Apports des méthodes d'apprentissage profond pour la reconnaissance automatique des modes d'occupation des sols et d'objets par télédétection en milieu tropical

Guillaume Rousset

▶ **To cite this version:**

Guillaume Rousset. Apports des méthodes d'apprentissage profond pour la reconnaissance automatique des modes d'occupation des sols et d'objets par télédétection en milieu tropical. Traitement des images [eess.IV]. Université de la Nouvelle-Calédonie, 2021. Français. NNT : 2021NCAL0006 . tel-03725622

HAL Id: tel-03725622
https://unc.hal.science/tel-03725622

Submitted on 18 Jul 2022

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

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

École doctorale n° 469 : École Doctorale du Pacifique

THÈSE

pour obtenir le grade de docteur délivré par

L'Université de Nouvelle-Calédonie

Discipline : Informatique

Apports des méthodes d'apprentissage profond pour la reconnaissance automatique des modes d'occupation des sols et d'objets par télédétection en milieu tropical

présentée et soutenue publiquement par

Guillaume Rousset

le 22 Septembre 2021

Directeur de thèse : **Dominique Simpelaere**
Co-directeur de thèse : **Morgan Mangeas**
Encadrant de thèse : **Marc Despinoy**

Jury

Dino Lenco	Chercheur HDR à l'INRAE	Rapporteur
Benoît Stoll	MCF HDR à l'Université de Polynésie	Rapporteur
Agnès Begue	DR au CIRAD	Examinatrice
Cécile Dupouy	DR à l'IRD	Examinatrice
Samira El Yacoubi	Professeure à l'Université de Perpignan	Examinatrice

Apports des méthodes d'apprentissage profond pour la reconnaissance automatique des modes d'occupation des sols et d'objets par télédétection en milieu tropical

Résumé

Le paysage néo-calédonien change rapidement avec le développement de nouveaux projets miniers, l'intensification de l'urbanisation et les impacts d'événements climatiques extrêmes comme les cyclones. Avec la démocratisation et l'accumulation des données satellite et l'avènement des méthodes d'intelligence artificielle, la mise en place de méthodes automatiques de détection devient un outil incontournable pour documenter et surveiller ces changements à l'échelle du territoire de façon régulière, rapide et objective. Parmi ces méthodes, l'apprentissage profond a montré des résultats performants sur des problématiques complexes, notamment sur le traitement d'images à l'aide de ces réseaux de neurones denses convolutionnels.

En tenant compte des contraintes liées au traitement de l'imagerie satellite et des problèmes liés aux algorithmes d'apprentissage, l'objectif de la thèse est multiple : contribuer à l'adaptation des techniques d'apprentissage profond à des problématiques de télédétection sur plusieurs points clés de la chaîne de traitement ; estimer les performances de ces techniques par rapport aux méthodes communément utilisées dans le domaine de la télédétection ; et développer des méthodes automatiques de détection pour délivrer des indices fiables à toute exploitation d'une imagerie satellitaire.

Cette thèse s'est concentrée sur trois applications : 1) la détection de la couverture et de l'usage des sols sur des données à très haute résolution ; 2) la détection de la couverture des sols en Nouvelle-Calédonie à une fréquence annuelle sur des données à haute résolution ; 3) et la détection de palmiers dans la région Pará du Brésil à l'aide de données simulées informatiquement.

Pour la première application, un jeu de données de référence basé sur les données du satellite SPOT 6 a été créé manuellement et mis à disposition de la communauté scientifique pour comparer les techniques de détection des classes d'occupation des sols en milieu tropical insulaire. Les réseaux de neurones denses affichent de meilleures performances notamment dans le cadre de la détection de l'usage des sols qui nécessite un niveau plus élevé de conceptualisation de l'environnement.

Pour la deuxième application, une chaîne de détection automatique de la couverture des sols, basée sur un réseau de neurones dense alimenté par des données Sentinel-2, a été réalisée. Ces couvertures sont comparées aux couvertures obtenues par des méthodes semi-automatiques en province Sud de la Nouvelle-Calédonie. Le modèle offre des performances égales sur quelques zones tests, mais des données terrain supplémentaires sont requises pour conforter la fiabilité sur l'ensemble du territoire néo-calédonien.

