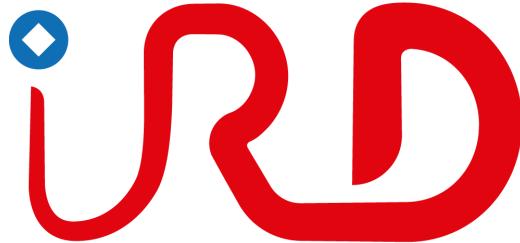


French National Research
Institute for Sustainable
Development



Institut de Recherche
pour le Développement

F R A N C E



Rapport de stage TN09

Apprentissage Non-Supervisé pour la Détection de Micro-Déforestation sur Image Satellite Radar

Étudiant : Yifan ZHANG - GI04

Institut : Institut de Recherche pour le Développement FRANCE

Encadrement Institut : M. Joris Guérin, M. Thibault Catry, Mme. Laure Berti-Equille

Encadrement UTC : Mme. Soundouss Messoudi

Semestre : A2024, Septembre 2024 - Février 2025

Résumé technique

Ce rapport résume les travaux réalisés dans le Projet "M2D2: Machine learning-based Micro Déforestation Détection", financé par le Programme National de Télédétection Spatiale (PNTS), à l'Institut de Recherche pour le Développement (IRD) dans l'unité de recherche ESPACE-DEV à Montpellier, en France.

L'objectif principal était de créer un modèle d'auto-encodeur non supervisé capable de compresser et de reconstruire des images SAR de forêt capturées par S1, et d'utiliser ce modèle pour détecter des anomalies dans les images et ainsi suivre les changements du couvert forestier à petite échelle. Ces changements sont considérés comme des "micro-déforestations", que nous cherchons à détecter en comparant des images prises à des moments différents. Nous utilisons ce modèle pour détecter les différences entre deux images prises à des moments différents. Cela représente une solution efficace pour surveiller la déforestation.

Le rapport présente en détail le processus de conception et d'entraînement du modèle, ainsi que les principaux défis techniques rencontrés et les solutions apportées. Les premiers résultats indiquent un potentiel pour identifier les zones où la forêt subit des transformations. De plus, nous mettons en avant des pistes d'amélioration pour optimiser la mise en œuvre actuelle du modèle et renforcer ses performances.

Mots-clés : Images SAR, Détection de la déforestation, Apprentissage profond, auto-encodeur, Python

Remerciements

Tout d'abord, je tiens à exprimer ma gratitude à mes trois encadrants pour m'avoir offert cette précieuse opportunité de stage.

Pendant mon stage, Monsieur Joris Guérin m'a fourni des conseils importants sur l'apprentissage profond et a pris le temps de répondre à mes questions sur le projet. Monsieur Thibault Catry m'a donné de nombreuses ressources pour comprendre les images SAR et a répondu rapidement à mes diverses questions pendant mon apprentissage. De plus, Madame Laure Berti m'a guidé dans la normalisation des processus du projet, en particulier pour l'utilisation des outils d'automatisation pour l'amélioration des hyperparamètres. Je tiens également à les remercier pour leurs précieux conseils concernant la rédaction et l'amélioration de mon rapport, qui m'ont permis de mieux structurer mes idées et de clarifier mes analyses.

Ensuite, je souhaite remercier Madame Nadia Aouri, pour avoir facilité mon intégration dans l'environnement de l'institut et m'avoir soutenu dans les tâches administratives.

Je remercie Madame Baoxia Chen et Madame Maryam Ly pour leur soutien lors de ma recherche de stage et leur aide rapide face aux imprévus. De plus, mon encadrante pour le stage, Madame Soundouss Messoudi, est restée en contact avec moi par email tout au long de mon stage pour s'assurer du bon déroulement.

Je veux remercier ma famille et mes amis pour leur soutien et leurs conseils pendant mon stage. Ils m'ont aidé à surmonter les difficultés.

Je remercie aussi tous les employés de l'UMR Espace-Dev pour leur accueil et leur gentillesse pendant ces six mois. Grâce à vous, mon stage a été très enrichissant et inoubliable.

Encore merci à tous mes encadrants, collègues, famille et amis qui m'ont soutenu pendant mon stage !

Sommaire

Résumé technique.....	2
Remerciements.....	3
1. Introduction.....	5
2. Présentation de l'Institut et de l'équipe.....	7
3. Description du projet.....	8
Objectifs.....	8
4. État de l'art.....	9
4.1 Comment les séries temporelles Sentinel-1 peuvent améliorer la mise en œuvre des programmes de conservation au Brésil.....	9
4.2 Détection de la déforestation à l'aide de techniques de segmentation sémantique basées sur l'apprentissage profond : une revue systématique.....	11
5. Données.....	13
6. Méthode.....	15
6.1 Aperçu de la proposition.....	16
6.2 Annotation manuelle des données.....	16
6.3 Prétraitement des données.....	18
6.4 Conception et entraînement de l'auto-encodeur.....	21
6.5 Détection des anomalies pour identifier les zones déforestées.....	24
6.6 Défis rencontrés et solutions.....	27
6.7 Gestion des versions du code.....	31
7. Résultats.....	34
7.1 Analyse qualitative des résultats obtenus.....	34
7.2 Analyse quantitative des résultats obtenus.....	35
8. Discussion.....	38
8.1 Contributions Méthodologiques et Performances.....	38
8.2 Innovations et Perspectives Futures.....	38
8.3 Limites et Améliorations Potentielles.....	39
9. Conclusion.....	40
10. Bibliographie.....	41
11. Glossaire.....	43
12. Annexes.....	44
12.1 Annexe 1: Dépôt GitHub de "SAR Deforestation Detection autoencoder".....	44
12.2 Annexe 2: Liste des Figures.....	45

1. Introduction

Dans le cadre d'un projet de recherche avec l'Institut de Recherche pour le Développement (IRD) et l'UMR Espace-Dev, les scientifiques cherchent à résoudre des problèmes clés liés à la protection des écosystèmes et au développement durable en utilisant des données spatiales et des technologies comme la télédétection et la modélisation. Des régions amazoniennes aux zones rurales et urbaines d'Europe et de Chine, ces équipes de recherche ne se limitent pas à étudier les tendances environnementales à grande échelle. Elles analysent également les impacts des changements à petite échelle sur l'ensemble des écosystèmes et des populations locales. Leur travail apporte, au travers de différents projets, des perspectives et des solutions pour préserver la diversité des écosystèmes de notre planète.

Le stage s'inscrit dans la ligne de plusieurs projets de l'unité dans laquelle j'ai fait mon stage:

Le projet F4F (Forest for Food) a pour objectif de lier la sécurité alimentaire et la protection des forêts en utilisant des contrats innovants de paiement pour services écosystémiques (PSE), notamment dans la région amazonienne du Brésil. Il vise les petits agriculteurs en leur offrant des motivations pour changer leurs méthodes d'utilisation des terres et réduire la déforestation. En combinant des technologies de télédétection et des approches économiques, ce projet évalue l'efficacité à long terme des politiques de conservation, fournissant ainsi des données essentielles pour élaborer des stratégies de protection plus efficaces. Ce projet utilise l'algorithme Cumulative Sum (CuSum) sur des séries d'images Sentinel-1. Cet outil, décrit par [Pfefer et al., 2024], permet de détecter la déforestation, ce qui aide à évaluer l'efficacité des politiques de conservation sur le long terme.

D'autre part, le projet Eco2Adapt se concentre sur le renforcement de la résilience écologique et sociale des forêts en Europe et en Chine. À travers une recherche collaborative et des outils techniques, ce projet propose des modèles de gestion et des méthodes économiques innovantes pour aider les propriétaires forestiers et les décideurs à surveiller la vulnérabilité et la capacité d'adaptation des forêts. Ces projets non seulement influencent les politiques mondiales de protection des forêts, mais montrent aussi comment la technologie et la coopération communautaire peuvent répondre aux défis du changement climatique.

Cependant, malgré les progrès importants réalisés dans ces projets, la surveillance des déforestations à petite échelle est encore confrontée à plusieurs limitations. Les outils actuels, tels que l'algorithme CuSum, bien qu'efficaces à l'échelle locale, présentent plusieurs limites. Les retards dans la détection, causés par le délai entre les changements de structure végétative et leur impact sur les signaux radar, réduisent leur utilité pour une alerte précoce en temps réel. De plus, le volume important des images et le temps nécessaire à leur traitement limitent l'applicabilité à de vastes régions, tandis que les erreurs de détection affectent la précision globale des évaluations. Ces limites soulignent la nécessité de développer de nouvelles solutions.

Le projet M2D2 (Machine learning-based Micro Déforestation Détection) [M2D2, 2024], qui finance ce stage, a pour objectif de répondre à ces enjeux en explorant des approches novatrices basées sur l'intelligence artificielle. L'objectif principal est de concevoir un modèle d'apprentissage automatique non supervisé capable de détecter automatiquement les changements forestiers à petite

échelle, sans nécessiter de grandes quantités de données annotées. Mon projet de recherche s'inscrit dans ce cadre en développant un modèle de détection d'anomalies basé sur un auto-encodeur appliqué aux images radar Sentinel-1. Ce modèle analysera les changements dans la structure des zones forestières et détectera précisément les déforestations à petite échelle grâce à l'analyse des erreurs de reconstruction. M2D2 vise à dépasser les limitations des approches actuelles afin de fournir des systèmes de détection plus rapides, précis et adaptés à des zones étendues. Cette avancée pourrait permettre une télédétection en temps réel pour améliorer la protection des écosystèmes.

2. Présentation de l'Institut et de l'équipe

L'Institut de Recherche pour le Développement (IRD), créé en 1943, est un organisme français qui mène des recherches dans de nombreux domaines pour soutenir les pays en développement. Ses activités couvrent des disciplines variées, comme les sciences naturelles, les sciences sociales et les technologies de l'ingénierie. L'objectif principal de l'IRD est d'encourager un développement durable à l'échelle mondiale grâce à la recherche scientifique et à l'innovation technologique. En tant qu'institution clé de la recherche en France, l'IRD collabore avec de nombreux centres et partenaires situés dans différentes régions du monde. L'institut s'efforce activement de proposer des solutions aux défis complexes auxquels les pays en développement doivent faire face.

L'UMR Espace-Dev, unité mixte de recherche associée à l'IRD, se concentre sur l'étude des transformations des territoires et des systèmes socio-environnementaux. Basée sur une approche interdisciplinaire, elle développe des outils et des modèles innovants pour soutenir la gestion durable des ressources naturelles et l'amélioration des conditions de vie dans les pays du Sud.

L'UMR Espace-Dev est hébergée à la Maison de la Télédétection, ce qui favorise les collaborations entre chercheurs et innovateurs. La Maison de la Télédétection à Montpellier est un centre de recherche et d'innovation de premier plan spécialisé dans les technologies de télédétection. Le centre se distingue par son expertise dans l'utilisation de la télédétection et des données spatiales pour l'étude de systèmes environnementaux complexes. Il offre un environnement collaboratif réunissant des scientifiques, des ingénieurs et des professionnels du secteur pour améliorer la compréhension et l'utilisation des données d'observation de la terre ou géospatiales dans le suivi de l'utilisation des terres, de la déforestation et de la gestion des ressources naturelles. Le centre participe également à des initiatives éducatives afin de promouvoir une approche multidisciplinaire pour l'application des technologies de télédétection au développement durable.

Dans le cadre de ce stage, l'équipe de projet était composée de mes trois encadrants et de moi-même. Mes encadrants m'ont fourni des conseils et un soutien professionnel dans les domaines de l'apprentissage profond, des technologies de télédétection et de l'analyse des données, ce qui a assuré la bonne progression du projet.

3. Description du projet

Ce projet se concentre sur l'utilisation des images SAR pour détecter les changements liés à la déforestation. Les travaux prévus durant ce stage mobilisent des connaissances sur le traitement d'images radar, l'apprentissage profond et l'analyse de données.

Les méthodes traditionnelles basées sur les images optiques haute résolution sont souvent affectées par des conditions climatiques et saisonnières. En revanche, le signal radar est peu affecté par les conditions atmosphériques. Cela permet d'acquérir des images en continu, même en présence de nuages. Ces caractéristiques sont particulièrement efficaces pour surveiller des régions telles que les forêts tropicales, souvent nuageuses et humides. Dans le cadre de la détection de la déforestation, les images SAR peuvent identifier précisément les changements de couverture forestière. Elles utilisent les canaux de polarisation VV et VH pour détecter les caractéristiques de rétrodiffusion des différents éléments au sol, ce qui facilite la détection des changements du couvert forestier. [MM Rahman et al., 2010][A Bouvet et al., 2018].

Un auto-encodeur [Bank et al., 2020] est un modèle d'apprentissage automatique non supervisé, conçu pour apprendre une représentation simplifiée des données d'entrée. Il se compose de deux parties : un encodeur qui compresse les données et un décodeur qui restaure les données d'origine à partir de cette compression. Ce modèle est largement utilisé pour analyser des données complexes en identifiant des motifs réguliers. Il peut également être utilisé pour détecter des anomalies à travers les erreurs de reconstruction, comme l'ont montré [Shvetsova et al., 2021] dans leur étude sur la détection d'anomalies dans les images médicales à l'aide d'auto-encodeurs perceptifs profonds. En raison de sa capacité à traiter de grandes quantités de données non étiquetées, l'auto-encodeur est souvent appliqué à divers domaines, tels que la détection d'anomalies dans les images, les séries temporelles ou encore les données textuelles.

