés par le *Pipeline IA* (1), encodés dans un espace latent robuste (2), puis soumis à la détection d’anomalies (3). Les patches classés comme anomalies sont segmentés (4) afin de caractériser le type d’anomalie. Les sorties de chaque modèle peuvent être visualisées avec le *Mosaic'er* qui recombine les patches de sortie.

réaction rapide des autorités compétentes et d’atténuer les impacts environnementaux, sociaux et économiques.

2.2 Justification de l’architecture

Cette partie fournit des détails sur les différents modèles et justifie leur implantation à la lumière des limitations et objectifs décrits dans la section 1.

2.2.1 Segmentation marine et caractérisation des anomalies

Pour les étapes de segmentation marine et de caractérisation des anomalies, nous avons opté pour des modèles ZGP [4]. L’algorithme ZGP (*Zoetrope Genetic Programming*), développé par *MyDataModels*, intègre à la fois la programmation génétique et la régression symbolique. Son principe d’entraînement consiste à faire évoluer des individus représentés par des expressions mathématiques combinant des variables d’entrée et des constantes. Après des processus de mutation et de recombinaison sur plusieurs générations, le meilleur individu est sélectionné comme modèle final [4]. Les modèles obtenus sont des formules mathématiques simples combinant les variables d’entrée du modèle et des constantes numériques, avec des opérateurs mathématiques pouvant être algébriques, discrets, quadratiques ou trigonométriques.

Le choix de ZGP est motivé par ses nombreux avantages : frugalité en termes de données d’entraînement (environ mille pixels annotés par classe), rapidité d’inférence et facilité d’implantation dans les systèmes embarqués. En effet, sa mise en œuvre nécessite uniquement l’évaluation de la formule, et les deux modèles ZGP seront exécutés sur le processeur matériel en tant qu’étapes de prétraitement (pour la segmentation terre/mer/nuages) et de post-traitement (pour la caractérisation des anomalies).

2.2.2 Détection d’anomalies

Pour l’étape de détection d’anomalies, nous avons opté pour une méthode d’apprentissage profond auto-supervisé. Elle consiste à utiliser un encodeur pour extraire les caractéristiques de haut niveau d’une scène, combiné à un algorithme

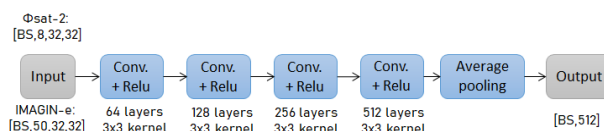


FIGURE 2 – Représentation de l’encodeur de patch composé de seulement quatre couches de convolution, et pouvant bénéficier d’une accélération matérielle. Les dimensions d’entrée et de sortie ainsi que la taille des couches apparaissent. BS (*Batch Size*) correspond à la taille des batches.

classique de détection d’anomalies. Ainsi, nous avons utilisé deux modèles pour cette étape : un encodeur SimCLR pour encoder chaque scène et un GMM pour leur attribuer un score d’anomalie.

L’approche SimCLR [5] est une méthode d’apprentissage auto-supervisé qui vise à représenter les patches de manière robuste en rapprochant les caractéristiques de patches similaires tout en éloignant celles de patches différents. C’est une méthode de référence en termes d’apprentissage de représentations, reconnue pour son efficacité avec des architectures de réseau de neurones petites et simples. Ainsi, notre choix s’est porté sur un petit encodeur, composé de seulement quatre couches de convolution (Figure 2). Ce modèle, avec une taille d’environ un million de paramètres, est environ dix fois plus petit que d’autres encodeurs légers courants tels que ResNet18 [8]. Une telle simplicité favorise l’adaptabilité à différentes missions en atténuant le risque de surapprentissage, tout en assurant la compatibilité avec divers accélérateurs matériels tels que l’Intel® Movidius™ Myriad™ 2 embarqué dans Φ sat-2. De plus, son entraînement auto-supervisé permet une adaptation aisée à différents ensembles de données et ne requiert pas d’annotation.

Le GMM [27] est un algorithme probabiliste d’apprentissage automatique qui utilise une combinaison de gaussiennes pour modéliser la distribution des patches enco-

TABLE 1 – Caractéristiques des modèles du pipeline en termes de frugalité des données et d’embarquabilité.

Modèle	App.	Frugalité données	Type modèle	Exécution
Modèles ZGP	supervisé	1000 pixels annotés par classe	Formule math.	CPU pour pré-/post-traitement
Encodeur SimCLR	auto-supervisé	10 000 patches non annotés	Petit CNN (4 couches)	Accélération matérielle ou CPU
GMM	faiblement supervisé	1000 patches sans anomalie	Formule math.	CPU

dés. Pendant la phase d’apprentissage, les paramètres statistiques sont optimisés pour modéliser la distribution de patches représentant la normalité. Ensuite, lors de la phase de détection d’anomalies, les scores de vraisemblance produits par le GMM mesurent la déviation de chaque patch encodé par rapport à cette normalité. En fixant un seuil approprié, les anomalies peuvent être identifiées parmi les patches encodés. Le choix de cet algorithme est motivé par sa frugalité en termes de ressources informatiques et de données (environ mille patches d’entraînement avec pour seule annotation la classe *mer sans anomalie*). Notre approche, axée sur l’apprentissage de la distribution de la mer plutôt que sur celui des menaces, permet de détecter toutes les anomalies. En termes de déploiement, l’évaluation du modèle nécessite peu de ressources et peut être effectuée sur le CPU.