Enfin, pour la dernière application, l'originalité du travail de recherche a consisté à tester l'apport dans la base d'apprentissage d'images satellites de synthèse. Pour cela des images du palmier ont été construites à partir d'un modèle de transfert radiatif. L'utilisation de ces images de synthèse en complément des images Pléiades a permis d'améliorer significativement la précision globale des modèles.

Contributions of deep learning methods for automatic recognition of land use and land cover patterns and objects by remote sensing in tropical environments

Abstract

The New Caledonian landscape is changing rapidly with the development of new mining projects, the intensification of urbanization and the impacts of extreme climatic events such as cyclones. With the democratization and accumulation of satellite data and the advent of artificial intelligence methods, the implementation of automatic detection methods is becoming an essential tool for documenting and monitoring these changes to the scale of a territory in a regular, rapid and objective manner. Among these methods, deep learning has shown effective results on complex problems, especially on image processing using dense convolutional neural networks.

Considering the constraints related to the processing of satellite imagery and the problems related to learning algorithms, the objective of the thesis is multiple : to contribute to the adaptation of deep learning techniques to remote sensing problems on several key points of the processing chain ; to estimate the performance of these techniques compared to the methods commonly used in the field of remote sensing ; and to develop automatic detection methods to deliver reliable indices to any exploitation of satellite imagery.

This thesis focused on three applications : 1) land cover and land use detection on very high resolution data ; 2) the detection of land cover in New Caledonia at an annual frequency on high resolution data ; 3) and the detection of palm trees in the Pará region of Brazil using computer simulated data.

For the first application, a reference dataset based on SPOT 6 satellite data was manually created and made available to the scientific community in order to compare land cover class detection techniques in tropical island environments. Dense neural networks show better performance, especially in the context of land cover detection which requires a higher level of conceptualization of the environment.

For the second application, an automatic land cover detection chain, based on a dense neural network fed by Sentinel-2 data, has been realized. This coverage is compared to coverage obtained by semi-automatic methods in the South Province of New Caledonia. The model performs equally well over a few test areas, but additional field data are required to confirm the reliability over the whole New Caledonian territory.

Finally, for the last application, the originality of the research work consisted of testing the contribution of synthetic satellite images in the learning base. For this purpose, images of the palm tree were constructed from a radiative transfer model. The use of these synthetic images in addition to the Pleiades images significantly improved the overall accuracy of the models.

Avant-Propos et Remerciements

Les travaux de recherche développés durant cette thèse ont été réalisés au sein de l'UMR ESPACE-DEV et l'équipe ISEA de l'Université de la Nouvelle-Calédonie avec un support financier issu de la Province Sud. L'ensemble de cette thèse a été accompli au sein des locaux de l'Institut de Recherche pour le Développement de Nouvelle-Calédonie, dans les locaux de l'UMR ESPACE-DEV.

Je tiens à remercier toutes les personnes qui m'ont soutenue et aidée pendant ces 4 années de thèse et pendant la rédaction de ce manuscrit.

Je voudrais en premier lieu remercier, mes encadrants de thèse, Dominique Simpelaere, Morgan Mangeas et Marc Despinoy qui m'ont accompagné tout au long de ce processus, et m'ont aidé jusqu'au bout pour que ce manuscrit soit écrit malgré leur charge de travail. Je remercie également Konrad Shindler pour ses corrections sur mon article scientifique et pour ses lumières et son expérience sur mon sujet de thèse.

Je remercie la Province Sud de Nouvelle-Calédonie, qui m'a financé pendant 3 ans pour réaliser cette thèse ; l'IRD et l'ESPACE-DEV, qui m'ont accueilli pendant toute la durée de ma thèse dans leurs locaux ; et l'école doctorale du Pacifique dans laquelle je présente ce manuscrit.