Objectifs

Les travaux menés au cours de ce stage avaient plusieurs objectifs:

1. Créer et améliorer une base de données d'images SAR annotées, afin d'évaluer l'efficacité des méthodes de détection de la déforestation.
2. Développer un modèle d'apprentissage non-supervisé pour la détection des micro-déforestations à l'aide des images radar Sentinel-1
3. Valider des méthodes développées sur des zones d'essai et les comparer avec les résultats obtenus par d'autres méthodes

4. État de l'art

Dans le cadre de mon stage, j'ai analysé plusieurs articles en lien avec le sujet du projet. Parmi ceux-ci, deux articles, [Pfefer et al., 2024] et [Md Jelas et al., 2024], se sont révélés particulièrement pertinents et utiles pour mieux comprendre le contexte du projet. Ces travaux m'ont permis de mieux cerner les méthodes utilisées, notamment l'utilisation de modèles d'apprentissage profond pour la détection de la déforestation, ainsi que les problèmes techniques et théoriques associés à l'application de l'algorithme CuSum [Ygorra et al., 2021] dans ce domaine. Ces études illustrent les progrès méthodologiques et les défis techniques récents dans la détection de la déforestation.

4.1 Comment les séries temporelles Sentinel-1 peuvent améliorer la mise en œuvre des programmes de conservation au Brésil

La déforestation menace la nature, le climat et la vie des communautés locales. Les petits exploitants jouent un rôle significatif dans ce problème en Amazonie brésilienne. Pour résoudre ce problème, des programmes de conservation comme le paiement pour services écosystémiques (PES) mettent en place des mécanismes de compensation financière encourageant les petits exploitants à protéger les forêts. L'étude de [Pfefer et al., 2024] a pour objectif de développer une méthode de surveillance précise et efficace en utilisant les séries temporelles des images SAR capturées par Sentinel-1 et l'algorithme CuSum, tout en minimisant les erreurs de détection.

La zone d'étude se situait dans l'État de l'Acre, au Brésil, couvrant deux sites de 176 000 hectares. Ces régions, caractérisées par une alternance de saisons sèches et pluvieuses, ont fourni un environnement idéal pour tester cette méthode. Les données comprenaient des images SAR capturées par Sentinel-1 (10 m de résolution), des images optiques capturées par PlanetScope (5 m de résolution) pour validation, et un masque de forêt (Tropical Moist Forest) [TMFv2023] pour exclure les zones non forestières.

L'algorithme CuSum [Ygorra et al., 2021] se compose de plusieurs étapes : un prétraitement des données Sentinel-1, une détection basée sur les variations des séries temporelles de rétrodiffusion, et une post-traitement pour réduire les erreurs de détection en appliquant des masques et des seuils de taille. Cet algorithme fonctionne en suivant l'évolution du signal radar au cours du temps. Il calcule les différences entre le signal actuel et une moyenne calculée sur une période. Lorsqu'un changement important est détecté dans l'intensité de ce signal, cela peut indiquer une déforestation. Si ces différences s'accumulent au-delà d'un seuil défini, une alerte est déclenchée. La validation a été faite avec des cartes de référence obtenues à partir des images PlanetScope, ce qui a permis d'évaluer les performances à l'aide de métriques comme la précision, le rappel et l'exactitude globale.

CuSum atteignait une exactitude globale de 98,9 % et une précision de 87,3 %, traduisant une bonne gestion des erreurs de détection. Son rappel, de 53,2 %, reflète une sensibilité moyenne aux événements de déforestation (figure 1). Le délai de détection était de 48 à 60 jours, mais des ajustements pourraient réduire ce délai. Par rapport à d'autres outils comme GFW ou Prodes, CuSum s'est distingué par sa précision supérieure, bien que son rappel reste modéré. De plus, le programme

PES a réduit la déforestation des participants de 40 %, prouvant son impact positif [Demarchi et al., 2023].

Cependant, l'étude a identifié certaines limites, notamment des erreurs de détection liées à la délimitation des zones forestières et aux variations naturelles de la rétrodiffusion, ainsi qu'un délai de détection de 48 jours pour identifier certains événements de déforestation. Pour améliorer la situation, il serait possible d'utiliser des algorithmes pour détecter les ombres radar et améliorer la qualité des masques d'entrée. De plus, la perte des données Sentinel-1B suite à une défaillance du satellite a légèrement réduit les performances, bien que l'utilisation des données de Sentinel-1A ait fourni des résultats favorables.

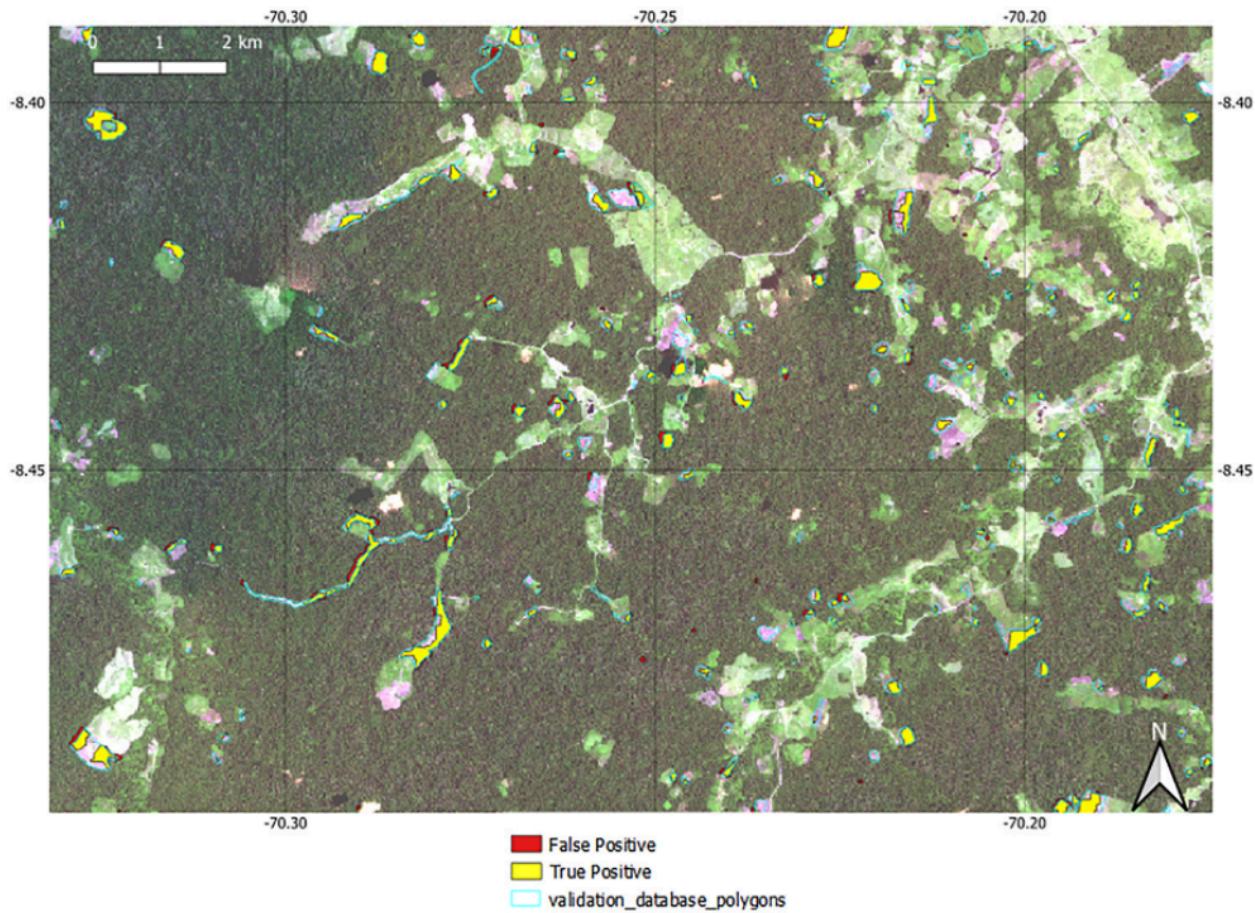


Figure 1: Résultats de l'algorithme CuSum sur un échantillon de la zone étendue de l'image PlanetScope d'octobre 2021

En conclusion, l'algorithme CuSum présente un équilibre efficace entre précision et capacité de détection, ce qui en fait un outil précieux pour les programmes de conservation. Sa flexibilité permet d'adapter les paramètres aux besoins spécifiques, ce qui répond aux défis clés de la surveillance de la déforestation. Des améliorations futures pourraient se concentrer sur la réduction des délais de détection et l'amélioration de la qualité des données d'entrée.

4.2 Détection de la déforestation à l'aide de techniques de segmentation sémantique basées sur l'apprentissage profond : une revue systématique

Les travaux de [Md Jelas, I., 2024] offrent une analyse approfondie des approches modernes permettant de détecter la déforestation grâce à l'intégration de techniques de segmentation sémantique basées sur l'apprentissage profond et des images satellites. En exploitant des modèles comme U-Net, DeepLabV3 et ResNet, cet article explique comment ces technologies améliorent la précision et l'efficacité de la détection des changements du couvert forestier, tout en mettant en évidence les méthodes d'amélioration comme l'apprentissage multi-échelles et les mécanismes d'attention.

Pour réaliser cette étude, les auteurs ont adopté une approche systématique basée sur la méthodologie PRISMA [BMJ 2021]. Ils ont analysé 22 articles publiés entre 2018 et 2022, en se concentrant sur les techniques d'apprentissage profond appliquées à la classification et à la segmentation de la déforestation. Les bases de données Scopus et Web of Science ont servi de sources principales, et chaque étape, de l'identification au filtrage, a été soigneusement documentée pour garantir la qualité et la pertinence des travaux inclus.

Les résultats mettent en lumière la performance impressionnante de certains modèles. Par exemple, l'architecture U-Net, utilisant des mécanismes d'attention, a obtenu des scores F1 supérieurs à 95 % dans plusieurs cas. Ces résultats sont renforcés par des processus de prétraitement, par exemple l'annotation des données, l'augmentation et la division stratégique des jeux de données d'entraînement et de test.

Pour entraîner ces modèles, les chercheurs utilisent des jeux de données variés et adaptés aux contextes géographiques, comme les images multispectrales des satellites Sentinel-2 et Landsat-8, dont la résolution spatiale varie de 10 à 30 mètres. Ces données sont largement exploitées pour analyser des régions comme la forêt amazonienne ou les zones forestières en Afrique et en Asie. Des bases de données ouvertes comme PRODES et DETER pour l'Amazonie ou les images SentinelHub pour l'Atlantique sont également exploitées. Ces jeux de données sont complétés par des images de drones pour des analyses plus détaillées, ainsi que par des annotations manuelles et automatiques. Ces ressources permettent aux chercheurs d'atteindre une précision élevée, bien que la disponibilité des données annotées reste un défi.

Néanmoins, certains défis persistent. Les modèles ont du mal à généraliser leurs résultats aux environnements forestiers complexes et hétérogènes. De plus, la production de grandes quantités de données annotées reste un processus long et coûteux. Toutefois, l'amélioration continue des techniques et l'accessibilité croissante des données permettent d'envisager un avenir prometteur pour la surveillance de l'environnement.

En conclusion, cet article met en évidence le potentiel des techniques de segmentation sémantique basées sur l'apprentissage profond pour une détection précise et efficace de la déforestation à grande échelle. Les auteurs insistent sur la nécessité de poursuivre les recherches afin d'améliorer la généralisation des modèles et de réduire les coûts associés à l'acquisition de données.

C'est précisément dans cette optique que s'inscrit mon stage, qui vise à développer des modèles plus robustes et adaptés à des environnements complexes, tout en explorant des approches pour réduire la dépendance aux annotations coûteuses. Ces travaux visent à rendre les solutions de détection de la déforestation plus accessibles et efficaces pour une gestion durable des forêts.

5. Données

Images SAR

Les images SAR de Sentinel-1 ([figure 2](#)) permettent d'obtenir des informations fiables sur les surfaces terrestres, quelle que soit la météo ou la luminosité. Elles sont utilisées pour détecter la déforestation en étudiant les variations des canaux de polarisation VV et VH.

Ces images, disponibles gratuitement grâce à l'Agence spatiale européenne (ESA), sont téléchargées depuis le portail de l'Alaska Satellite Facility (ASF). Avec une résolution de 10 mètres et une couverture globale tous les 12 jours, elles sont fournies au format GRD (Ground Range Detected) et capturées en mode IW (Interferometric Wide Swath). Leur fiabilité et leur robustesse face aux conditions climatiques en font un outil précieux pour le suivi de la déforestation et d'autres changements environnementaux à long terme.

Images optiques

Les images optiques PlanetScope ([figure 3](#)) permettent de visualiser clairement les différences entre les zones boisées et déforestées. Les changements de couleur indiquent différentes phases de déforestation ou de reforestation. Ces images sont téléchargées depuis la plateforme Norway's International Climate and Forests Initiative (NICFI), qui fournit des données optiques accessibles pour soutenir les études environnementales. Elles offrent une résolution spatiale d'environ 3 à 5 mètres et une fréquence d'acquisition mensuelle.

Dans mon travail, j'utilise ces images pour annoter avec précision les zones de déforestation, en identifiant et en classifiant les changements de couvert forestier. Cependant, elles présentent des limites dans la détection systématique de la déforestation. Leur utilisation est fortement affectée par les conditions climatiques, comme la couverture nuageuse fréquente dans les régions tropicales, rendant l'observation moins régulière et fiable. Ces contraintes soulignent l'intérêt d'utiliser les images SAR pour compléter l'analyse, car elles ne sont pas affectées par ces conditions.

Masque de forêt

Ce masque a été développé dans le cadre du projet Tropical Moist Forests v2023 [[TMFv2023](#)], qui vise à surveiller les évolutions des forêts tropicales humides à l'échelle mondiale ([figure 6](#)). Il permet d'identifier les zones boisées et sert de référence pour valider les zones détectées comme déforestées dans les données SAR et optiques.

Dans le projet TMF, une forêt est définie comme une zone avec au moins 30 % de couverture arborée et des arbres d'une hauteur moyenne supérieure à 5 mètres. Les masques sont utilisés pour comparer les zones boisées entre différentes périodes et valider les résultats des algorithmes. Cependant, ils ne suffisent pas pour détecter les changements subtils ou les petites déforestations, leur précision temporelle limitée empêchant d'analyser les causes des changements observés.