Le Tableau 1 récapitule les caractéristiques des modèles composant le pipeline. Nous avons choisi un modèle d’apprentissage profond auto-supervisé pour une bonne généralisation à différentes missions. L’encodeur et le GMM ont été sélectionnés pour la détection d’anomalies en raison de leurs faibles besoins en données étiquetées. Le modèle supervisé ZGP a été choisi pour sa capacité à caractériser précisément des anomalies en s’entraînant sur peu de données. Ajoutons que l’encodeur, qui est le modèle le plus gourmand en termes de capacités informatiques, a été réduit à quatre couches et que les autres modèles nécessitent très peu de ressources matérielles.

3 Base de données

Nous avons utilisé plusieurs ensembles de données d’images et d’annotations représentatives de Φ sat-2 et IMAGIN-e pour entraîner et tester notre pipeline. Nous les décrivons dans cette section.

3.1 Choix des données en fonction des exigences des modèles

Les modèles constituant notre pipeline nécessitent des ensembles de données adaptés à leurs spécificités d’entraînement respectives.

Pour l’encodeur qui requiert une importante quantité de

données non annotées représentatives des capteurs, nous avons choisi d’utiliser les ensembles de données Φ sat-2 et IMAGIN-e fournis par l’ESA dans le cadre du défi OrbitalAI, et désignés dans la suite de l’article sous le nom [DS_challenge] (section 3.2).

Pour le GMM, nous avons constitué un sous-ensemble de [DS_challenge] composé uniquement d’images de mer sans anomalies afin de l’entraîner à assimiler le concept de *normalité marine*. Puis, nous avons construit un ensemble de données contenant des anomalies, désigné sous le nom [DS_custom], qui a été créé en simulant des images Φ sat-2 et IMAGIN-e à partir de dates d’événements connus (section 3.3). Un sous-ensemble de ces images a été manuellement annoté pour une évaluation quantitative (section 4.2.2), tandis que les autres ont été utilisées pour une évaluation visuelle (section 4.2.1).

Le premier ZGP est entraîné sur des images annotées terre/mer/nuages afin de filtrer les patches marins, tandis que le deuxième ZGP est entraîné sur des images annotées d’anomalies afin de les caractériser. Pour entraîner et tester les ZGP, nous avons respectivement annoté manuellement un sous-ensemble de [DS_challenge] et utilisé un sous-ensemble de [DS_custom].

Des détails supplémentaires sur ces ensembles de données sont fournis dans ce qui suit.

3.2 DS_challenge : jeux de données fournis dans le cadre du défi OrbitalAI

Deux ensembles de données représentatifs de Φ sat-2 et d’IMAGIN-e ont été fournis dans le cadre des défis OrbitalAI de l’ESA. Ils comprennent 489 scènes et masques nuageux simulés à partir de produits Sentinel-2 L1C multispectraux convertis en unités de radiance en haut de l’atmosphère (*Top Of Atmosphere*, TOA). Dans le cas d’IMAGIN-e, 497 scènes supplémentaires ont également été simulées et fournies par l’ESA à partir de produits Level-1 PRISMA hyperspectraux. Nous n’avons pas utilisé ces dernières dans notre étude car le délai pour obtenir les produits PRISMA supplémentaires dont nous aurions eu besoin pour annoter des anomalies était incompatible avec le calendrier serré du défi. Chaque scène est divisée en patches de 256×256 pixels (sans chevauchement), ce qui permet d’obtenir respectivement 256 et 16 patches par scène avec 8 et 50 bandes spectrales VIS/NIR pour Φ sat-2 et IMAGIN-e.

Un sous-ensemble de [DS_challenge], comprenant environ 60 scènes, a été annoté manuellement avec des masques de segmentation terre/mer afin d’entraîner et de tester le premier ZGP dédié au filtrage des scènes marines. Un autre sous-ensemble, comprenant environ 70 patches, a été classifié manuellement en catégories telles que végétation, sol nu, neige, nuage et eau afin d’évaluer les capacités de représentation de l’encodeur. Enfin, un sous-ensemble de patches marins a été sélectionné pour définir l’état normal de la mer en prévision de l’entraînement du GMM.

Remarquons que pour entraîner l’encodeur et le GMM, les patches sont subdivisés en patches plus petits de 32×32 pixels. Nous obtenons ainsi environ 8 millions de patches pour Φ sat-2 et 500 000 pour IMAGIN-e.

3.3 DS_custom : jeu de données d’anomalies personnalisé

Des données supplémentaires ont été recueillies pour évaluer la performance du détecteur d’anomalies et entraîner le ZGP dédié à leur caractérisation. Ces données comprennent des événements de pollution extraits de la littérature scientifique, de contributions d’experts et de communiqués de presse : marées noires [12], proliférations d’algues [7, 11], sédiments et déchets plastiques [3, 24, 17]. Ces événements sont répertoriés dans le Tableau 2, accompagnés des coordonnées géographiques et des dates utilisés pour acquérir les images Sentinel-2 correspondantes.

Les images représentatives des capteurs de Φ sat-2 et IMAGIN-e ont été générées par simulation à partir de ces scènes Sentinel-2 (section 3.4). Les anomalies ont été annotées manuellement par des experts en traitement d’images, avec des masques de segmentation au niveau des pixels pour chaque type d’anomalie ainsi que pour les zones terrestres et marines. Les masques nuageux proviennent de Sentinel-2 et ont été récupérés par le simulateur.

Comme il était impossible de segmenter manuellement les images contenant des anomalies larges et diffuses telles que des proliférations d’algues, celles-ci ont été conservées dans un ensemble distinct dédié à l’évaluation visuelle des performances des modèles.

3.4 Présentation des simulateurs Φ sat-2 et IMAGIN-e

Des *notebooks* Python utilisant la bibliothèque EO-learn ainsi que des exécutables spécifiques ont été fournis dans le cadre d’OrbitalAI pour simuler des images représentatives de Φ sat-2 et IMAGIN-e :

- simulation d’images Φ sat-2 à partir de données Sentinel-2 L1C ;
- simulation d’images IMAGIN-e à partir de données Sentinel-2 L1C ou de données PRISMA.