Je tiens à remercier tous mes collègues, stagiaires, gens de passage et amis, qui m'ont vu végéter derrière mon bureau, et qui ont réussi à me faire sortir de temps en temps de mon antre. La liste serait trop longue pour tous les citer, et je m'excuse par avance si votre nom n'apparaît pas : Chloé, Solène, Solène (Frida), Chumil, Lucie, Camille, Catherine, Sabrina (une chanson me vient soudain : "Où sont les hommes"). Je tiens à aussi remercier mes amis : Oona, Hiroki, Fanny, Jeffrey, Ophélie, Caroline, Florian et Jean qui ont su me faire oublier le travail, malgré la question sempiternelle "Tu en es où exactement de ta thèse ?" ...

Mes parents, bien sûr, qui m'ont encouragé pendant les premières années de thèse. Ma soeur, qui n'attend heureusement pas mon initiative pour qu'on se voie de temps en temps. Et enfin Laura, qui m'a soutenu jusqu'au bout, quitte à me bousculer pour que je me mette au travail.

TABLE DES MATIÈRES

Table des matières **7**

Liste des figures **9**

Liste des tableaux **12**

1 Introduction **14**
1.1 Le contexte général . . . 15
1.2 Le contexte spécifique de la thèse . . . 17
1.3 Matériels et Logiciels . . . 19
1.4 Contribution . . . 20
1.5 Plan . . . 21

2 Le Deep Learning **22**
2.1 L'apprentissage d'un réseau de neurones . . . 23
2.2 Les réseaux de neurones convolutifs . . . 33
2.3 Amélioration de l'apprentissage . . . 39
2.4 Pixel classification . . . 44
2.5 Segmentation sémantique . . . 52

3 Classification des Occupations du Sol **60**
3.1 Introduction . . . 61
3.2 Les Données . . . 66
3.3 Méthodes . . . 77
3.4 Analyses et Performances des modèles . . . 85
3.5 Discussions et conclusions . . . 97

4 Vers une Détection de Changement du Sol **108**
4.1 Introduction . . . 109

4.2 Adaptation et création de données 111
4.3 Analyse et discussion autour des résultats 123
4.4 Conclusions 130

5 Détection du Babaçu au Brésil **132**
5.1 Introduction 133
5.2 Données et méthodes 138
5.3 Analyse et performances des modèles 146
5.4 Conclusions 153

6 Conclusions et Perspectives **154**
6.1 Une cartographie de l'occupation du sol automatisée 155
6.2 Une détection automatique d'espèces végétales avec des données simulées 159

A Références des illustrations **162**

B Références des images satellites **163**

LISTE DES FIGURES

2.1 Représentation schématique générale d'un réseau de neurones 23
2.2 Fonctionnement simple d'un perceptron 24
2.3 Exemple simple de l'utilisation d'un perceptron 24
2.4 Exemple de fonctions de transfert 25
2.5 Jeu de données de chiffres manuscrits 25
2.6 Architecture en réseau de neurones reconnaissant les chiffres manuscrits 26
2.7 Illustration d'un réseau neuronal convolutif, LeNet-5 33
2.8 Détection de contours à l'aide d'un filtre 34
2.9 Exemple du fonctionnement d'une couche de convolution 35
2.10 Exemple de conservation des dimensions d'une image 36
2.11 Exemple du fonctionnement de la fonction de transfert ReLU 37
2.12 Exemple du fonctionnement d'une couche de *Max Pooling* 37
2.13 Exemple du fonctionnement du *Drop-out* 39
2.14 L'architecture AlexNet 45
2.15 L'architecture ResNet-34 47
2.16 Représentation d'un « Dense Block » de 5 couches 49
2.17 L'architecture DenseNet-121 50
2.18 L'architecture FCN 53
2.19 Exemple des sorties des différentes architectures FCN 53
2.20 Détail des différentes variations de l'architecture FCN 54
2.21 L'architecture SegNet 56
2.22 Exemple de fonctionnement d'une couche Atrous 57
2.23 L'architecture DeepLabV3+ 58

3.1 Carte de la Nouvelle-Calédonie et de ses provinces 61
3.2 Carte géologique simplifiée de la Nouvelle-Calédonie 62
3.3 Carte de la province Sud situant les 5 zones d'études 72