Sorties CuSum pour la validation

Les résultats CuSum [Pfefer et al., 2024] permettent de détecter les anomalies dans les données SAR en utilisant une analyse statistique (figures 1 et 4). Ces sorties consistent en des polygones identifiant les changements entre deux dates précises, permettant ainsi de localiser les zones d'évolution significative dans les surfaces boisées. Les résultats couvrent une période d'observation étendue, et nous comparons ces résultats avec ceux obtenus par notre approche basée sur les auto-encodeurs.

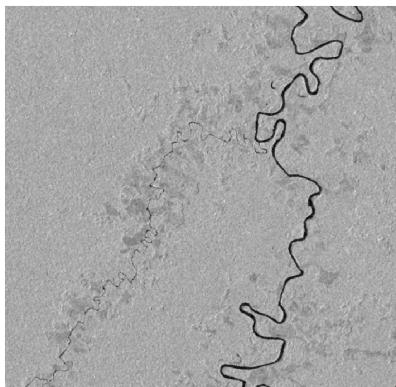


Figure 2: Exemple d'Image SAR

Sentinel-1 2021.4

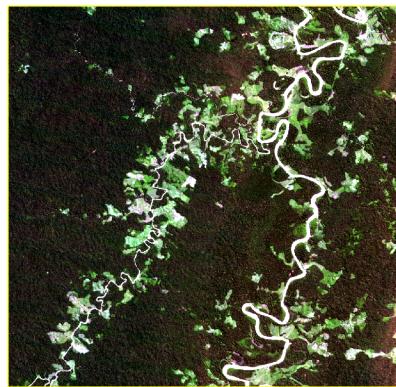


Figure 3: Exemple d'Image Optique

PlanetScope 2021.4

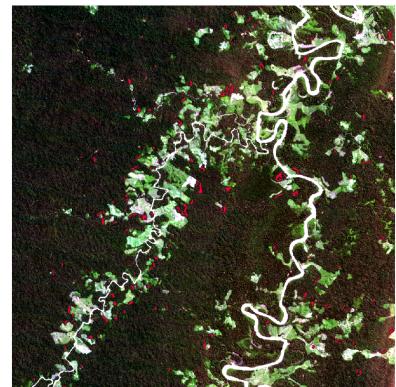


Figure 4: Exemple d'Sorties Cusum

2020.10.11 - 2021.12.17

Annotations des images optiques

Les annotations (figure 7) comprennent des polygones qui montrent les zones déforestées, créés à la main avec QGIS. Ces données servent à vérifier les résultats de l'entraînement du modèle.

6. Méthode



Figure 5: Organigramme du processus de développement du projet

6.1 Aperçu de la proposition

L'approche proposée repose sur l'utilisation d'un modèle d'auto-encodeur conçu pour analyser les images SAR des forêts. Cet auto-encodeur est entraîné à compresser et reconstruire les caractéristiques des zones forestières intactes. L'idée principale est que le modèle apprend à reconstruire précisément les images des forêts qui n'ont pas subi de changements, tandis que les zones déforestées ou modifiées génèrent des erreurs de reconstruction significatives. Ces erreurs deviennent ainsi des signaux clairs des changements environnementaux, tels que la déforestation.

Ce processus se divise en plusieurs étapes :

1. Le modèle est d'abord entraîné sur des images SAR représentant des zones de forêt intacte, afin qu'il puisse comprendre les structures récurrentes et les textures typiques de ces régions.
2. Une fois entraîné, le modèle est utilisé pour analyser de nouvelles images SAR, où il tente de reconstruire les zones forestières. Les différences entre l'image originale et l'image reconstruite (erreurs de reconstruction) montrent les zones ayant subi des changements, comme la déforestation ou d'autres changements.
3. Les erreurs de reconstruction sont ensuite analysées et comparées à des annotations manuelles pour valider les performances du modèle.

6.2 Annotation manuelle des données

Dans la première étape de mon stage, j'ai passé environ 2 à 3 semaines à m'informer sur les images SAR. Cela m'a aidé à comprendre les bases des images radar et leur utilité pour détecter la déforestation. Ensuite, j'ai choisi deux zones dans la forêt amazonienne sur le site NICFI, où il y avait des traces de déforestation. Pour chacune de ces zones, j'ai téléchargé des images optiques prises par PlanetScope, entre septembre 2020 et septembre 2022, avec une image par mois.

6.2.1 Annotation

Dans cette région, il y a une saison sèche et une saison des pluies. De octobre à mars, les images prises peuvent être cachées par des nuages, ce qui complique la détection des changements. C'est pourquoi je me suis concentré sur les images **d'avril à septembre pour les années 2021 et 2022**. De plus, il y a peu de déforestation significative pendant la période exclue, ce qui limite l'intérêt de l'analyse pour ces mois.

Pour réaliser l'annotation, j'ai utilisé l'outil des polygones dans QGIS pour identifier et tracer les zones où des changements visibles dans les forêts étaient présents. Les polygones sont créés en observant les différences de couleur, de texture et de structure observées dans les images optiques. Par exemple, les zones déforestées apparaissent souvent avec une couleur plus claire ou des motifs irréguliers, contrastant avec les zones intactes de forêt. Une fois les zones suspectes repérées, je trace manuellement des polygones pour délimiter précisément les contours des changements détectés ([figure 7](#)).

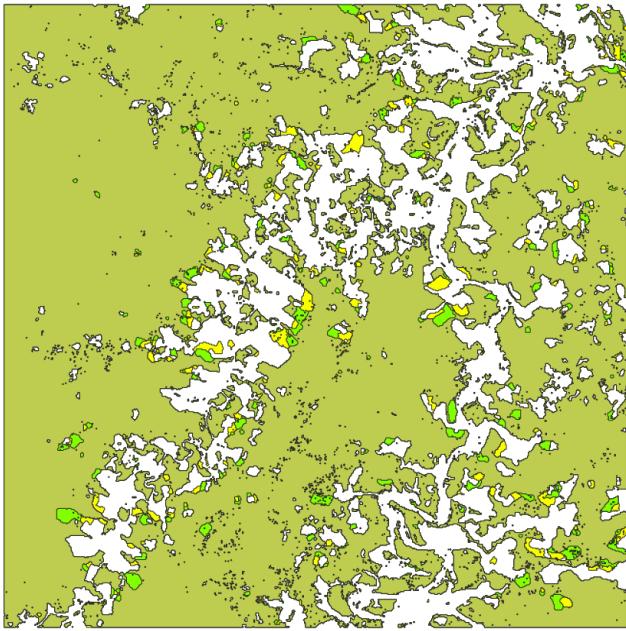


Figure 6: Masque de forêt/non forêt

vert-jeune: 2022, vert: 2021, jeune: 2020, blanc: non forêt

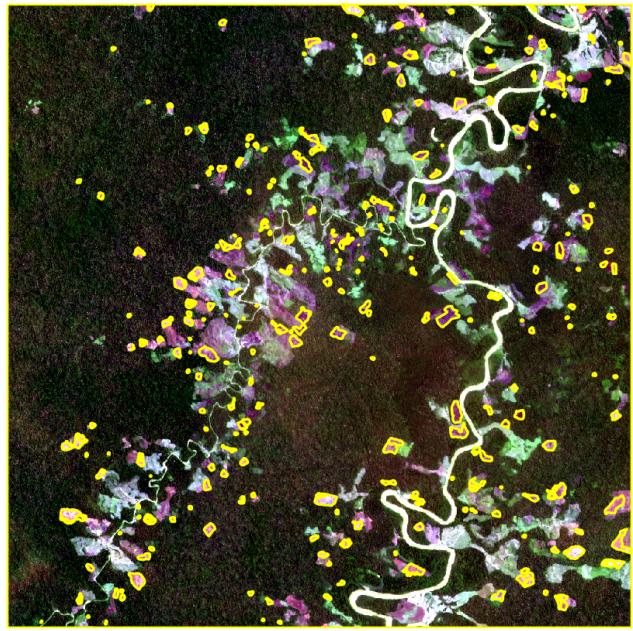


Figure 7: Annotation manuelle

PlanetScope 2022.4 - 2022.10

Pour vérifier la qualité des annotations, j'ai également utilisé la différence entre les deux masques de forêt du projet open-source Tropical Moist Forests [TMFv2023]. J'ai calculé les métriques entre la différence raster et mes annotations afin d'examiner la précision de mes travaux initiaux. Finalement, j'ai choisi la zone numéro 622_975 parmi les deux zones présélectionnées, car l'autre zone était surtout composée de terres agricoles et présentait peu de changements forestiers.

6.2.2 Utilisation

Ce jeu de données annoté sert de référence pour évaluer l'efficacité du modèle de détection de déforestation que je crée pendant mon stage. Les annotations manuelles fournissent des données de référence que nous utiliserons pour comparer les résultats du modèle. Cela permet d'évaluer précisément la capacité du modèle à détecter les zones déforestées.

6.2.3 Partage

J'ai créé un répertoire contenant les fichiers de zones géographiques sur mon dépôt GitHub [Annexe 1]. Ces annotations, associées aux images optiques et SAR du projet, indiquent les zones identifiées comme déforestées. Pour mettre en relation les images SAR Sentinel-1, j'ai sélectionné des images SAR correspondant à la même région et aux mêmes périodes que celles utilisées pour créer les annotations optiques. Chaque shapefile est lié à une image spécifique grâce aux métadonnées (nom, date, etc.). Cette démarche garantit une cohérence géographique et temporelle entre les deux types de données. Pour les exploiter, des bibliothèques comme [Geopandas](#) ou [Rasterio](#) peuvent

être utilisées en Python, permettant de visualiser les polygones et de les superposer aux images satellites pour valider des modèles ou entraîner des algorithmes.

6.3 Prétraitement des données

6.3.1 Répartition des données

Pour la répartition des données entre les ensembles d'entraînement, de validation et de test, j'ai utilisé des images optiques prises par le satellite Planetscope. J'ai choisi quatre zones comme ensemble d'entraînement (No. 622_971, No. 623_972, No. 623_973, No. 624_972)(figure 8), une zone comme ensemble de validation (No. 621_970) et une autre zone comme ensemble de test (No. 622_975). [Ces numéros sont les numéros de zone des images téléchargées sur le NICFI](figure 9)

Les raisons du choix des cinq premières zones comme ensembles d'entraînement et de validation sont liées à leur couverture forestière dense. Ces zones permettent à l'auto-encodeur d'apprendre les caractéristiques essentielles des « forêts » à partir d'exemples suffisants et purs. La dernière zone a été sélectionnée spécifiquement dans la forêt amazonienne pour ses signes marquants de déforestation, ce qui en fait un choix idéal pour tester les capacités de détection du modèle.

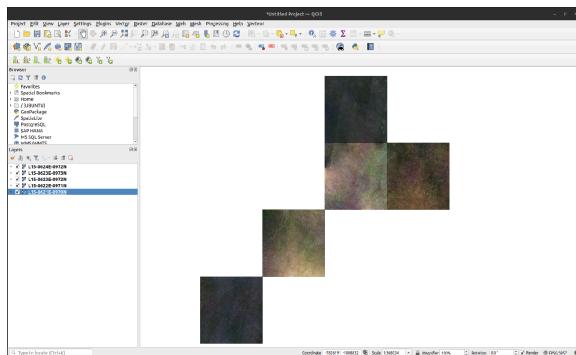


Figure 8: images optiques dans QGIS



Figure 9: téléchargement de NICFI

Pour l'ensemble d'entraînement, j'ai sélectionné des images prises entre **le 1er mai et le 1er octobre de chaque année de 2018 à 2024**. L'ensemble de validation suit la même période, entre **le 1er mai et le 1er octobre de 2018 à 2024**. Quant à l'ensemble de tests, il est basé sur les annotations manuelles que j'ai réalisées à partir des images Planetscope, qui couvrent les mois **de mai à octobre pour les années 2021 et 2022**. Ainsi, l'ensemble de test utilise des images de la même période pour assurer la cohérence des données.

Le choix de la période de mai à octobre repose sur le fait qu'elle correspond à la saison sèche en Amazonie, une période où la déforestation est plus fréquente. En outre, la saison sèche présente moins de nuages, ce qui rend les annotations manuelles sur les images optiques plus précises. Ces images incluent de nombreuses variations saisonnières et annuelles, apportant une précieuse diversité aux exemples utilisés pour l'entraînement et le test du modèle.

Après le téléchargement et le prétraitement des données, nous avons 404 images SAR (2 canaux VV et VH) pour l'ensemble d'entraînement, 122 images pour l'ensemble de validation et 31 images pour l'ensemble de test dans les trois ensembles de données. Toutes les images ont une taille approximative de **2 × 2000 × 2000** pixels. Après avoir utilisé un programme écrit avec Rasterio pour le prétraitement, la taille finale de l'ensemble de données est la suivante : l'ensemble d'entraînement comprend environ **32 000** images, l'ensemble de validation en contient environ **8 000**, et l'ensemble de test environ **1 600** images. Toutes les images ont une taille uniforme de **2 × 256 × 256** pixels.

6.3.2 Prétraitement des données

Pour le prétraitement des données, j'ai fait référence à la fonction de prétraitement du programme d'algorithme CuSum précédemment réalisé à l'IRD pour la détection de la déforestation, et j'ai ajouté quelques étapes supplémentaires. Plus précisément, j'ai combiné les images des canaux VV et VH prises au même moment et au même endroit. Ensuite, je les ai découpées en petites zones de taille uniforme pour faciliter leur analyse. J'ai également éliminé les zones inutilisables, comme celles contenant des valeurs 0 ou des valeurs manquantes ou invalides, afin d'améliorer la qualité des données. Enfin, j'ai ajusté les images SAR originales d'ASF à la taille 2 (canaux) × 256 × 256 pixels conformément aux exigences de mon modèle d'auto-encodeur.