Ces *notebooks* permettent de définir une liste de demandes et de récupérer les données Sentinel-2 correspondantes rééchantillonnées depuis SentinelHub (avec les identifiants appropriés).

Les simulateurs permettent ensuite de produire des données de niveau 1A (radiance TOA sans correction des décalages spatiaux entre les bandes spectrales) ou de niveau 1C (réflectance TOA avec les bandes spectrales corégréées). Nous avons choisi d’entraîner les modèles de notre pipeline exclusivement sur des images L1C en raison du risque d’erreurs que pourrait générer la dérégistration entre bandes spectrales. À l’avenir, nous pourrions envisager d’adapter le pipeline à des données L1A.

L’architecture des simulateurs Φ sat-2 et IMAGIN-e est similaire, notamment avec les produits Sentinel-2. Le processus de traitement est illustré sur la Figure 3, qui met en évidence les étapes spécifiques à Φ sat-2 et à IMAGIN-e. À la fin du processus, l’image simulée peut être découpée selon les besoins des algorithmes d’apprentissage, et est sauvegardée au format Geotiff, incluant toutes les bandes et certains masques (nuages provenant des données Sentinel-2).

TABLE 2 – Dates, positions et types d’anomalies marines utilisées pour simuler les images de Φ sat-2 et IMAGIN-e à partir de celles de Sentinel-2.

ID	Date	Latitude	Longitude	Anomalie
AN1	07/08/17	20,8399	37,6159	Marée noire
AN2	26/02/23	11,2167	57,7782	Marée noire
AN3	07/08/17	20,8452	37,6104	Marée noire
AN4	27/08/17	20,8511	37,5657	Marée noire
AN5	08/09/17	-0,7802	37,7166	Algues
AN6	29/07/17	114,0258	22,256	Algues
AN7	29/07/17	114,0961	22,1866	Algues
AN8	14/05/18	-14,2314	65,6917	Sédiments
AN9	08/08/22	21,632	59,458	Algues
AN10	08/08/22	22,476	59,601	Algues
AN11	11/06/21	26,5247	39,0395	Plastique
AN12	21/06/21	26,5247	39,0395	Plastique
AN13	31/10/18	0,1420	5,6920	Plastique
AN14	31/10/18	0,0435	5,6154	Plastique
AN15	01/03/23	17,3799	55,7791	Marée noire
AN16	10/08/21	26,5247	39,0395	Plastique
AN17	27/09/21	28,152	121,216	Algues
AN18	08/03/21	1,75	105,17	Marée noire
AN19	02/04/21	1,75	105,17	Marée noire
AN20	18/03/21	2,5229	106,0052	Marée noire
AN21	02/04/21	2,0914	105,1947	Marée noire
AN22	02/05/21	2,4545	105,1541	Marée noire
AN23	10/11/20	17,4875	-88,175	Sédiments
AN24	15/11/20	17,4875	-88,175	Sédiments
AN25	30/11/20	17,4875	-88,175	Sédiments
AN26	28/08/21	35,1854	35,8728	Marée noire
AN27	02/09/21	35,1854	35,8728	Marée noire
AN28	07/09/21	35,3942	35,8363	Marée noire
AN29	07/02/19	5,6147	95,2998	Sédiments
AN30	09/02/19	5,6004	95,3226	Sédiments
AN31	18/01/22	-11,9581	-77,2195	Marée noire

4 Expériences menées

Cette section détaille l’entraînement des modèles, puis présente les résultats de détection pour diverses anomalies (plastiques, marées noires, proliférations d’algues, sédiments) sur deux missions (Φ sat-2 et IMAGIN-e) afin de mettre en avant la polyvalence de notre approche. Les performances de déploiement des modèles sur deux processeurs embarqués, ainsi que celles de l’encodeur sur l’accélérateur matériel de Φ sat-2, seront également présentées.

4.1 Paramètres d’entraînement des modèles

Pour entraîner le premier et le second ZGP, nous avons sélectionné de manière aléatoire 1000 pixels par classe respectivement dans [DS_challenge] et dans [DS_custom]