3.4 Images satellites SPOT 6 des 5 zones sélectionnées 74
3.5 Cartographie de la couverture du sol pour les 5 zones d'études 75
3.6 Cartographie de l'usage du sol pour les 5 zones d'études 76
3.7 Exemple d'application de néocanaux 79
3.8 Exemple d'application de textures Haralick 80
3.9 Exemple de fonctionnement d'une fenêtre coulissante 83
3.10 Importance des variables d'un modèle XGBoost pour la classification de la couverture du sol 87
3.11 Importance des variables d'un modèle XGBoost pour la classification de l'usage du sol 89
3.12 Importance des variables du modèle XGBoost pour la couverture du sol sur le jeu de données *RVBPIRNEO* 90
3.13 Importance des variables du modèle XGBoost pour la couverture du sol sur le jeu de données *RVBPIRNEO* 90
3.14 Résultats de DeepLab pour la couverture du sol sur la commune de Thio 98
3.15 Illustration du débordement de la détection de bâtiments 99
3.16 Illustration de la confusion dans la classification de la couverture du sol entre les végétations de différentes densités 99
3.17 Illustration de la précision des modèles de machine learning par rapport à l'étiquetage plus grossier d'un opérateur humain 100
3.18 Résultats de DeepLab pour l'usage du sol sur la commune de Thio . . 101
3.19 Résultats de l'architecture DeepLab sur la commune de Thio pour la tâche d'usage des sols 103
3.20 Couverture du sol de l'intégralité de la Nouvelle-Calédonie à l'aide de l'architecture DeepLab 106
3.21 Usage du sol de l'intégralité de la Nouvelle-Calédonie à l'aide de l'architecture DeepLab 107

4.1 Exemple de deux images satellites Sentinel-2 inutilisables 112
4.2 MOS de 2016 de la province Sud 114
4.3 Illustration des différentes classes du MOS 116
4.4 Grille de découpage des données satellites. En vert les images utilisées pour l'apprentissage. En rouge les images utilisées pour le test. 121
4.5 Adaptation de la classification manuelle et du MOS à la nouvelle nomenclature 125

4.6 Cartographie de la couverture du sol de la province Sud des années 2016 à 2020 à partir de Deeplab 128
4.7 Cartographie de la couverture du sol de Nouvelle-Calédonie en 2016 et 2020 à partir de Deeplab 131

5.1 Image satellite Pléiades de la région agricole de Benfica 133
5.2 Les 3 derniers cycles de vie du babaçu 134
5.3 Images DART d'une section de forêt 139
5.4 Images DART d'un palmier au milieu d'une section de forêt 139
5.5 Illustration d'images Pléiades présentant les différentes situations possibles de configuration pour la détection d'un babaçu 141
5.6 Image Pléiades où le centre de babaçus et des zones sans babaçus sont pointés 142
5.7 Graphique des résultats de DenseNet et AlexNet pour la détection du babaçu à partir de données photo-interprétées 147
5.8 Graphique de comparaison du F1-score de l'architecture DenseNet en fonction du nombre d'entrées DART 148
5.9 Détection du babaçu sur une image Pléiades 152

6.1 Illustration de la fusion des sorties de deux architectures de Deep Learning 158
6.2 Illustration de la fusion des encodeurs de deux architectures de Deep Learning 158
6.3 Illustration de deux méthodes de détection d'un objet 159

LISTE DES TABLEAUX

2.1 Configuration des couches de l'architecture AlexNet 45
2.2 Table des paramètres des architectures ResNet 48
2.3 Table des paramètres des architectures DenseNet 51