Fonction preprocess

L'objectif de cette étape est de réduire le bruit présent dans les données et d'améliorer les signaux de changement de couverture forestière, afin d'accroître la précision de détection du modèle.

La fonction preprocess comprend principalement les étapes suivantes :

Mode de mise à l'échelle : Il est possible de choisir une échelle linéaire ou en décibels (dB) pour la sortie. J'ai sélectionné l'échelle en dB afin de renforcer le contraste des changements de couverture forestière.

Choix des canaux de polarisation : Les images SAR comportent plusieurs canaux de polarisation, tels que VV et VH.

Suppression du bruit de bordure : Pour les données Sentinel-1, j'ai sélectionné la méthode recommandée par l'ESA pour supprimer le bruit de bordure, en utilisant la bibliothèque [[pyroSAR](#)], ce qui diminue les interférences non nécessaires dans l'image.

Correction du relief et utilisation d'un modèle numérique d'élévation : J'ai sélectionné le SRTM 1Sec HGT comme modèle numérique d'élévation (DEM) et appliqué une interpolation bilinéaire dans la correction du relief pour compenser l'impact du terrain sur les images SAR.

Filtrage du bruit granuleux : Le bruit de type "speckle" est courant dans les images SAR et nuit à leur qualité. J'ai sélectionné un filtre de réduction du bruit Lee Sigma, améliorant ainsi leur clarté et leur netteté.

J'ai ajusté à plusieurs reprises les paramètres de prétraitement, en expérimentant différentes combinaisons de méthodes de réduction du bruit de bordure, filtres de bruit granuleux, DEM pour analyser leur influence sur les images SAR prétraitées. Ces observations m'ont permis de choisir les paramètres optimaux pour garantir des données de qualité adaptées à mon modèle d'auto-encodeur.

6.3.3 Réalisation de la classe Dataset et DatasetLoader

Création du jeu de données personnalisé

La classe de jeu de données personnalisée `ProcessedForestDataset` a été développée en tant qu' instance de la classe Dataset de PyTorch. Elle est conçue pour charger les images SAR prétraitées, qui sont des blocs d'images TIFF à 2 canaux. Chaque image est traitée de la manière suivante :

Préparation des données : Les images SAR brutes sont découpées en blocs de taille $2 \times 256 \times 256$, afin de faciliter l'entraînement du modèle d'apprentissage profond.

Formatage des données : Lors du chargement des données, chaque image est mise au format (C, H, W), où C=2 représente les deux canaux de polarisation de l'image, H représente la hauteur et W la largeur de l'image, toutes deux de 256 pixels.

Normalisation : Après leur chargement, en utilisant les valeurs des images du jeu d'entraînement, dont les pixels se situent dans la plage approximative de [-15, -3]. Cette normalisation globale est appliquée aux trois ensembles de données (entraînement, validation, test) pour garantir que les caractéristiques forestières soient correctement réparties entre [0, 1]. [[explication dans 6.5.2](#)]

6.3.4 Amélioration du traitement des jeux de données

Lors des premières phases de chargement des données, j'ai essayé de découper dynamiquement les images SAR pendant le chargement. Cette méthode permettait de réduire l'espace de stockage nécessaire, mais elle a considérablement diminué l'efficacité de l'entraînement. En effet, chaque chargement implique de découper à nouveau les images brutes, ralentissant ainsi la vitesse de chargement des données et rendant l'ensemble du processus d'entraînement plus long. De plus, cette méthode pouvait parfois générer des images ayant des valeurs manquantes ou invalide, provoquant des erreurs et des interruptions lors de l'entraînement sur ces images problématiques.

Pour améliorer l'efficacité de l'entraînement et éviter ces erreurs d'interruption, j'ai décidé de prétraiter les images SAR avant leur chargement, en les découpant en blocs de 256×256 pixels et en sauvegardant ces images traitées dans de nouveaux fichiers. Ainsi, le processus de chargement des données a été simplifié pour consister uniquement en la lecture et la transformation, évitant ainsi les opérations répétées de découpage des images.

Dans le contexte de l'apprentissage profond, un DataLoader est essentiel pour gérer et traiter les données lors de l'entraînement d'un modèle. Il divise automatiquement les données en batches,

applique les transformations nécessaires et les charge efficacement en utilisant le parallélisme. Cela permet de maximiser les performances du GPU et d'assurer un flux de données constant au modèle, ce qui réduit les temps d'attente et accélère l'entraînement.

J'ai donc réalisé la classe **ProcessedForestDatasetLoader**, qui est utilisée pour charger les jeux de données d'entraînement, de validation et de test. Les données sont rangées dans des dossiers spécifiques et chargées via une classe de jeu de données personnalisée.

6.3.5 Augmentation des données

Quand j'ai travaillé sur la classe **DatasetLoader**, j'ai également testé l'augmentation des données pour évaluer son impact sur les performances. J'ai essayé des transformations comme la rotation et le redimensionnement sur les images SAR. Cependant, j'ai remarqué que ces images présentent peu de variations, ce qui rend ces transformations moins utiles. Les tests ont montré que l'augmentation des données n'améliorait pas les résultats de l'entraînement de manière significative et consommait des ressources de calcul supplémentaires. Par conséquent, je n'ai pas intégré l'étape d'augmentation des données dans le chargement des données.

6.4 Conception et entraînement de l'auto-encodeur

6.4.1 Environnement d'entraînement et configuration

6.4.1.1 L'environnement matériel et logiciel

Système d'exploitation: Ubuntu 22.04

Environnement virtuel: Anaconda 24.7.1

Bibliothèques et versions: **requirement.txt** dans mon dépôt GitHub [Annexe 1]

6.4.1.2 Réalisation du mécanisme d'arrêt anticipé

Dans l'entraînement des modèles d'apprentissage profond, il est essentiel d'éviter le sur-apprentissage (overfitting), qui se produit lorsque le modèle s'adapte trop aux données d'entraînement et ne fonctionne pas bien sur des données non vues. Le mécanisme d'arrêt anticipé permet de résoudre ce problème en interrompant l'entraînement dès que la performance sur les données de validation ne s'améliore plus.

Ce mécanisme surveille les pertes de validation pendant l'entraînement. Lors de chaque époque, la classe **EarlyStopping** reçoit la perte de validation actuelle. Elle calcule ensuite une perte moyenne avec une moyenne des pertes récentes, pour obtenir une valeur plus stable pour évaluer les améliorations. Si cette perte lissée diminue et dépasse le meilleur score précédemment enregistré, ce score est mis à jour et le modèle correspondant est sauvegardé. Sinon, un compteur

augmente pour chaque époque sans amélioration significative. Si les pertes ne s'améliorent pas au-delà d'un seuil prédéfini (`delta`) sur un certain nombre d'époques consécutives (`patience`), l'entraînement s'arrête pour prévenir tout entraînement inutile et éviter le risque de sur-apprentissage. Ce mécanisme assure que le modèle sauvegardé représente la meilleure performance obtenue pendant l'entraînement, et garantit son adaptabilité à de nouvelles données.

6.4.1.3 Configuration des hyperparamètres

Le modèle utilise un ensemble bien défini d'hyper paramètres pour assurer un entraînement et des tests robustes. Voici une brève introduction aux principaux hyperparamètres et configurations utilisés :

`Epochs` = 10

`Learning rate` = 1e-4

`Weight decay` = 6e-6

`Loss function`: l'erreur quadratique moyenne (MSE) de chaque pixel

`Optimizer` : Adam with L2 regularization (`weight_decay`)

`Scheduler` : StepLR (Réduction par étapes)

La combinaison de ces hyperparamètres est le résultat de nombreuses expérimentations manuelles. Plus précisément, la combinaison du taux d'apprentissage, de la réduction de poids, de la taille d'étape (StepLR) et de la valeur gamma (StepLR) a été dérivée en utilisant `Optuna` et représente le meilleur résultat possible dans les conditions données. [explication dans 6.5.1]

6.4.2 Processus de conception et d'amélioration de la structure du modèle

Le processus de développement du modèle peut être divisé en trois étapes principales.

Dans la première étape, je me suis formé au processus d'entraînement d'un auto-encodeur en utilisant le jeu de données MNIST [Deng, 2012] ainsi qu'un encodeur et décodeur CNN (Convolutional Neural Networks) [LeCun et al., 1989] de base. Cette phase m'a permis de comprendre les étapes essentielles pour entraîner un modèle d'apprentissage profond.

La deuxième étape consistait à entraîner le modèle avec un jeu de données sans déforestation. Cela avait pour objectif d'apprendre et de reconstruire les caractéristiques des zones forestières. Pour améliorer les performances, j'ai intégré plusieurs modules inspirés par la lecture de publications scientifiques et l'étude de projets open source, tels que le Pyramid Pooling Module, les residual blocks et les skip connections.

Une fois que le modèle pouvait générer des images SAR de forêts sans déforestation de manière satisfaisante, j'ai commencé à le tester avec un jeu de données contenant des zones déforestées. Cette phase visait à vérifier si le modèle pouvait détecter efficacement les zones de déforestation. Pendant ce processus, j'ai continué à ajuster la structure du modèle pour améliorer ses performances.

6.4.2.1 Structure initiale du modèle

Le modèle auto-encodeur initial se compose d'une classe d'encodeur et d'une classe de décodeur, toutes deux basées sur CNN. Comme le jeu de données final n'était pas encore prêt, le modèle a d'abord été conçu pour tester si l'auto-encodeur pouvait effectuer correctement le processus d'entraînement. Pour cela, les tests ont été réalisés avec le jeu de données MNIST, un ensemble bien connu d'images monochromes de chiffres manuscrits, de taille 28×28 pixels. Cette phase a permis de valider que le modèle était capable de compresser et reconstruire des images simples avant de l'appliquer aux données SAR.

6.4.2.2 Amélioration avec des images de foret

Dans la deuxième phase du développement du modèle, plusieurs modifications importantes ont été apportées à l'architecture pour améliorer ses performances. Ces changements ont été introduits en quatre étapes principales qui se sont enchaînées naturellement tout au long de cette période.

Tout d'abord, j'ai commencé par adapter les dimensions d'entrée et de sortie pour traiter des images SAR en deux canaux (VV et VH) de taille 256×256 . Pour ce faire, le nombre de couches de convolution de l'encodeur a été augmenté à six, avec une réduction progressive de la résolution spatiale à chaque étape. La structure de l'encodeur a été modifiée pour gérer un vecteur de caractéristiques généré par une couche dense et fournir un vecteur en sortie. Le décodeur, quant à lui, a été ajusté pour inclure davantage de couches de déconvolution et ainsi rétablir la taille des images à leur dimension initiale.

Ensuite, pour résoudre les problèmes liés à la profondeur accrue du modèle, j'ai introduit des blocs résiduels [He et al., 2015]. Les blocs résiduels ont permis de prévenir la disparition ou l'explosion du gradient, tout en améliorant la capacité du réseau à apprendre des caractéristiques complexes.

De plus, les connexions directes ont été ajoutées pour transmettre directement les informations des couches précoces aux couches tardives, inspirées des architectures de type U-Net [Ronneberger et al., 2015]. Cela a été essentiel pour conserver les détails à faible niveau lors de la reconstruction des images SAR.

Enfin, pour enrichir la perception du modèle des informations à différentes échelles, un PPM (module de regroupement pyramidal) [Zhao et al., 2016] a été intégré. Ce module a permis d'extraire

des caractéristiques globales et locales des images SAR, renforçant la robustesse du modèle face aux variations de taille et de contexte.

Ces modifications, dans un processus d'amélioration continue de l'auto-encodeur, ont transformé le modèle initial en un système capable de capturer efficacement les caractéristiques des zones forestières et de reconstruire avec précision les images SAR.

6.5 Détection des anomalies pour identifier les zones déforestées

6.5.1 Utilisation de l'Auto-Encodeur pour la Détection de la Déforestation

Pendant cette étape d'utilisation de l'auto-encodeur préalablement entraîné, plusieurs modifications ont été apportées pour adapter l'approche et maximiser la précision de détection dans le contexte de la déforestation.

Pour améliorer la capacité du modèle à se concentrer sur les caractéristiques propres aux zones forestières intactes, le nombre de connexions directes dans l'architecture a été réduit. Initialement, quatre connexions étaient présentes, mais elles ont été diminuées à deux, ciblant les résolutions de 16×16 et 8×8 pixels. Cette modification vise à réduire l'accès du modèle aux informations de bas niveau provenant de l'image d'origine, ce qui avait pour effet d'améliorer artificiellement la reconstruction des zones non forestières. En limitant ces connexions, le modèle a été contraint de mieux analyser les caractéristiques des zones forestières, ce qui a renforcé sa capacité à détecter les anomalies liées à la déforestation.

Pour améliorer l'intégration des informations à différentes échelles, le PPM a été remplacé par le FPN (réseau pyramidal de fonctionnalités) [Lin et al., 2017]. Contrairement au PPM, qui applique des convolutions sur des résolutions différentes à partir de la couche la plus profonde, le FPN combine des informations provenant de différentes couches de l'encodeur à l'aide de connexions latérales. Cette méthode a pour objectif de mieux saisir les éléments particuliers des zones forestières préservées et les modifications associées à la déforestation. Le décodeur exploite ces caractéristiques pour mieux isoler les anomalies et fournir une détection plus fiable.

Des modules de self-attention [Vaswani et al., 2017] ont été intégrés dans l'encodeur et le décodeur pour améliorer la prise en compte des relations spatiales complexes. Dans l'encodeur, un module de self-attention a été ajouté au niveau le plus profond pour permettre au modèle de se focaliser sur les zones forestières intactes et d'analyser les changements globaux. Dans le décodeur, un module similaire a été inséré après la troisième étape de upsampling, afin de renforcer la reconstruction des anomalies liées à la déforestation et d'améliorer la précision au niveau des bordures des zones de changement.