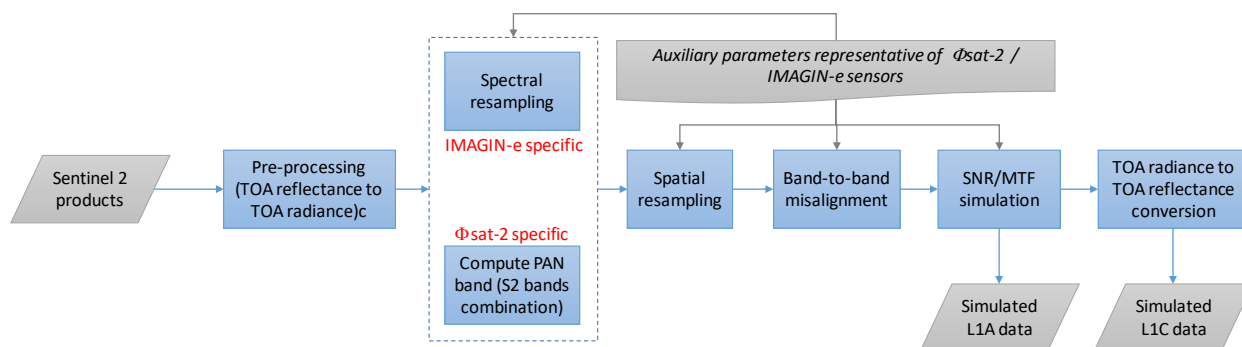


FIGURE 3 – Architecture des simulateurs IMAGIN-e et Φ sat-2, avec les spécificités des capteurs mises en évidence (rééchantillonnage spectral afin de correspondre au capteur hyperspectral d’IMAGIN-e, calcul de la bande panchromatique à partir des bandes multispectrales pour Φ sat-2). Plusieurs paramètres sont simulés lors des différentes étapes : résolution spatiale (50 m pour IMAGIN-e, 5 m pour Φ sat-2), décalages entre les bandes spectrales liés à l’acquisition, rapport signal sur bruit (*Signal-to-Noise Ratio*, SNR) et fonction de transfert de modulation (*Modulation Transfer Function*, MTF) des capteurs. Les simulateurs permettent de choisir le niveau du produit, avec des différences notables entre L1A et L1C (décalage entre les bandes plus important pour des données L1A car non corrigé, nécessité de convertir les données de radiance TOA en données de réflectance TOA si l’on souhaite simuler des données L1C).

(section 3). Cette sélection s’est faite en séparant méticuleusement les données d’entraînement et de test pour assurer l’indépendance entre ces deux phases. Les modèles sont ensuite entraînés de manière supervisée en utilisant l’algorithme de *MyDataModels* [4].

L’entraînement de l’encodeur s’inspire quant à lui d’une implantation PyTorch [22] qui vise à minimiser la perte d’*entropie croisée à température normalisée* [5] de l’architecture représentée sur la Figure 2. Les données d’entraînement comprennent 10 000 patches non étiquetés de 32×32 pixels issus de diverses scènes de [DS_challenge], en utilisant toutes les bandes spectrales. Les modèles ont été entraînés sur 80 époques, avec une taille de batch de 16 patches, un coefficient de température de 50 % et un taux d’apprentissage de 0,0003, en démarrant avec des poids aléatoires. Ce modèle prend environ une heure pour converger sur un processeur Intel® Xeon® Silver 4114 à quatre cœurs. L’architecture du modèle génère en sortie des vecteurs de 512 dimensions correspondant aux représentations encodées des patches dans un espace latent.

Enfin, le GMM a été entraîné sur 7 000 patches sans anomalies issus d’un sous-ensemble de [DS_challenge] et [DS_custom]. L’entraînement consiste à optimiser les caractéristiques de dix composantes gaussiennes de façon à ajuster leur distribution à celle des patches via un algorithme EM (*Expectation-Maximization*) implémenté à partir de la bibliothèque scikit-learn [18]. Le score d’anomalie fourni pour chaque patch de test est calculé à partir de la densité de la composante gaussienne la plus probable.

Les modèles entraînés pour les deux défis seront disponibles sur [1].

4.2 Résultats algorithmiques

Nous présentons dans cette partie une validation visuelle des sorties de notre solution pour chacune des deux missions (section 4.2.1), suivie d’une évaluation quantitative

de la performance algorithmique sur quatre images d’anomalies (4.2.2).

4.2.1 Évaluation qualitative

Pour valider nos modèles, nous analysons leur sortie avec le *Mosaic’er* (section 2.1). La Figure 4 montre les résultats obtenus sur deux images pour les missions Φ sat-2 et IMAGIN-e. Dans les deux cas, le premier ZGP segmente correctement la mer, les patches marins sont analysés par l’encodeur puis le GMM détecte bien l’anomalie, qui est ensuite classifiée correctement (prolifération d’algues) par le deuxième ZGP. Pour confirmer l’absence de faux positifs, ce processus d’évaluation visuelle a été, pour chaque mission, reproduit sur des images supplémentaires sans anomalie.

4.2.2 Évaluation quantitative

Nous avons évalué quantitativement les performances des ZGP et du GMM en utilisant des métriques classiques telles que la précision, le rappel, le *FI-score* et l’aire sous la courbe précision-rappel (AUROC). Ces métriques ont été calculées en comparant les sorties des modèles à des vérités terrain annotées manuellement. Nous avons utilisé deux images par mission, totalisant 35,6 millions de pixels ou 34,8 milliers de patches (32×32 pixels par patch). Les performances algorithmiques sont regroupées dans le Tableau 3, et la Figure 5 montre la visualisation des résultats des modèles pour les quatre images.

Pour évaluer les performances du premier ZGP, nous nous sommes concentrés exclusivement sur la segmentation des pixels marins, puisque l’objectif de ce modèle est d’envoyer uniquement des patches de mer à l’encodeur. Cette tâche affiche des performances très élevées, avec des scores de précision et de rappel dépassant en général 97 %.

Nous avons ensuite évalué la performance en détection du GMM indépendamment des résultats du modèle précédent. Pour ce faire, nous l’avons fait tourner sur des patches dont