3.1 Nomenclature de la couverture des sols 68
3.2 Nomenclature hiérarchisée de l'usage des sols 69
3.3 Détails des 5 zones d'études sélectionnées 72
3.4 Détails des différents néocanaux utilisés 78
3.5 Synthèse des jeux de données utilisées en apprentissage 84
3.6 Résultats du modèle XGBoost pour la classification de la couverture du sol 86
3.7 Résultats du modèle XGBoost pour la classification de l'usage du sol 88
3.8 Résultats de XGBoost et du Deep Learning pour la couverture du sol sur le jeu de données *RVBPIR* 91
3.9 Résultats de XGBoost et du Deep Learning pour la couverture du sol sur le jeu de données *RVBPIR* 93
3.10 Résultats de DenseNet et DeepLab pour la classification de la couverture du sol à partir des canaux bruts et de néocanaux 94
3.11 Résultats de DenseNet et DeepLab pour la classification de l'usage du sol à partir des canaux bruts et de néocanaux 95
3.12 Résultats de DenseNet et DeepLab pour la classification de l'usage du sol à partir des canaux bruts et de la couverture du sol 96
3.13 Matrice de confusion de la couverture du sol créée par DeepLab . . . 97
3.14 Matrice de confusion de l'usage du sol créée par DeepLab 102

4.1 Nomenclature hiérarchisée du MOS de Nouvelle-Calédonie 115
4.2 Nomenclature de la couverture des sols 119

4.3 Précision utilisateur de la matrice de confusion comparant le MOS et la classification manuelle . . . 123
4.4 Précision producteur de la matrice de confusion comparant le MOS et la classification manuelle . . . 124
4.5 Précision utilisateur de la matrice de confusion comparant le MOS au résultat de DeepLab . . . 126
4.6 Précision producteur de la matrice de confusion comparant le MOS au résultat de DeepLab . . . 126

5.1 Configuration des couches de l'architecture AlexNet pour la détection de babaçu . . . 143
5.2 Exemple d'une matrice de confusion standard . . . 144
5.3 Résultats de DenseNet et AlexNet pour la détection du babaçu à partir de données photo-interprétées . . . 146
5.4 Résultats de l'architecture DenseNet pour la tâche de détection du babaçu avec des données DART . . . 149

B.1 Détails des images SPOT 6 utilisées dans l'étude . . . 163
B.2 Nom des images Sentinel-2 utilisées . . . 164

CHAPITRE 1

INTRODUCTION

"Permettez-moi de vous présenter mon ennemi intime..."

Jules Verne

Sommaire

1.1 Le contexte général 15
1.2 Le contexte spécifique de la thèse 17
1.3 Matériels et Logiciels 19
1.4 Contribution 20
1.5 Plan 21

1.1 Le contexte général

Avec les récentes avancées scientifiques et technologiques, il est possible aujourd'hui d'accéder facilement à un flux d'informations toujours plus important. Des termes comme « Big Data » et « Data Mining » sont apparus pour désigner les problématiques entourant le stockage, la mise en forme et l'exploitation de cet afflux massif de données. Le domaine spatial, plus précisément l'observation de la terre par satellite, participe activement à cette expansion. Les satellites fournissent des images de la surface de la Terre [1] à des résolutions spatiales, spectrales et temporelles de plus en plus fines [2], demandant des ressources matérielles toujours plus considérables (capacités de stockage, vitesse de calcul, taux de transfert, mémoire vive, etc.).

La télédétection est un domaine de la recherche qui a pour objectif, à partir de données satellites brutes, de caractériser et de comprendre des mécanismes environnementaux complexes sur l'intégralité du globe terrestre [3], fournissant des indices d'estimation nécessaires dans de nombreux domaines de la recherche [4]. L'extraction de nouvelles connaissances, à partir de données satellites et de ces produits dérivés, est devenue un enjeu majeur de nos jours du fait d'une forte demande liée à la surveillance de l'environnement et au suivi des impacts, dus notamment aux changements climatiques et aux activités humaines [5].

À partir d'images satellites, il est possible d'extraire plusieurs types d'informations, au niveau d'un pixel (information radiométrique), d'un objet (information géométrique) ou d'une structure (information texturale par exemple). Les méthodes les plus simples en télédétection s'arrêtent au premier niveau, c'est-à-dire qu'elles ne prennent pas en compte la relation complexe existant entre un pixel et ses voisins [6]. Les techniques dites de détection d'objets, c'est-à-dire un ensemble de pixels connexes différenciés par leur valeur spectrale, mais qui peuvent être assemblés pour former un objet (ex. : un bâtiment), ont su montrer leurs performances en télédétection [7, 8, 9, 10].