En adaptant ces différents aspects de l'approche, le modèle a été amélioré pour détecter les anomalies liées à la déforestation. Plutôt que de se concentrer sur une reconstruction globale précise des images SAR, les modifications apportées ont permis au modèle de mieux différencier les zones forestières intactes des zones affectées, réduisant ainsi les erreurs de détection et améliorant la détection des changements significatifs.

6.5.2 Conclusion de la version finale du modèle

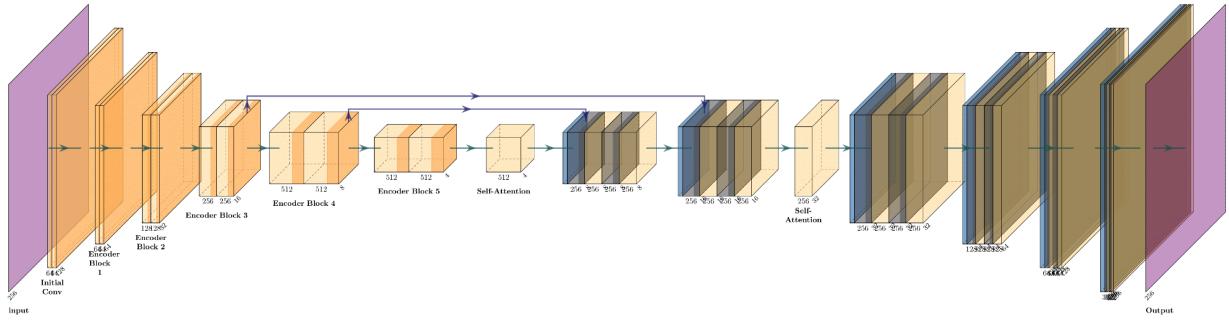


Figure 10: Architecture de l'auto-encodeur

L'encodeur convertit les données d'entrée en versions plus petites en utilisant des blocs résiduels et un module de self-attention. Ce dernier aide à se concentrer sur les zones importantes de l'image. Le FPN combine les informations de différents niveaux pour capturer à la fois les détails précis et globaux.

Le décodeur utilise des blocs de déconvolution qui augmentent progressivement la résolution, tout en fusionnant les caractéristiques du FPN pour préserver les détails. Un module d'auto-attention dans le décodeur améliore la détection des changements subtils, en particulier aux frontières des zones forestières et déforestées.

Ce modèle (Figure 10) utilise plusieurs modules pour détecter efficacement les changements dans les zones de déforestation. La détection repose sur les erreurs de reconstruction : les zones intactes de forêt sont bien reconstruites par le modèle, tandis que les zones déforestées présentent des erreurs plus marquées. Ces erreurs permettent d'identifier les anomalies correspondant aux changements environnementaux.

6.5.3 Le traitement des résultats du modèle

Le traitement des résultats du modèle consiste à générer des cartes d'erreurs de reconstruction pour chaque pixel, ce qui est essentiel pour distinguer les zones « forêt » et « non-forêt ». Étant donné que les seuils de classification diffèrent d'une image SAR à l'autre, l'application d'un seuil constant s'est révélée peu efficace. Pour améliorer la généralisation du modèle et simplifier le processus de classement tout en augmentant sa précision, j'ai choisi d'utiliser des méthodes de clustering, comme les algorithmes de Gaussian Mixture Model (GMM) implémentés dans la bibliothèque **Scikit-learn**.

Le GMM est une technique qui classe les données en plusieurs types de distributions gaussiennes. Grâce à l'algorithme d'espérance-maximisation, il calcule des valeurs comme la moyenne, la variance et la part de chaque groupe. En assignant chaque donnée au groupe le plus probable, GMM permet de repérer des motifs et des structures variées dans les données.

Dans le contexte de la détection de la déforestation, le GMM est appliqué aux erreurs de reconstruction générées par l'auto-encodeur. Les données correspondant aux zones forestières intactes présentent des erreurs de reconstruction faibles et relativement homogènes, tandis que les zones déforestées montrent des erreurs plus élevées et hétérogènes. Le GMM regroupe ces erreurs en plusieurs classes (par exemple, erreurs faibles pour les forêts intactes et erreurs élevées pour les zones déforestées). Ce regroupement permet d'isoler les anomalies correspondant aux changements de couverture forestière, facilitant ainsi l'identification des zones déforestées.

6.5.4 Visualisation des résultats

La première fonction que j'ai développée permet de visualiser en même temps l'image originale, la carte thermique des erreurs de reconstruction et les résultats de clustering (Figure 12). Cette fonction offre une vue d'ensemble des performances du modèle sur une image, tout en facilitant l'analyse des détections et des pertes pour chaque pixel (Figure 11).

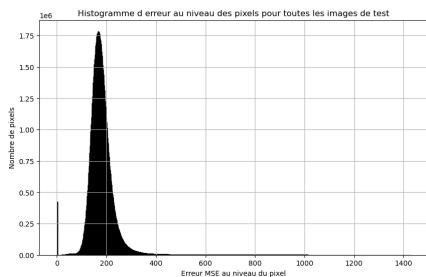


Figure 11: Distribution des valeurs de MSE

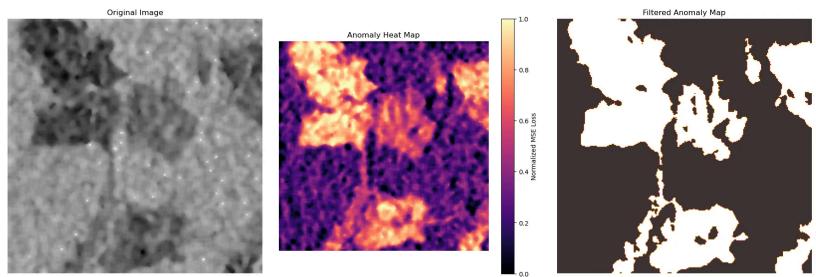


Figure 12: Visualisation des résultats préliminaires

6.5.5 Analyse des changements temporels

Je me suis basé sur la première fonction pour développer une deuxième fonction qui analyse les données temporelles. Cette nouvelle fonction permet de choisir une date et une position spécifiques, puis de récupérer les deux images précédent et suivant cette date. Ces images sont visualisées ensemble pour montrer les changements dans le temps, ce qui aide à étudier les tendances de déforestation.

Cette analyse repose sur deux approches principales. La première consiste à comparer les résultats de clustering de deux images consécutives, les zones où des différences apparaissent étant identifiées comme des « zones de déforestation » (Figure 13). La deuxième stratégie va plus loin en classifiant les changements observés : les zones nouvellement détectées comme non forestières sont considérées comme « déforestations récentes », tandis que celles détectées comme non forestières depuis plusieurs mois consécutifs sont considérées comme « anciennes déforestations » (Figure 14).

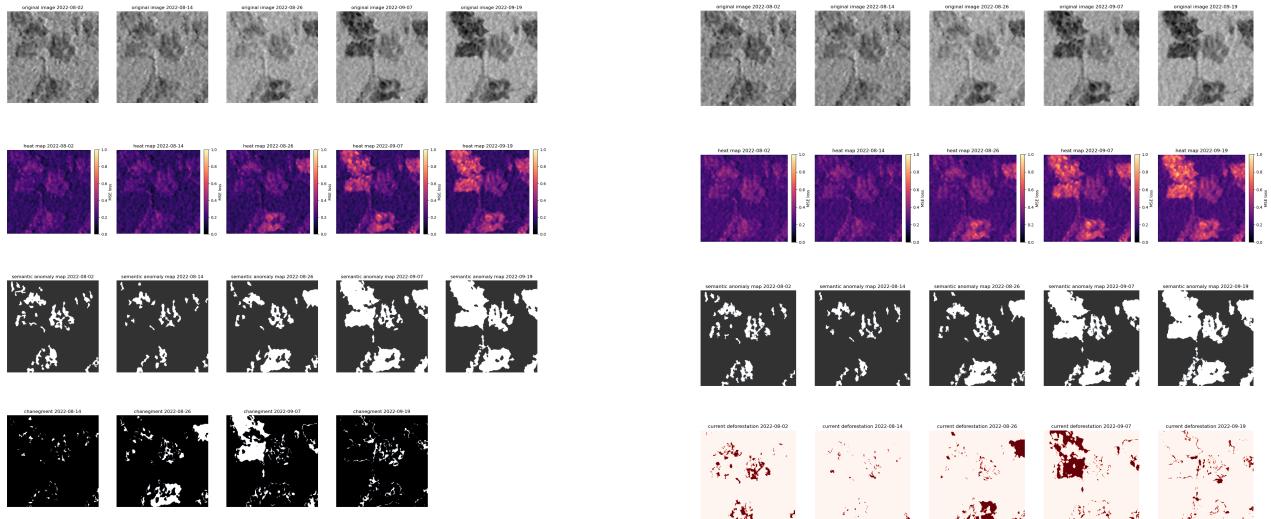


Figure 13: Clustering par GMM

Figure 14: Classification par séries chronologiques

6.5.6 Génération des résultats complets de détection d'anomalies

Pour simplifier l'utilisation du modèle et éviter d'exécuter les détections image par image, j'ai développé une troisième fonction. Cette fonction prend toutes les images d'une date précise, effectue la classification et la projection géographique en une seule étape, puis exporte les résultats sous forme de fichier Shapefile. Cette méthode permet une intégration simple avec des outils systèmes d'information géographique et améliore la précision grâce à une gestion uniforme des informations géographiques. [explication dans 6.5.3] [Résultat de 2022.4 à 2022.9 dans 7.1]

6.6 Défis rencontrés et solutions

6.6.1 Le problème d'amélioration systématique des hyperparamètres

L'amélioration des hyperparamètres est une étape essentielle pour améliorer les performances des modèles d'apprentissage automatique. Dans mon projet, j'ai commencé par passer environ une semaine à faire des tests manuels en utilisant une méthode de contrôle des variables. Cela m'a aidé à essayer différentes combinaisons des hyperparamètres et à trouver certaines configurations efficaces.

6.6.1.1 Approche manuelle

En suivant une méthode systématique, j'ai optimisé certains hyperparamètres. Par exemple, l'optimiseur Adam, associé au planificateur de taux d'apprentissage StepLR, donnait les meilleurs résultats en termes de perte. Après avoir fait ces choix, j'ai examiné d'autres hyperparamètres, comme le taux d'apprentissage, la pénalisation des poids, ainsi que le pas et le facteur de réduction

du planificateur. Cependant, ces paramètres ont montré une variabilité significative et une forte interdépendance, ce qui a rendu difficile l'obtention d'une configuration stable.

6.6.1.2 Approche automatique

Malgré plusieurs cycles de tests manuels, les performances du modèle sont restées fluctuantes. Sur les recommandations de Mme. Berti, j'ai examiné divers frameworks d'amélioration automatique des hyperparamètres, notamment **Optuna**, **RayTune** et **Hyperopt**. Après une analyse comparative, j'ai décidé de retenir **Optuna** en raison de sa facilité d'intégration et de sa flexibilité.

Avec Optuna, j'ai configuré une fonction qui propose des valeurs pour des hyperparamètres clés. Et pour chaque proposition, le modèle a été réinitialisé et entraîné sur 10 époques avec un contrôle de validation et un arrêt anticipé. Optuna vérifie la perte de validation pour déterminer la meilleure combinaison.

code exemple:

```
def objective(trial, args, architectures):
    lr = trial.suggest_float('lr', 1e-4, 5e-4, step=1e-4)
    weight_decay = trial.suggest_float('weight_decay', 1e-6, 1e-5,
                                      step=1e-6)
    args.lr = lr
    args.weight_decay = weight_decay
    autoenc = architectures[args.model]
    autoenc.__init__(args)
    for epoch in range(1, args.epochs + 1):
        autoenc.train(epoch)
        should_stop, val_loss = autoenc.test(epoch)
        if should_stop:
            break
    return val_loss
```

```
if args.use_optuna:
    study = optuna.create_study(direction='minimize')
    study.optimize(lambda trial: objective(trial, args, architectures),
                  n_trials=10)
```

6.6.2 Le problème de normalization des jeu de données

Dans les ensembles d'entraînement et de validation, les images SAR ne contiennent que des zones forestières. Après le passage par la fonction de prétraitement, les valeurs se situent dans l'intervalle étroit de [-15, 3], avec une moyenne d'environ « -7 ». En revanche, l'ensemble de test

inclus à la fois des zones forestières et non forestières, ce qui élargit l'intervalle des valeurs à [-25, 5], tout en conservant la même moyenne de « -7 ».

Lors de l'implémentation initiale de la classe **DataLoader**, chaque ensemble de données a été normalisé selon ses propres valeurs maximales et minimales, ce qui a ramené les données entre « 0 et 1 ». Cependant, cette méthode a conduit à une normalisation des valeurs des zones forestières qui n'était pas la même pour les ensembles d'entraînement et de validation par rapport à l'ensemble de test. Cela a entraîné un effet inattendu : dans les résultats de la détection, les zones forestières obtenaient des MSE plus élevées que celles des zones non forestières (Figure 16).

Ce problème a été compliqué à identifier. Après avoir vérifié plusieurs causes possibles, lors d'une réunion hebdomadaire avec mon tuteur, M. Guérin, l'idée d'une erreur dans la normalisation des données a été mentionnée. Cela m'a aidé à orienter mes recherches et à confirmer que cette erreur était la cause du problème.

6.6.2.1 Solution et expérimentations

Pour corriger ce problème, j'ai modifié la classe **DataLoader** pour appliquer une normalisation globale à tous les ensembles de données, en utilisant les mêmes limites de valeurs.

Tout d'abord, j'ai testé une approche sans aucune normalisation. Bien que cela gardait les valeurs originales, cette méthode rendait la descente de la fonction de perte difficile à comprendre pendant l'entraînement. Cela a rendu l'évaluation de l'efficacité du modèle plus difficile et a compliqué l'analyse des résultats intermédiaires.