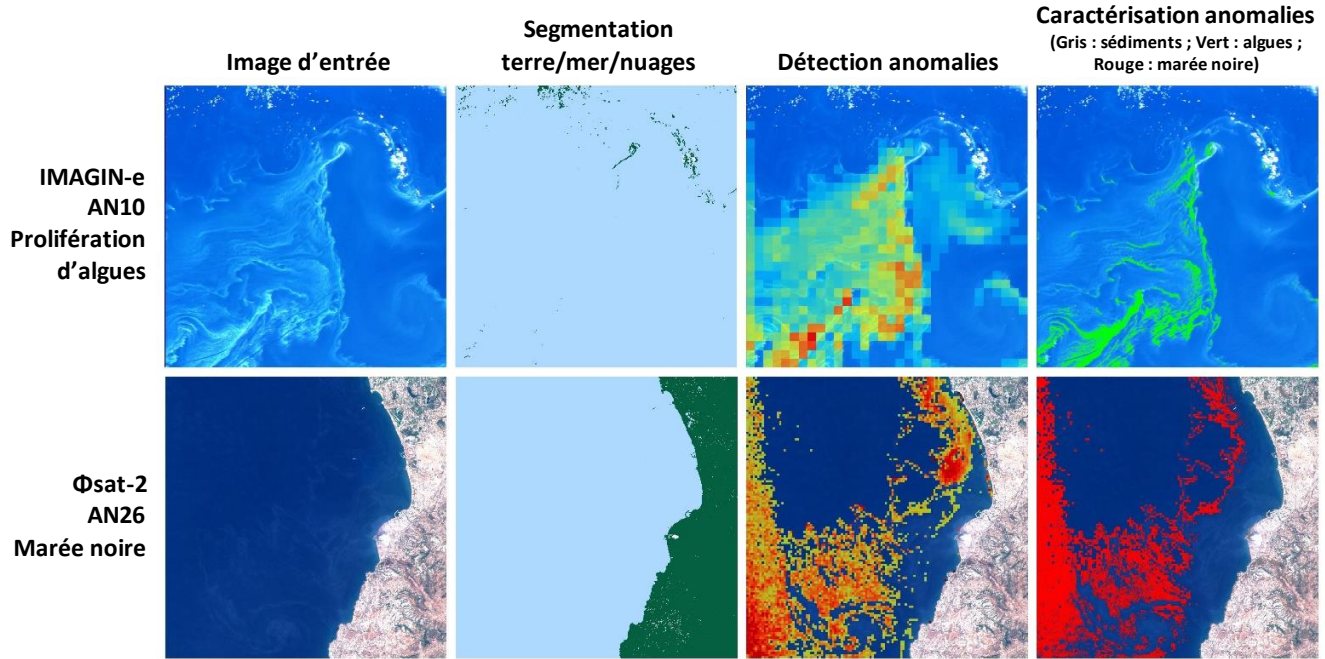


FIGURE 4 – Validation visuelle des sorties des ZGP et du GMM pour IMAGIN-e sur AN10 (en haut) et pour Φ sat-2 sur AN26 (en bas) (cf. Tableau 2). Dans l'ordre : visualisation RGB de l'image d'entrée, segmentation terre/mer/nuages par ZGP, carte de chaleur des anomalies par GMM (rouge : score proche de 1 ; bleu : score proche de 0), et caractérisation des anomalies par ZGP. La prédiction du modèle est superposée à l'image originale pour la détection et la caractérisation.

TABLE 3 – Performances algorithmiques des modèles sur les quatre images de test (Figure 5). V.T. = Vérité Terrain.

Image	V.T.	Préc.	Rap.	F1-score	AUROC
Segmentation mer, ZGP (V.T. = pixels mer)					
#1 AN13	11,1M	96,6 %	99,9 %	98,2 %	–
#2 AN5	6,1M	99,1 %	83,1 %	90,4 %	–
#3 AN23	558K	98,8 %	96,4 %	97,5 %	–
#4 AN27	1042K	97,7 %	97,9 %	97,8 %	–
moy.	33,3M	98,1 %	94,3 %	96,0 %	–
Détection anomalies, GMM (V.T. = patches anom.)					
#1 AN13	17	47,8 %	64,7 %	55,0 %	44,2 %
#2 AN5	4501	74,8 %	99,8 %	85,5 %	87,4 %
#3 AN23	71	50,0 %	61,9 %	55,3 %	60,8 %
#4 AN27	33	37,7 %	69,7 %	48,9 %	38,6 %
moy.	4622	52,6 %	74,0 %	61,2 %	57,7 %
Caractérisation anomalies, ZGP (V.T. = pixels anom.)					
#1 AN13	2K	100 %	99,7 %	99,8 %	–
#2 AN5	4M	100 %	99,8 %	99,9 %	–
#3 AN23	37K	100 %	97,3 %	98,6 %	–
#4 AN27	6K	100 %	96,5 %	98,2 %	–
moy.	4M	100 %	98,3 %	99,1 %	–

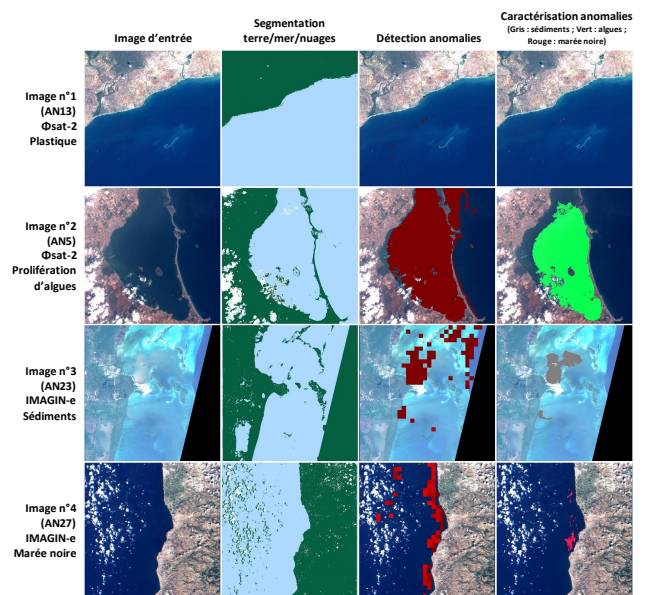


FIGURE 5 – Exemples de prédictions sur quatre images de test (AN13, AN5, AN23, AN27, cf. Tableau 2) pour Φ sat-2 (lignes 1 et 2) et IMAGIN-e (lignes 3 et 4). Colonne 1 : images ; Colonnes 2 à 4 : sorties reconstruites des modèles, avec une visualisation similaire à celle de la Figure 4.