Cependant, des limitations apparaissent lorsqu'une connaissance sémantique est nécessaire pour répondre à certaines problématiques de télédétection [11]. Par exemple, une tâche simple de classification liée à la couverture biophysique du sol est possible, mais pour déterminer le type d'usage anthropique de ce même sol, des connaissances exogènes sont nécessaires pour interpréter la donnée nécessitant un haut degré d'abstraction. L'expertise humaine ou ce qui pourrait s'apparenter à de l'intelligence artificielle, même si ce terme reste à définir, est alors indispensable pour discriminer les

types d'usages en incluant des informations liées au contexte (spécificités de la région) et au paysage (au sens des diverses « couches » de l'information géographique : aménagements, nature géomorphologique et biopédologique des sols, habitats, activités humaines répertoriées, etc.).

Avec l'abondance des données spatialisées [12] et le besoin de méthodes automatiques pour créer des indices complexes, les techniques de « machine learning » (approches mathématiques et statistiques créant des modèles de prédiction à partir de données) ont su évoluer et se faire une place dans le paysage de la télédétection [13]. De plus, l'émergence de nombreux jeux de données, mis à disposition sur les bancs de tests au public scientifique, a poussé de plus en plus loin les performances dans la détection sémantique d'objets [14]. Parmi ces méthodes de machine learning, le « Deep Learning » possède un fort potentiel dans le cadre des classifications d'images de télédétection [15, 16, 17].

Les architectures en réseaux de neurones dites profondes ont fait leur apparition en 2006. Plus communément appelée architecture de Deep Learning, cette nouvelle méthode a révolutionné les méthodes de machine learning [18] avec sa capacité à organiser les données sur plusieurs niveaux d'abstractions pour créer des liens entre plusieurs variables et faire émerger des relations complexes [19]. Les résultats obtenus par une architecture de référence, AlexNet [20], ont démontré que la profondeur d'un réseau de neurones était essentielle pour atteindre de bonnes performances. Néanmoins, si certaines techniques de machine learning classiques fonctionnent sur de petits jeux de données, les architectures de Deep Learning ont besoin de millions d'images ainsi que d'une ressource informatique importante pour un apprentissage idéal. Cette puissance de calcul est de plus en plus accessible de nos jours grâce au développement des GPU (de l'anglais *Graphics Processing Units*) [21] qui permettent de faire fonctionner des architectures complexes et très profondes [22].

S'il est possible d'extraire de nombreuses informations à l'aide de la télédétection, les techniques de machine learning et de Deep Learning ont su ajouter un niveau supplémentaire de fiabilité en abordant des problématiques plus complexes. Ainsi, la littérature récente apporte beaucoup d'exemples d'application du Deep Learning au domaine de la télédétection pour des problématiques complexes dans le domaine du suivi environnemental. On peut citer : le suivi de la fonte des glaciers [23], les risques de raz de marée [24], les dommages des cyclones [25], la détection d'objets [26, 27, 28] ou de modes d'occupation des sols [29, 30, 31] ou encore la mise au point d'indices d'expansion urbaine [32, 33].

1.2 Le contexte spécifique de la thèse

L'étude de l'environnement terrestre par le traitement d'imagerie satellitaire est une problématique complexe du fait des nombreux paramètres à prendre en compte. Entre le signal brut enregistré par les capteurs satellites et les images délivrées, plusieurs étapes de traitements sont nécessaires : traitements corrigeant les biais inhérents aux capteurs, les corrections liées à la composition en aérosols de l'atmosphère, ou encore des traitements liés à la luminescence, l'altitude et l'angle du capteur [34]. Ces traitements ne sont pas parfaits et des bruits subsistent pouvant perturber les modèles les plus sensibles. La robustesse des modèles est alors un critère important pour la production de résultats fiables.

Le Deep Learning a fait récemment de grands progrès dans le domaine de la télédétection en raison de son haut degré d'abstraction de la donnée, lui permettant d'extraire les informations essentielles d'un environnement en évolution rapide [35]. De nombreuses méthodes appliquées au Deep Learning se sont créées, notamment certaines permettant d'éviter le phénomène de « surapprentissage » des modèles et ainsi garantir leur robustesse [36, 37]. Le Deep Learning est majoritairement utilisé dans la reconnaissance d'images, allant de la caractérisation des sols à la détection d'objets [13, 38].