J'ai ensuite essayé de normaliser les données en appliquant une **normalisation MinMax** sur l'intervalle de [-25, 5], qui représente les valeurs extrêmes de l'ensemble de test. Même si cette méthode a permis d'inclure toutes les données dans la plage normalisée, elle a réduit l'importance relative des caractéristiques liées aux zones forestières. Cela a rendu le modèle moins performant pour apprendre les motifs liés aux forêts (Figure 16).

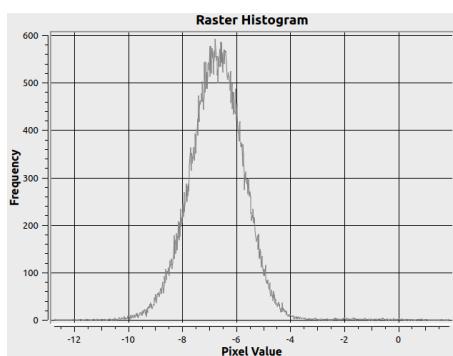


Figure 15: La plage de valeurs des images SAR

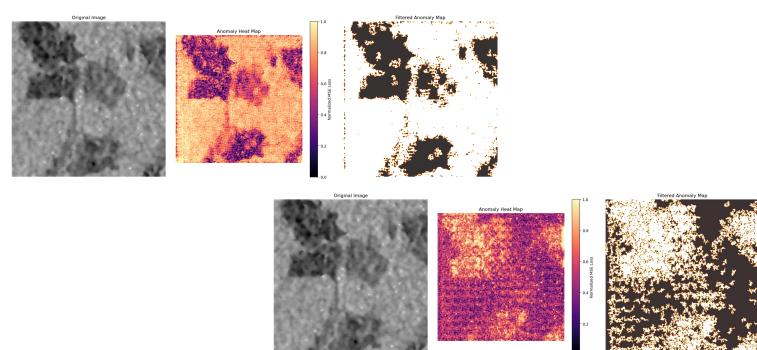


Figure 16: Les résultats d'une mauvaise normalisation

Enfin, j'ai choisi de normaliser les données sur l'intervalle [-15, 3] (Figure 15), qui correspond aux valeurs des ensembles d'entraînement et de validation. Cela a permis de normaliser les zones

forestières dans l'intervalle [0, 1], tout en maintenant les valeurs anormales en dehors de cette plage. Les expérimentations ont montré que cette approche était la plus efficace, offrant des résultats fiables et précis, tout en facilitant l'analyse des performances du modèle.

6.6.3 Le problème de projection des résultat de détection

Lors de l'utilisation des fonctions de prétraitement du projet CuSum, j'ai rencontré des difficultés pour projeter les résultats de détection de mon modèle sur le système de coordonnées géographiques. Ce problème était lié à l'utilisation de la fonction `geo_parameterize` de la bibliothèque `PyroSAR`, qui projette les images SAR sur le système `EPSG:4326` avec une inclinaison. Pendant le prétraitement des images, cette inclinaison introduit des zones contenant des valeurs NaN dans les coins des images après les découpes. Ces valeurs affectent la cohérence des ensembles de données d'entraînement et de validation.

6.6.3.1 Approche initiale

Pour résoudre ce problème initial, j'ai choisi de désactiver temporairement la fonction `geo_parameterize` dans le processus de prétraitement. Cela m'a permis de créer des jeux de données sans coordonnées géographiques, éliminant ainsi les valeurs manquantes ou invalides. Toutefois, cette décision a généré un nouveau problème lors de l'évaluation des résultats du modèle. Sans coordonnées géographiques préservées, il était impossible de comparer directement les détections avec mes annotations manuelles précédemment créées.

6.6.3.2 Projection manuelle

Dans un premier temps, j'ai développé une fonction capable de positionner les résultats dans le système de coordonnées géographiques en utilisant les coordonnées du coin nord-ouest de l'image et l'angle d'inclinaison. Cette méthode a donné une projection approximative, mais les modifications manuelles des paramètres n'ont pas permis d'atteindre une précision satisfaisante ([Figure 17](#)).

6.6.3.3 Refaire le preprocess fonction

En réévaluant le processus de prétraitement, j'ai compris qu'il était essentiel de garder les informations géographiques dans les images du jeu de test. Donc, j'ai activé la fonction `geo_parameterize` et recréé le jeu de test tout en préservant les mêmes coordonnées géographiques. Avec cette méthode, les résultats de détection peuvent être projetés à l'aide uniquement des coordonnées du coin nord-ouest, ce qui a amélioré significativement la précision de la projection.

Cependant, cette méthode a révélé une limite : en raison des différences entre les systèmes de coordonnées géographiques et projetés, des erreurs de conversion entre « mètres » et « degrés »

apparaissaient. Ces erreurs s'accumulent lors de la juxtaposition des blocs individuels, entraînant une précision moindre dans les zones éloignées du point de référence ([Figure 18](#)).

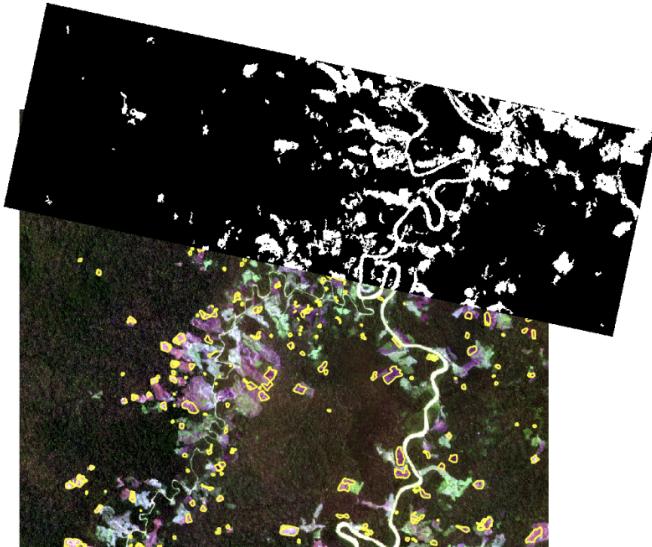


Figure 17: Projection manuelle

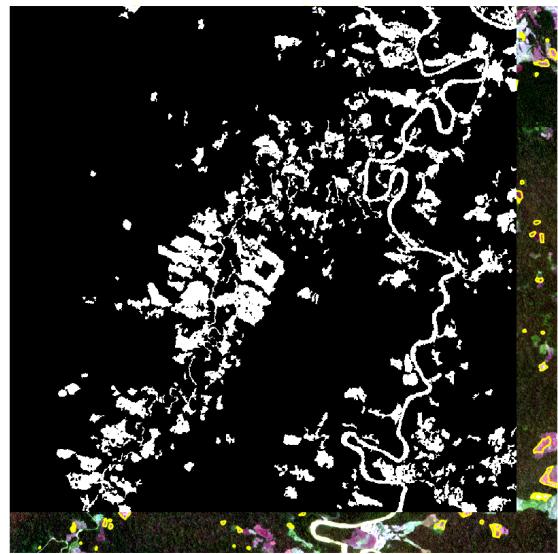


Figure 18: Projection avec geo_parameterize

6.6.3.4 Projection avant la combination

Pour réduire ces erreurs cumulatives, j'ai essayé une nouvelle méthode. Au lieu de mettre ensemble les blocs avant de les projeter, j'ai décidé de projeter chaque bloc séparément dans le système de coordonnées géographiques avant de les assembler complètement.

Cette nouvelle méthode a nécessité un ajustement du processus de préparation. En remplaçant la bibliothèque `tifffile` par `rasterio` pour la découpe des images SAR, j'ai pu conserver les informations géographiques de chaque bloc. Les détections faites par le modèle incluent alors automatiquement les informations géographiques précises des données sources. Cette méthode a permis de réduire considérablement les erreurs de projection et d'améliorer la précision globale des résultats. [[Résultat de 2022.4 à 2022.9 dans 7.1](#)]

6.7 Gestion des versions du code

La gestion des versions du code a été réalisée principalement à l'aide de Git et de GitHub [[Annexe 1](#)], ce qui permet de suivre facilement toutes les modifications effectuées pendant le développement du projet. Voici la structure du dépôt :

```
SAR_Deforestation_Detection_Autoencoder/
├── config/
│   └── parse_args.py
```

```

└── pipeline/
    ├── anomaly_detection/
    │   ├── anomaly_detection_pipeline.py
    │   ├── anomaly_detection.py
    │   └── visualization.py
    ├── test/
    │   └── test_pipeline.py
    ├── train/
    │   └── train_pipeline.py
    ├── datasets/
    │   └── preprocessing/
    │       └── data_loader.py
    ├── models/
    │   ├── plotneuralnet/
    │   ├── architectures.py
    │   ├── auto-encodeur.py
    │   └── variational_auto-encodeur.py
    ├── utils/
    │   ├── early_stop.py
    │   └── hyperparameter_optimize.py
    └── train.py
    └── requirements.txt
    └── README.md

```

Le dossier `config` contient un fichier nommé `parse_args.py`, qui permet de gérer les paramètres de configuration de manière intuitive pour l'ensemble du processus d'entraînement. Ce fichier offre une grande flexibilité au projet.

Dans le dossier `pipeline`, le code est divisé en trois parties correspondant à des fonctions distinctes. Le sous-dossier `anomaly_detection_pipeline` gère les tâches liées à la détection d'anomalies. Les fichiers de ce dossier collaborent pour effectuer les post-traitements et analyser les résultats d'entraînement. Le sous-dossier `test_pipeline` organise les processus de test des modèles, tandis que `train_pipeline` gère l'entraînement des modèles. Ces deux parties travaillent directement avec les fonctions appelées dans le fichier principal `train.py`.

Le traitement des données est centralisé dans le dossier `datasets`. Celui-ci contient un sous-dossier `preprocessing`, dédié au prétraitement des données. Cela inclut des étapes comme le téléchargement, la division et le nettoyage des données. Le fichier `data_loader.py` se charge de gérer le chargement efficace des données pour les modèles.

Le code des modèles se trouve dans le dossier `models`. On y trouve des fichiers comme `architectures.py`, qui définit les structures de réseau, ainsi que deux implémentations de

modèles clés : `auto-encodeur.py` et `variational_auto-encodeur.py`. De plus, la partie `plotneuralnet` utilise un projet open source pour visualiser les architectures définies dans `'architectures.py'`, ce qui rend le projet plus compréhensible et facile à présenter.

Les outils utilisés pendant l'entraînement sont regroupés dans le dossier `utils`. Parmi eux, `early_stop.py` met en œuvre un mécanisme d'arrêt anticipé pour économiser des ressources, tandis que `hyperparameter_optimize.py` se concentre sur l'amélioration des hyperparamètres afin d'améliorer les performances du modèle.

Le répertoire racine contient également plusieurs fichiers importants. `train.py` sert de point d'entrée principal pour l'entraînement des modèles. `requirements.txt` liste toutes les dépendances nécessaires pour assurer la cohérence de l'environnement. Enfin, `README.md` présente une vue d'ensemble du projet et explique son utilisation.

Dans mon flux de travail, je commence par apporter des modifications au code localement. Après chaque modification importante, je pousse les changements vers une branche nommée `test` afin d'éviter toute perte de données. Dans les messages de commit, je décris en détail les modifications apportées, ce qui permet de suivre facilement l'historique des modifications. Une fois qu'une tâche est complète et que ses résultats ont été validés, je fusionne le contenu de la branche `test` dans la branche principale `main`.

Au début, avant d'apprendre certaines techniques avancées de Git, j'ai rencontré des problèmes liés à l'utilisation inappropriée de la commande `stash`, ce qui a créé des conflits entre mon dépôt local et distant. Finalement, j'ai résolu ces problèmes en utilisant des `pull requests` et des fusions manuelles, ce qui m'a permis de mieux comprendre Git.

Utiliser GitHub pour la gestion des versions a été une expérience très enrichissante. Cela m'a permis de travailler de manière plus structurée et de comprendre pleinement l'importance de conserver un historique clair et organisé des modifications. De plus, l'utilisation des branches et des pull requests a renforcé mes compétences en gestion de projet, rendant le processus de développement plus fluide et efficace.

7. Résultats

7.1 Analyse qualitative des résultats obtenus

Dans cette étude, l'erreur quadratique moyenne (MSE) a été utilisée comme indicateur principal pour évaluer la qualité des reconstructions générées par l'auto-encodeur. Une diminution du MSE reflète une amélioration notable de la capacité du modèle à reconstruire les zones forestières avec précision.

Cependant, en raison des nombreux ajustements effectués sur la structure du modèle et les hyperparamètres, il a été difficile de définir un MSE optimal fixe. Les performances du modèle ont été principalement évaluées via les courbes MSE générées par TensorBoard ([Figure 20](#)), permettant de suivre l'évolution des reconstructions durant l'entraînement. Les reconstructions finales montrent une nette amélioration en termes de distinction entre les zones forestières et déforestées. Les erreurs de reconstruction se concentrent désormais principalement sur les zones présentant des changements réels, ce qui facilite l'identification des anomalies environnementales.

Ces observations qualitatives démontrent l'efficacité du modèle dans la capture des caractéristiques essentielles des données SAR ([Figure 19](#)).