la vérité terrain contenait plus de 80 % de mer. Pour classer un patch comme anomalie, nous avons sélectionné des

seuils de détection optimisant la performance du modèle. La performance moyenne du modèle tourne autour de 60 % en termes de rappel et de précision. Cette performance est

dégradée par le fait que le critère de 80 % pour l'identification de la mer conduit à inclure des patches comprenant des lignes côtières qui génèrent des faux positifs et réduisent la précision. D'autre part, la prédiction des anomalies au niveau patch, comparée à l'annotation au niveau pixel, entraîne simultanément la diminution de la précision et du rappel — en particulier pour de petits objets tels que les marées noires. Notons que les cas d'utilisation présentés dans la section 1, et notamment la priorisation du téléversement des images, ne nécessitent ni une précision ni un rappel très élevé. Pour une mesure de performance indépendante du seuil de détection choisi, nous fournissons également dans le Tableau 3 les résultats de l'aire sous la courbe précision-rappel (AUROC).

Nous avons également évalué le second ZGP indépendamment des résultats du modèle précédent : nous avons effectué la caractérisation des anomalies sur des pixels annotés comme tels dans la vérité terrain, plutôt qu'en se basant sur les prédictions du GMM. Ceci a conduit à des performances élevées, atteignant près de 100 % en précision et en rappel. Ces résultats sont illustrés sur la Figure 5.

4.3 Performances matérielles

Dans cette dernière partie, nous présentons le déploiement de nos modèles sur des cibles matérielles représentatives des deux missions. Pour Φ sat-2, nous avons déployé l'encodeur de patch sur la cible Myriad™ 2, et les autres modèles ZGP / GMM sur un processeur ARM Cortex-A53. Pour IMAGIN-e, qui ne dispose pas d'accélérateur matériel embarqué, nous avons utilisé un CPU ARM Cortex-A72 pour le déploiement et l'évaluation des performances de l'ensemble des modèles.

4.3.1 Déploiement sur cibles matérielles

Déploiement sur Φ sat-2. La plateforme d'exécution du satellite Φ sat-2 est le Ubotica CogniSAT-XE1™. Il est construit à partir de l'accélérateur Intel® Myriad™ 2, conçu pour l'inférence de réseaux de neurones sur des cibles embarquées. À des fins de validation, nous avons utilisé le Movidius Neural Compute Stick 1 (NCS1), qui abrite l'unité de traitement de vision Myriad™ 2 (VPU). Myriad™ 2 comprend douze cœurs SHAVE (cœurs de traitement vectoriel sur 128 bits) pour les calculs des réseaux de neurones et prend en charge les arithmétiques 16 et 32 bits en virgule flottante. Avec 2 Mo de mémoire intégrée, elle peut être configurée pour prendre en charge différentes combinaisons d'instructions et de données en fonction de la charge de travail. Le déploiement sur Myriad™ 2 utilise l'outil logiciel OpenVINO 2020.3, qui permet de charger un modèle ONNX et de le convertir au format du moteur d'inférence avec le *Model Optimizer*. Nous avons validé le déploiement de l'encodeur sur la cible en comparant, pour plusieurs patches d'entrée, la cohérence des sorties (patches encodés) avant et après le déploiement. Les ZGP et le GMM ont quant à eux été déployés sur le processeur ARM Cortex-A53 de la carte d'évaluation Xilinx Zynq® UltraScale+™ MPSoC ZCU104. Un code Python instrumenté a été utilisé pour exécuter les trois modèles,

TABLE 4 – Performances matérielles des modèles.

Modèle	Cible	Données traitées	Latence (ms)	Débit par patch
ZGP (seg. terre/ mer/nuages)	ARM A53	1 patch (32 × 32 pixels)	11,3	90,3 Kpixels/s
	ARM A72		4,6	224,9 Kpixels/s
Encodeur (taille batch = 128)	Myriad™ 2	1 patch (32 × 32 pixels)	16,6	61,5 Kpixels/s
	ARM A72		57,4	17,8 Kpixels/s
GMM (détection anomalies)	ARM A53	1000 patches encodés	29,3	34,5 Kpatches/s
	ARM A72		15,7	67,1 Kpatches/s
ZGP (caract. anomalies)	ARM A53	1 patch (32 × 32 pixels)	13,1	77,8 Kpixels/s
	ARM A72		5,9	174,4 Kpixels/s

mesurant simultanément leurs temps d'exécution moyens. Nous avons également confirmé que les sorties de ces trois modèles étaient identiques avant et après le déploiement.

Déploiement sur IMAGIN-e. La cible matérielle à bord de l'ISS comporte un processeur ARM Cortex-A72 à 16 cœurs et dispose de 32 Go de mémoire SDRAM DDR4. Les deux ZGP, l'encodeur et le GMM ont été déployés sur le processeur ARM Cortex-A72 bicœur de la carte d'évaluation Xilinx Versal™ VCK190. Le même code Python instrumenté a été utilisé pour exécuter les quatre modèles, mesurant simultanément leurs temps d'exécution moyens. Nous avons confirmé que toutes les sorties étaient identiques avant et après le déploiement.

4.3.2 Métriques d'inférence

Pour évaluer les performances matérielles de nos modèles, nous avons mesuré le temps d'exécution moyen (latence) et le débit par patch pour chaque mission (Tableau 4).

Pour les ZGP exécutés sur le CPU, le débit est d'environ 90 Kpixels/s sur ARM A53 et 225 Kpixels/s sur ARM A72.