Cette technique est particulièrement adaptée pour la surveillance automatique d'un environnement complexe telle que celui de la Nouvelle-Calédonie. Cette île tropicale du Pacifique Sud dispose d'un relief montagneux divisant l'île en deux zones nommées « côte Est » et « côte Ouest », composées de nombreux types de sols [39]. La géologie et la topologie particulière de l'île offrent de nombreux milieux hétérogènes fragmentés qui font de l'île un « hotspot » de biodiversité mondiale [40]. En plus de ces particularités, la Nouvelle-Calédonie est soumise à des pressions naturelles et anthropiques fortes [41] : cyclones, incendies [42] ou encore activités minières entraînant des phénomènes d'érosion importants qui doivent faire l'objet d'une surveillance accrue.

Une cartographie de l'occupation des sols sur l'ensemble du territoire a été réalisée en 2014 (à partir de données de 2010 [43]). Bien que faite à une résolution spatiale de l'hectare, elle est utilisée pour calculer l'impact environnemental et l'évolution de la forêt tropicale sur l'ensemble du territoire. Ce travail, produit pour le compte du gouvernement de Nouvelle-Calédonie par l'intermédiaire de l'Observatoire de l'Environnement de Nouvelle-Calédonie (OEIL), a été un projet demandant la mobilisation d'un opérateur humain pour classifier scrupuleusement chaque parcelle du territoire.

Cette carte d'occupation des sols, si elle est régulièrement actualisée, devient un Mode d'Occupation des Sols (MOS), permettant de faire état du changement d'affectation des sols. Il est cependant difficile d'établir une telle cartographie tous les ans du fait des coûts financiers et humains que nécessite cette méthode. Avec les satellites documentant régulièrement le territoire de Nouvelle-Calédonie, l'application des techniques de Deep Learning au contexte calédonien est possible. Cependant, l'ensemble des études sur l'occupation du sol et sur les objets se focalise majoritairement sur les zones continentales [14] et quasi exclusivement à l'environnement urbain, peu étendu en Nouvelle-Calédonie.

Par ailleurs, l'identification d'objets au sol, à partir de données très haute résolution spatiale (métrique), est également un enjeu majeur de la télédétection et du suivi environnemental [44, 45]. Appliquée aux techniques de Deep Learning, de nombreuses détections d'objets sont rendues possibles avec d'excellentes performances [46], comme la détection de routes [47, 48, 49], de bâtiments [32, 50], de voitures [51], ou encore d'arbres [52, 53]. La détection du palmier babaçu par exemple, est un enjeu majeur de la région de Pará au Brésil [54]. Cette plante est une source économique pour les populations les plus pauvres de cette région [55] et des études ont été réalisées pour la reconnaissance automatique de ce palmier par imagerie satellitaire [56].

Ces techniques automatiques ont été optimisées pour fonctionner dans les pâturages, mais ne sont pas applicables pour le milieu forestier du fait de la complexité de sa détection dans la canopée [57]. Les méthodes de Deep Learning seraient applicables sous condition d'une base de données de terrain conséquente, permettant d'aboutir à un apprentissage convenable. L'ajout d'images spatialisées simulées, imitant une observation satellite et dont la composition serait complètement maîtrisée, à ce jeu d'apprentissage, serait une solution efficace pour pallier le manque de données. Ce travail n'a encore jamais été réalisé à ce jour, mais pourrait devenir un enjeu majeur de la télédétection par machine learning [58]. Dans le contexte de la Nouvelle-Calédonie, un tel travail pourrait être exporté sur d'autres espèces d'arbres telles que les arbres kaoris, qui ont été grandement impactés par l'exploitation forestière au siècle précédent [59, 60].

La télédétection systématique et l'utilisation de méthodologies de machine learning, comme le Deep Learning, pourraient fournir aux décideurs du pays toutes les informations nécessaires pour des prises de décisions, notamment aider à une meilleure planification de l'extension urbaine ainsi qu'une meilleure surveillance et préservation de l'environnement unique du territoire de la Nouvelle-Calédonie.