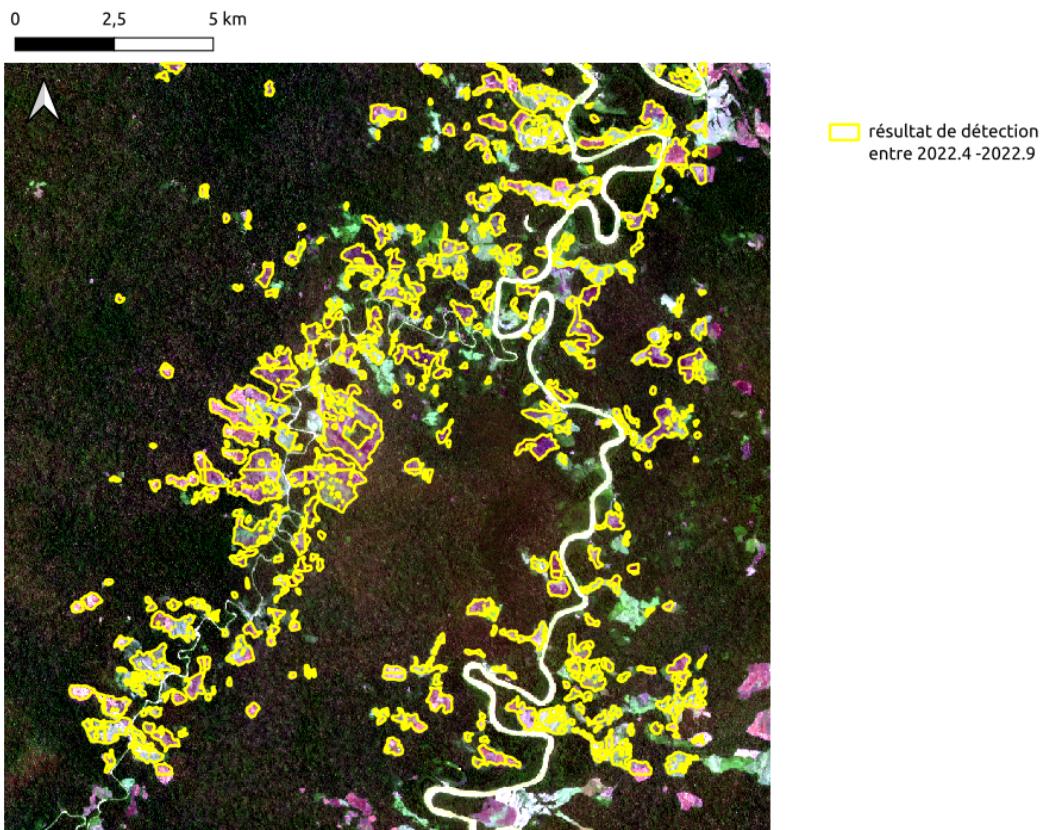


Figure 19: Exemple de détection - résultat final, PlanetScope capturé le 19 septembre 2022

jeune polygones correspond à : Déforestation détectée entre avril 2022 et septembre 2022

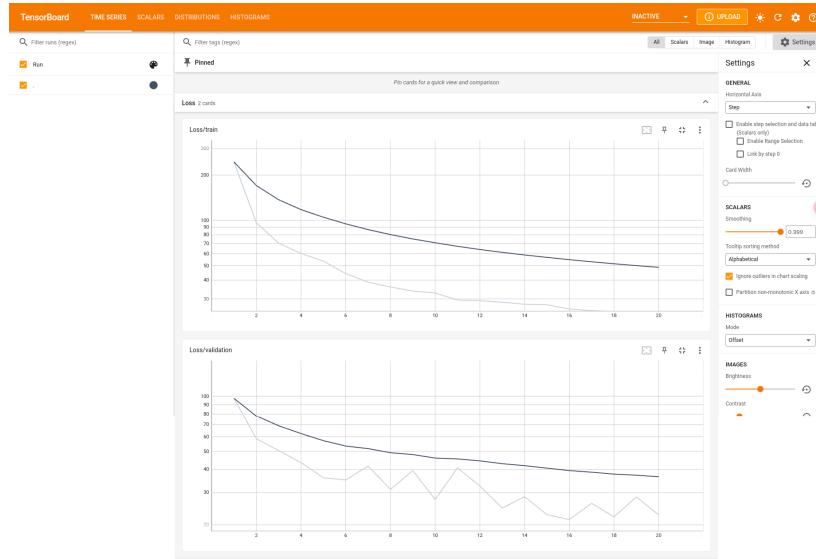


Figure 20: Exemple de courbe de fonction de perte générée par Tensorboard

7.2 Analyse quantitative des résultats obtenus

7.2.1 Métriques d'évaluation des performances

Pour évaluer les performances du modèle, un processus systématique a été mis en place en utilisant trois fichiers shapefile : **forest mask**, **annotation** et **détection résultat**. Les étapes suivantes ont été réalisées :

Prétraitement des données :

Les trois fichiers shapefile ont été lus, nettoyés et convertis en un tableau de pixels (rasterization) pour garantir la même zone et la même échelle. Les polygones vides ou invalides ont été supprimés afin d'éviter les erreurs de géométrie, et chaque shapefile a été projeté dans un même système de coordonnées. Ces opérations assurent la cohérence spatiale et la fiabilité des calculs.

Suppression des zones non pertinentes :

En soustrayant les résultats de détection du fichier forest mask, les zones initialement non forestières ont été éliminées des détections. Cela garantit que seules les anomalies pertinentes (zones forestières détectées comme ayant changé) sont prises en compte dans l'évaluation.

Calcul des métriques :

Les résultats finaux ont été comparés aux annotations manuelles pour calculer les métriques suivantes :

True Positives (TP) : Les zones détectées comme changées par le modèle, qui correspondent réellement à des changements de forêt selon les annotations.

False Positives (FP) : Les zones détectées comme changées par le modèle, mais qui ne correspondent pas à des changements réels dans les annotations.

False Negatives (FN) : Les zones réellement changées selon les annotations, mais qui n'ont pas été détectées par le modèle.

Définitions des Métriques :

Précision (Precision) : Proportion de prédictions correctes parmi toutes les zones détectées comme changées. Une précision élevée signifie peu de fausses alarmes.

$$Precision = \frac{TP}{TP + FP}$$

Rappel (Recall) : Proportion des changements réels correctement détectés par le modèle. Un rappel élevé signifie que la majorité des changements sont détectés, même si cela inclut quelques erreurs.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score : Moyenne harmonique entre la précision et le rappel. Il équilibre la précision et le rappel pour donner une vue d'ensemble des performances.

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Intersection over Union (IoU) : Mesure de la qualité de la prédition en comparant l'intersection et l'union des zones prédites et réelles.

$$IoU = \frac{\text{Zone commune (Intersection)}}{\text{Zone totale (Union)}}$$

7.2.2 Évaluation des performances de détection

Les images optiques de PlanetScope utilisées dans cette étude sont des compositions **mensuelles**, générées à partir de nombreuses acquisitions quotidiennes assemblées pour minimiser la couverture nuageuse. En revanche, Sentinel-1 fournit une image SAR **tous les 12 jours**. Cette différence de fréquence et de mode d'acquisition a influencé l'organisation des annotations : celles-ci ont été réalisées manuellement en s'appuyant sur les compositions mensuelles de PlanetScope, tandis que les résultats de mon modèle sont issus des images SAR individuelles de Sentinel-1. En raison de cette divergence temporelle, un même événement de déforestation peut apparaître à des moments différents dans les deux jeux de données, ce qui complique l'évaluation. Pour y remédier, j'ai décidé de comparer les résultats issus de la première et la dernière image SAR avec l'ensemble des annotations mensuelles réalisées entre **avril et octobre 2022**, afin de calculer les indicateurs finaux.

Résultats obtenus :

Le modèle atteint une **Precision = 0.8284**, ce qui indique qu'il produit peu de fausses alarmes. Le **Recall = 0.6865** montre que certaines zones réelles de changement n'ont pas été détectées par le modèle, ce qui peut s'expliquer par des limites dans les données d'entraînement ou la résolution des images. Le **F1-Score = 0.7508** reflète un bon équilibre entre précision et rappel. Le **IoU = 0.6011** indique une qualité correcte des prédictions en termes de correspondance spatiale avec les annotations manuelles.

Ces résultats montrent que le modèle est capable d'identifier efficacement les zones de déforestation, tout en laissant une marge d'amélioration, notamment en réduisant les faux négatifs et en améliorant le rappel.

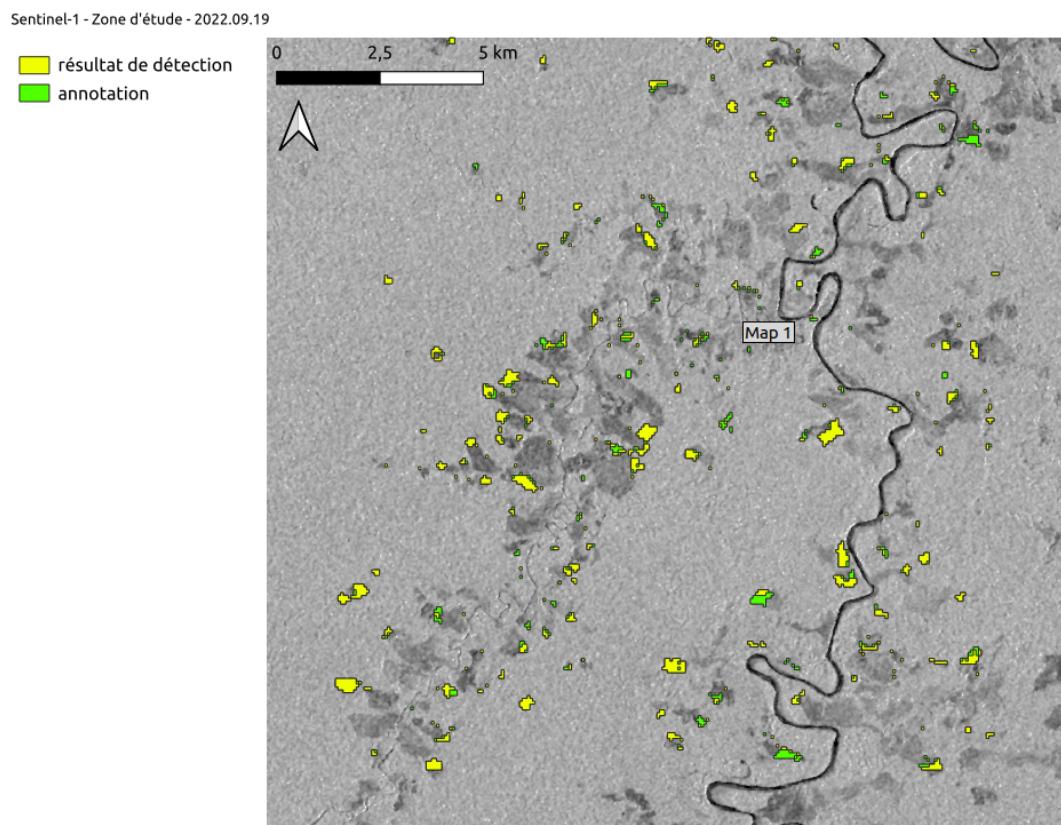


Figure 21: Comparaison du résultat de détection et d'annotation masquées

8. Discussion

8.1 Contributions Méthodologiques et Performances

Ce projet a permis d'apporter des contributions méthodologiques importantes pour l'analyse des changements environnementaux à partir des données SAR.

D'un point de vue méthodologique, une première contribution réside dans l'élaboration d'un pipeline complet de prétraitement des données SAR. Ce pipeline inclut des étapes spécifiques de segmentation et nettoyage des images, ainsi qu'une normalisation globale basée sur les plages de valeurs issues des images du jeu d'entraînement. Cette méthodologie garantit une meilleure cohérence dans les résultats du modèle tout en réduisant les variations entre les images.

De plus, un modèle flexible, basé sur un auto-encodeur, a été développé pour détecter les changements anormaux dans les images SAR. Contrairement à des approches comme CuSum, qui nécessitent des séries temporelles complètes et se concentrent uniquement sur les changements statistiques des pixels, notre modèle intègre la cohérence spatiale des prédictions. Cette intégration des relations spatiales permet une détection plus précise des anomalies liées à la déforestation, même avec des données temporelles limitées. Les premiers tests ont montré des résultats prometteurs, avec une détection précise des zones de déforestation grâce à l'analyse des erreurs de reconstruction.

Enfin, les approches actuelles de post-traitement, basées sur le clustering et l'analyse de séries temporelles, représentent des choix méthodologiques adaptés aux objectifs du projet. Ces méthodes offrent une base solide, tout en laissant entrevoir des perspectives d'amélioration pour des travaux futurs. Par exemple, l'intégration de seuils adaptatifs ou de techniques de séries temporelles avancées pourrait encore améliorer la capacité à généraliser des modèles.

Ces résultats montrent que les approches développées dans ce projet offrent des solutions concrètes et adaptables pour répondre aux besoins de surveillance environnementale, tout en jetant des bases solides pour des améliorations futures.

8.2 Innovations et Perspectives Futures

Cette méthode apporte des progrès significatifs grâce à son approche non supervisée, qui réduit la dépendance aux données annotées manuellement. Cela permet d'entraîner le modèle même sans disposer d'un jeu de données SAR spécifiquement annoté pour la déforestation. Cette flexibilité rend l'approche plus accessible et adaptée à des contextes variés.

En outre, elle se distingue par une grande flexibilité par rapport à des algorithmes comme CuSum, qui nécessitent des séries temporelles étendues d'images SAR pour détecter les changements. Ici, le modèle peut analyser une seule image SAR pour distinguer les zones forestières et non forestières, puis comparer les résultats entre deux images prises à des moments différents afin d'identifier les changements. Cette méthode est particulièrement utile pour des situations où les données temporelles sont limitées.

Le potentiel de cette méthode ne se limite pas à la déforestation en Amazonie. Elle est également capable de détecter des problèmes touchant des forêts déjà endommagées. Cela suggère qu'elle pourrait être utilisée pour identifier divers types de dégradations forestières, comme les incendies ou les coupes répétées. Cette adaptabilité en fait un outil précieux pour développer des solutions de surveillance flexibles et efficaces, capables de répondre aux principaux défis environnementaux mondiaux.

8.3 Limites et Améliorations Potentielles

Étant donné que ce stage était ma première expérience en apprentissage profond de manière autonome, je reconnais qu'il reste de nombreux aspects à améliorer, notamment en termes de méthodologie et de processus. La durée limitée du stage, de seulement six mois, a également rendu la finalisation du projet un peu précipitée.