Concernant l'encodeur de patch exécuté sur Myriad™ 2, nous avons étudié l'influence de la taille du batch sur la latence et le débit. Les résultats, représentés sur la Figure 6, indiquent une utilisation optimale de la cible pour des batches de 128 patches. Au-delà, la latence et le débit restent constants (respectivement 16,6 ms/patch et 61,5 Kpixels/s) indépendamment de l'arithmétique choisie (virgule flottante de 16 ou 32 bits). Cependant, sans accélération matérielle, la latence chute à 18 Kpixels/s sur ARM A72.

Le GMM présente une remarquable capacité de débit, traitant jusqu'à 34 Kpatches encodés par seconde pour ARM A53 et 67 Kpatches pour ARM A72. Remarquons que ce modèle fonctionne sur des patches encodés de 32 × 32 pixels en entrée de l'encodeur. Par conséquent, ce débit

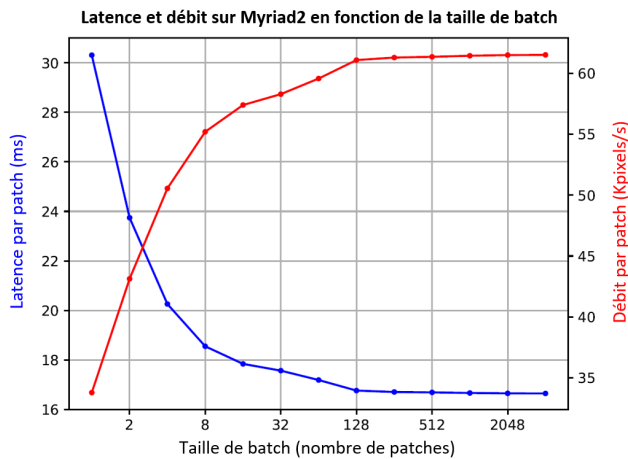


FIGURE 6 – Latence et débit de l’encodeur en fonction de la taille de batch. Optimum : batches de 128 patches.

dépasse significativement celui des ZGP.

Remarquons que les exigences en débit ne sont pas uniformes tout au long de la chaîne de traitement : le premier ZGP doit classifier tous les pixels, l’encodeur et le GMM ne traitent que les patches marins, et le dernier ZGP ne traite que les patches marins identifiés comme des anomalies. De plus, le GMM opère sur des patches encodés de 32×32 pixels.

Ainsi, le goulot d’étranglement de la chaîne est soit le premier ZGP, soit l’encodeur, selon la proportion de patches marins dans l’image. Pour accélérer le premier ZGP, prédire seulement deux classes (mer/non-mer) au lieu de trois (terre/mer/nuages) pourrait réduire le temps d’inférence de 33 %. Il pourrait également être envisagé de le déployer sur un accélérateur matériel tel que Myriad™ 2. Enfin, remarquons que si la cible matérielle dispose d’un CPU multicœur, toute la chaîne peut être exécutée simultanément.

5 Conclusion & Perspectives

Cet article présente une solution originale combinant un réseau de neurones auto-supervisé avec des modèles frugaux pour la détection d’anomalies dans des images satellites. En se concentrant sur l’apprentissage d’une distribution de la normalité plutôt que sur des menaces spécifiques, cette approche améliore la généricité de la détection, permettant l’identification d’une grande variété d’événements. Nous avons illustré cette polyvalence en traitant trois types d’événements qui affectent les écosystèmes marins : les proliférations d’algues, les marées noires et les inondations de sédiments.

Nous avons démontré l’efficacité de notre approche et son adaptabilité en ayant été respectivement sélectionnés et pré-sélectionnés sur les deux défis de l’Agence Spatiale Européenne pour embarquer de l’intelligence artificielle à bord de Φ sat-2 et IMAGIN-e. La validation sur des données représentatives des missions a montré de bonnes performances en détection et caractérisation des anomalies. De plus, le déploiement sur l’accélérateur matériel Intel® My-

riad™ 2 et les processeurs ARM Cortex-A53/A72 a permis d’évaluer les performances de latence et de débit de chaque modèle et de déterminer les facteurs limitants.

À l’avenir, nous prévoyons d’améliorer notre solution en intégrant davantage de données provenant de différentes sources. L’objectif sera d’améliorer le champ de la normalité marine et d’affiner la caractérisation des performances algorithmiques. De plus, nous envisageons de tester la sensibilité de notre solution à différents phénomènes provenant des capteurs tels que le bruit ou le décalage entre bandes spectrales. Notre objectif ultime demeure de démontrer le potentiel et la réactivité de notre solution grâce à son déploiement opérationnel sur Φ sat-2. Enfin, nous espérons également faire partie des deux équipes qui pourront déployer leur solution sur la mission IMAGIN-e.

Remerciements

Le travail présenté dans cet article est mené dans le cadre du projet IRMA (IA pour la Réactivité Mission basée sur l’Analyse image) de l’IRT Saint Exupéry. Ce projet vise à développer des briques technologiques basées sur l’intelligence artificielle pour la planification de missions et le traitement des données, à bord ou au sol, avec l’objectif de répondre aux nouveaux besoins du marché de l’observation de la Terre par satellites, tels que la transformation de données brutes en informations interprétées, et la gestion réactive de constellations de satellites. IRMA s’est également donné pour objectif d’illustrer l’efficacité de ces technologies au travers de démonstrations de boucle système, que ce soit en laboratoire ou en orbite.

Les auteurs expriment leur gratitude envers les partenaires industriels et académiques du projet : Thales Alenia Space, Activeeon, MyDataModels, Geo4i, JoliBrain, et l’Université Côte d’Azur. Nous remercions également l’Agence Spatiale Européenne qui nous a permis de participer aux défis OrbitalAI.

Références

- [1] Marine anomaly detection presentation. <https://www.irt-saintexupery.com/fr/marine-anomaly-detection/>.
- [2] Rami Al-Ruzouq, Mohamed Barakat A Gibril, Abdallah Shanableh, Abubakir Kais, Osman Hamed, Saeed Al-Mansoori, and Mohamad Ali Khalil. Sensors, features, and machine learning for oil spill detection and monitoring : A review. *Remote Sensing*, 12(20) :3338, 2020.
- [3] Lauren Biermann, Daniel Clewley, Victor Martinez-Vicente, and Konstantinos Topouzelis. Finding plastic patches in coastal waters using optical satellite data. *Scientific Reports*, 10 :5364, 04 2020.
- [4] Aurélie Boisbunon, Carlo Fanara, Ingrid Grenet, Jonathan Daeden, Alexis Vighi, and Marc Schoenauer. Zoetrope genetic programming for regression. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 776–784, 2021.

- [5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [6] Hongtao Duan, Zhigang Cao, Ming Shen, Dong Liu, and Qitao Xiao. Detection of illicit sand mining and the associated environmental effects in china’s fourth largest freshwater lake using daytime and nighttime satellite images. *Science of the total environment*, 647 :606–618, 2019.
- [7] Diego Gómez, Pablo Salvador, Julia Sanz, and Jose Luis Casanova. A new approach to monitor water quality in the menor sea (spain) using satellite data and machine learning methods. *Environmental pollution*, 286 :117489, 2021.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
- [9] Xing Hu, Chun Xie, Zhe Fan, Qianqian Duan, Dawei Zhang, Linhua Jiang, Xian Wei, Danfeng Hong, Guoqiang Li, Xinhua Zeng, Wenming Chen, Dongfang Wu, and Jocelyn Chanussot. Hyperspectral anomaly detection using deep learning : A review. *Remote Sensing*, 14(9), 2022.
- [10] EB Jewett, CB Lopez, Q Dortch, SM Etheridge, and LC Backer. Harmful algal bloom management and response : Assessment and plan. 2008.
- [11] Morteza Khalili and Mahdi Hasanlou. Harmful algal blooms monitoring using sentinel-2 satellite images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2019.
- [12] Polychronis Kolokoussis and Vassilia Karathanassi. Oil spill detection and mapping using sentinel 2 imagery. *Journal of Marine Science and Engineering*, 6 :4, 2018.
- [13] Christos A Kontovas, Harilaos N Psaraftis, and Nikolaos P Ventikos. An empirical analysis of iopcf oil spill cost data. *Marine pollution bulletin*, 60(9) :1455–1466, 2010.
- [14] Huizeng Liu, Qingquan Li, Tiezhu Shi, Shuibo Hu, Guofeng Wu, and Qiming Zhou. Application of sentinel 2 msi images to retrieve suspended particulate matter concentrations in poyang lake. *Remote Sensing*, 9(7) :761, 2017.
- [15] Nikolai Maximenko, J Arvesen, G Asner, J Carlton, M Castrence, Luca Centurioni, Yi Chao, John Chapman, Ved Chirayath, P Corradi, et al. Remote sensing of marine debris to study dynamics, balances and trends. *White Paper, Decadal Survey for Earth Science and Applications from Space*, 22, 2016.
- [16] J. Mifdal, N. Longépé, and M. Rußwurm. Towards detecting floating objects on a global scale with learned spatial features using sentinel 2. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-3-2021 :285–293, 2021.
- [17] Dimitris Papageorgiou, Konstantinos N. Topouzelis, Giuseppe Suaria, Stefano Aliani, and Paolo Corradi. Sentinel-2 detection of floating marine litter targets with partial spectral unmixing and spectral comparison with other floating materials (plastic litter project 2021). *Remote. Sens.*, 14 :5997, 2022.
- [18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12 :2825–2830, 2011.
- [19] Miguel Potes, Gonçalo Rodrigues, Alexandra Marchã Penha, Maria Helena Novais, Maria João Costa, Rui Salgado, and Maria Manuela Morais. Use of sentinel 2–msi for water quality monitoring at alqueva reservoir, portugal. *Proceedings of the International Association of Hydrological Sciences*, 380 :73–79, 2018.
- [20] Oliver Rippel and Dorit Merhof. Anomaly detection for automated visual inspection : A review. *Bildverarbeitung in der Automation : Ausgewählte Beiträge des Jahreskolloquiums BVAu 2022*, pages 1–13, 2023.
- [21] Hongjun Su, Zhaoyue Wu, Huihui Zhang, and Qian Du. Hyperspectral anomaly detection : A survey. *IEEE Geoscience and Remote Sensing Magazine*, 10(1) :64–90, 2021.
- [22] thunderInfy. Simclr pytorch implementation. <https://github.com/thunderInfy/simclr>, 2013.
- [23] Kaire Toming, Tiit Kutser, Alo Laas, Margot Sepp, Birgot Paavel, and Tiina Nõges. First experiences in mapping lake water quality parameters with sentinel-2 msi imagery. *Remote Sensing*, 8(8) :640, 2016.
- [24] Konstantinos Topouzelis, Apostolos Papakonstantinou, and Shungudzemwoyo P. Garaba. Detection of floating plastics from satellite and unmanned aerial systems (plastic litter project 2018). *International Journal of Applied Earth Observation and Geoinformation*, 79 :175–183, 2019.
- [25] Yichu Xu, Lefei Zhang, Bo Du, and Liangpei Zhang. Hyperspectral anomaly detection based on machine learning : An overview. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15 :3351–3364, 2022.
- [26] Jie Yang, Ruijie Xu, Zhiquan Qi, and Yong Shi. Visual anomaly detection for images : A survey. *arXiv preprint arXiv :2109.13157*, 2021.
- [27] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection : A survey. *arXiv preprint arXiv :2110.11334*, 2021.