Tout d'abord, dans la modification de la structure du modèle, les expérimentations que j'ai réalisées se sont appuyées principalement sur l'observation de la courbe de perte et sur l'analyse visuelle des reconstructions des images du jeu de validation. Les choix relatifs aux types, quantités et positions des modules ajoutés ont été faits en fonction de mes connaissances actuelles, ce qui ne garantit pas une rigueur scientifique complète. Bien que la structure actuelle soit celle ayant donné les meilleurs résultats parmi mes essais, je suis convaincu qu'elle peut encore être améliorée.

Pour l'amélioration des hyperparamètres, même si j'ai intégré l'outil automatique Optuna, le manque de temps et l'accès limité aux ressources de calcul ont restreint mes expérimentations à un petit nombre de cycles d'entraînement. Ainsi, les combinaisons d'hyper-paramètres que j'ai obtenues représentent uniquement un meilleur résultat local plutôt qu'un meilleur résultat global.

Pour le post-traitement des résultats, la méthode de classification des anomalies est encore expérimentale. Déterminer un seuil fixe adapté aux besoins de généralisation du modèle s'est avéré difficile. Actuellement, j'ai combiné une approche basée sur des méthodes de clustering et d'analyse de séries temporelles, mais je pense qu'il reste encore de nombreuses pistes pour améliorer ces étapes.

Pour l'évaluation des performances du modèle, un défi majeur réside dans l'absence de métriques standardisées et de données de référence pour valider les résultats. Actuellement, je m'appuie sur deux sources principales : les annotations que j'ai réalisées manuellement et les sorties de l'algorithme CuSum. Cependant, ces deux références présentent des limites importantes. Les annotations manuelles ne disposent pas d'une mesure objective de précision, ce qui rend difficile leur évaluation. Quant à l'algorithme CuSum, il souffre d'un rappel relativement faible, limitant ainsi sa fiabilité en tant que standard pour évaluer les performances de mon modèle. Ces contraintes réduisent la possibilité de valider de manière rigoureuse et standardiser les résultats obtenus.

9. Conclusion

Pendant ces six mois de stage, j'ai eu une expérience très enrichissante qui m'a permis de travailler sur un projet intéressant et complexe. En participant à ce projet, j'ai non seulement acquis de nombreuses compétences techniques, mais j'ai également développé mes compétences analytiques et scientifiques, telles que la programmation, le traitement et l'analyse des données, ainsi que l'entraînement des modèles de l'apprentissage profond.

Ce projet s'est articulé autour de plusieurs étapes pratiques : de l'apprentissage initial des bases de la télédétection et de l'apprentissage profond, à la création manuelle des ensembles de données, en passant par la conception de l'architecture du modèle et la finalisation du modèle. Chaque étape a nécessité une méthode rigoureuse et de l'initiative. Ce processus a non seulement renforcé mon intérêt pour les applications de l'intelligence artificielle, mais il m'a aussi permis d'améliorer considérablement mes compétences personnelles, comme la gestion de projet, le travail en équipe et la résolution de problèmes complexes.

Au-delà des aspects techniques, ce projet m'a permis d'approfondir ma compréhension des défis environnementaux liés à la déforestation. Il m'a fait prendre conscience de l'impact potentiel des outils technologiques sur la gestion durable des ressources naturelles.

Je suis très reconnaissant pour cette opportunité d'apprentissage unique. Cela a contribué à mon développement personnel et a établi de solides bases pour ma carrière. Je suis convaincu que les compétences et connaissances acquises au cours de ce stage auront un effet positif et durable sur mon avenir professionnel.

10. Bibliographie

- Kalamandeen, M., Gloor, E., Mitchard, E., Quincey, D., Ziv, G., Spracklen, D., ... & Galbraith, D. (2018). *Pervasive rise of small-scale deforestation in Amazonia*. *Scientific reports*, 8(1), 1-10.
- Ola, O., Menapace, L., Benjamin, E., & Lang, H. (2019). *Determinants of the environmental conservation and poverty alleviation objectives of Payments for Ecosystem Services (PES) programs*. *Ecosystem services*, 35, 52-66.
- Demarchi, G., Subervie, J., Catry, T., & Tritsch, I. (2023). *Using publicly available remote sensing products to evaluate REDD+ projects in Brazil*. Available at SSRN 4090218.
- Manogaran, G., & Lopez, D. (2018). *Spatial cumulative sum algorithm with big data analytics for climate change detection*. *Computers & Electrical Engineering*, 65, 207-221.
- Ygorra, B., ..., C., Catry, T., ... & Riazanoff, S. (2021). *Monitoring loss of tropical forest cover from Sentinel-1 time-series: A CuSum-based approach*. *Int. Journal of Applied Earth Observation and Geoinformation*, 103, 102532.
- Bank, D., Koenigstein, N., & Giryes, R. (2020). *auto-encodeurs*. arXiv preprint arXiv:2003.05991.
- Shvetsova, N., Bakker, B., Fedulova, I., Schulz, H., & Dylov, D. V. (2021). *Anomaly detection in medical imaging with deep perceptual auto-encoders*. *IEEE Access*, 9, 118571-118583.
- Pfefer, A., Ygorra, B., Frappart, F., Demarchi, G., Pillot, B., Subervie, J., ... & Catry, T. (2024). *How Sentinel-1 timeseries can improve the implementation of conservation programs in Brazil*. *Remote Sensing Applications: Society and Environment*, 35, 101241
- Md Jelas, I., Zulkifley, M. A., Abdullah, M., & Spraggon, M. (2024). *Deforestation detection using deep learning-based semantic segmentation techniques: A systematic review*. *Frontiers in Forests and Global Change*, 7, 1300060.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). *Backpropagation Applied to Handwritten Zip Code Recognition*. *Neural Computation*, 1(4), 541-551.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, LNCS, vol. 9351, 234-241.
- Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). *Pyramid Scene Parsing Network*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2881-2890.

Lin, T. Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2117–2125.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS), 5998–6008.

Rahman, M. M., & Sumantyo, J. T. S. (2010). Mapping tropical forest cover and deforestation using synthetic aperture radar (SAR) images. Applied Geomatics, 2, 113–121.

Bouvet, A., Mermoz, S., Ballère, M., Koleck, T., & Le Toan, T. (2018). Use of the SAR shadowing effect for deforestation detection with Sentinel-1 time series. Remote Sensing, 10(8), 1250.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4), 541–551.

Deng, L. (2012). The MNIST database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, 29(6), 141–142.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv preprint arXiv:1505.04597.

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid Scene Parsing Network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2881–2890.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 936–944.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems, 30.

11. Glossaire

SAR: Synthetic Aperture Radar

VV: Vertical-Vertical, ce qui signifie que l'antenne SAR émet et reçoit les ondes électromagnétiques avec une polarisation verticale.

VH: Vertical-Horizontal, ce qui signifie que l'antenne SAR émet les ondes électromagnétiques avec une polarisation verticale et les reçoit avec une polarisation horizontale.

NICFI: Norwegian International Climate and Forest Initiative

ASF: Alaska Satellite Facility

QGIS: Quantum geographic information system

CuSum: cumulative sum algorithm

MSE: mean-square error

CNN: Convolutional Neural Networks

PPM: Pyramid Pooling Module

FPN: Feature Pyramid Network

GMM: Gaussian Mixture Model

IoU: Intersection over Union

12. Annexes

12.1 Annexe 1: Dépôt GitHub de “SAR Deforestation Detection autoencoder”

https://github.com/ZhangYiFan2003/SAR_Deforestation_Detection_Autoencoder

12.2 Annexe 2: Liste des Figures

Figure 1: Résultats de l'algorithme Cusum sur un échantillon de la zone étendue de l'image PlanetScope d'octobre 2021

Figure 2: Exemple d' Image SAR 2021.4

Figure 3: Exemple d' Image Optique 2021.4

Figure 4: Exemple d'Sorties Cusum 2020.10.11 - 2021.12.17

Figure 5: Organigramme du processus de développement du projet

Figure 6: Masque de forêt/non forêt

Figure 7: Annotation manuelle

Figure 8: images optiques dans QGIS

Figure 9: téléchargement de NICFI

Figure 10: Architecture de l'auto-encodeur

Figure 11: Distribution des valeurs de MSE

Figure 12: Visualisation des résultats préliminaires

Figure 13: Clustering par GMM

Figure 14: Classification par séries chronologiques

Figure 15: La plage de valeurs des images SAR

Figure 16: Les résultats d'une mauvaise normalisation

Figure 17: Projection manuelle

Figure 18: Projection avec geo_parameterize

Figure 19: Exemple de détection - résultat final, PlanetScope capturé le 19 septembre 2022

Figure 20: Exemple de courbe de fonction de perte générée par Tensorboard

Figure 21: Comparaison du résultat de détection et d'annotation masquées

实习报告中文简介

本次实习于2024年9月至2025年2月在法国蒙彼利埃的法国发展研究院 (Institut de Recherche pour le Développement, IRD) 进行, 具体所属单位为混合研究单元UMR Espace-Dev, 研究主题为利用非监督学习方法实现卫星雷达影像中微小尺度森林砍伐的自动检测。

森林砍伐是全球环境保护面临的严峻问题, 尤其是在巴西亚马逊热带雨林地区, 这里的生态系统极为丰富, 对全球气候调节具有重要意义。然而, 由于气候和季节的影响, 传统光学卫星影像观测经常受到云层的干扰, 难以进行连续监测。与之相比, 合成孔径雷达 (SAR) 影像因其不受天气状况限制, 能够全天候、全天时获取地表信息, 已逐渐成为森林砍伐监测的重要技术手段之一。

现有的SAR影像分析方法中, 广泛使用的累积和 (CuSum) 算法在局部检测中表现出一定的有效性, 但仍存在明显的局限性, 例如响应延迟较长、误报率高以及处理大范围区域时的计算复杂度过高等问题。因此, 开发更加高效、准确、实时的检测方法成为研究的热点问题。本实习项目所在的研究小组提出了名为M2D2 (Machine learning-based Micro Déforestation Détection) 的研究计划, 旨在利用基于深度学习的非监督技术, 实现对微尺度森林砍伐事件的自动化、精细化监测。

具体而言, 本人实习期间的主要任务为设计、实现并优化一个基于自编码器 (Autoencoder) 的深度学习模型, 应用于Sentinel-1 SAR卫星影像中森林砍伐的自动异常检测。该任务的具体步骤和方法分为以下几个阶段:

首先, 数据的收集与预处理阶段。项目使用了欧洲航天局 (ESA) 的Sentinel-1卫星提供的SAR影像以及由PlanetScope卫星提供的光学影像, 以确保数据来源的多样性和可靠性。这些数据以巴西亚马逊地区为研究区域, 通过QGIS软件进行人工标注, 建立了详细的森林砍伐区域数据库, 包含多个时间段的影像标注信息, 确保了训练模型所需的数据精度。

在数据预处理阶段, 主要采取了多种优化技术, 包括SAR影像的去噪处理、边缘杂波的有效去除、基于数字高程模型 (DEM) 的地形校正、图像裁剪和重采样等, 保证了数据的

统一性和模型输入数据的质量。此外，利用Rasterio、Geopandas和PyroSAR等开源的地理空间数据处理工具，实现了数据的高效处理与存储管理。

其次，深度学习模型的设计与优化阶段。自编码器模型由编码器和解码器两个主要部分组成，编码器负责对原始影像数据进行特征压缩与提取，解码器则基于压缩特征进行影像重建。模型通过对正常未发生砍伐区域的影像进行学习，能够精确重建未被砍伐区域的数据，而砍伐区域在重建过程中会表现出明显的差异，即较高的重构误差。

模型的设计中采用了卷积神经网络(CNN)作为基础架构，并引入了多种先进的模块和策略以提升其性能。例如，为解决深层网络中梯度消失问题，引入了残差网络(Residual Network)的设计；为提升网络对不同尺度特征的感知能力，增加了跳跃连接(skip connections)和特征金字塔网络(FPN)；为增强网络对空间上下文的理解，集成了自注意力机制(Self-Attention)。

在训练过程中，为了优化模型的泛化能力和防止过拟合，采用了Optuna自动化调参框架对模型的学习率、权重衰减、学习率调整策略(scheduler)等超参数进行了系统化的探索和优化。同时，设置了提前停止(early stopping)策略，有效避免了训练过程中的过拟合问题。

接下来，利用训练好的自编码器模型实现异常检测与分析。通过计算模型对新影像数据的重构误差，生成异常分布图，以有效识别可能的砍伐区域。为进一步提升检测的准确性，引入了高斯混合模型(Gaussian Mixture Model, GMM)对重构误差数据进行聚类分析，以自动识别和区分森林与非森林区域，减少误报并提高检测精确性。

为了验证模型检测结果的准确性，进行了严格的量化与定性分析，包括与人工标注数据、现有算法输出(如CuSum算法)进行对比评估，结果显示，自编码器模型显著提高了检测精度，且能够更快速地发现砍伐事件。

在空间数据处理与结果展示方面，通过实现GIS坐标校正与空间投影转换技术，将检测结果以标准化的Shapefile文件形式输出，保证了模型检测结果的空间准确性和可视化便利性。实习期间，还开发了相关自动化处理工具与脚本，简化了数据处理与结果分析流程。

实习过程中，本人深入理解和掌握了遥感数据处理、深度学习模型设计与训练、异常检测方法、空间数据分析与可视化等多个前沿技术领域，不仅提高了自身的科研能力和实际问题解决能力，也为研究团队提供了有价值的技术支持。

尽管项目已取得显著成果，但仍存在一些待改进之处，如数据标注的准确性和多样性问题、模型在复杂地形或不同生态环境下泛化能力的问题。未来工作将围绕这些问题进一步展开，尝试扩展更多不同生态环境的数据集，优化模型的结构与性能，探索多源遥感数据的融合分析方法，全面提升森林砍伐自动检测技术的实际应用能力。

此次实习不仅极大地丰富了本人在遥感与深度学习领域的理论知识与实践技能，也为未来从事生态环境保护领域的研究工作奠定了坚实基础，提供了重要的实践经验和技术创新。