1.3 Matériels et Logiciels

Pour le déploiement des architectures de Deep Learning, une machine suffisamment puissante doit être paramétrée pour assurer un apprentissage efficace des architectures profondes. Dans le cadre de cette thèse, l'ordinateur utilisé disposait : de 64 *Gb* de mémoire vive, d'une carte graphique Nvidia Titan X et d'un processeur Intel® Core™ i7-5820K.

La puissance mémoire et la puissance GPU sont réquisitionnées principalement pour l'affichage des images satellites et des résultats, ainsi que pour l'apprentissage des architectures de Deep Learning. La puissance mémoire et la puissance CPU (de l'anglais *Central Processing Unit*) sont utilisées pour les prétraitements des images et la constitution des jeux de données.

Les logiciels *CUDA* et *CUDNN* sont obligatoires pour faire fonctionner le *framework Pytorch* et pour optimiser la mise en parallèle des calculs entre les nombreux coeurs de la carte graphique. À noter que le *framework Café* a été mis en place au départ avant d'être abandonné. *Python* est le langage de programmation servant à coder et à mettre en place les différentes architectures de Deep Learning. Fonctionnant sous *Pytorch*, il a permis la création de l'ensemble des jeux de données, ainsi que le code permettant l'apprentissage et le rendu des modèles de Deep Learning. L'ensemble des codes python ont été réalisés en utilisant l'interface *Jupyter Notebook*.

Des scripts *R* en coordination avec des algorithmes de *Qgis* ont assuré le prétraitement des données. Ces prétraitements prennent en compte les opérations de coupage, de « reprojection » et de « rasterization » de la donnée satellite. Le code *R* est utilisé conjointement avec l'interface *RStudio*. *Qgis* est aussi employé pour l'affichage des images satellites et des résultats. C'est sous ce dernier programme que la cartographie de la couverture et de l'usage du sol a été finalisée.

1.4 Contribution

Les principales contributions de cette thèse sont :

1. L'étude de plusieurs thématiques démontrant l'adaptabilité des techniques de Deep Learning au contexte d'une île tropicale et leur efficacité face à d'autres techniques de machine learning plus classiques ;
2. La comparaison de diverses architectures de Deep Learning pour la création d'une carte de la couverture et de l'usage des sols automatisée [61] ;
3. La constitution d'un jeu de données étiqueté manuellement de 5 zones de Nouvelle-Calédonie à partir d'images satellites SPOT6. Cette classification a été effectuée aussi bien pour la couverture terrestre que pour l'usage des sols. Ces données, ainsi que les images satellites, ont été rassemblées sous la forme d'un *benchmark* mis à la disposition de la communauté scientifique [62] ;
4. L'adaptation des données satellites de Sentinel-2 pour la détection de la couverture terrestre. Cette adaptation permet d'effectuer une carte d'occupation des sols de façon annuelle, ouvrant la possibilité à une étude sur le suivi de changement d'occupation du sol ;
5. L'utilisation de données virtuelles de télédétection pour pallier au manque de données pour la détection du palmier babaçu au Brésil. Ce travail expérimental prouve l'apport des modèles de simulation au monde de la télédétection et du machine learning.

En plus de ces contributions, un article portant sur la mise en point d'un algorithme de détection automatique de l'algue Trichodesmium sur des données du satellite MODIS a été rédigé pendant cette thèse [63].

1.5 Plan

Dans cette section, un bref résumé des différents chapitres rythmant ce manuscrit est présenté.

Chapitre 2 - Dans ce chapitre, une définition du terme Deep Learning est introduite pour présenter plus en détail le fonctionnement de ces architectures. Les principes fondamentaux des différentes méthodes de Deep Learning y sont expliqués et le détail mathématique du fonctionnement interne d'une architecture du type Convolutional Neural Network (CNN) est posé. Enfin, les architectures utilisées dans la suite de la thèse y sont illustrées et explicitées.

Chapitre 3 - Ce chapitre s'intéresse aux notions autour de la télédétection et de leurs applications à l'occupation du sol. Cette classification des sols s'est faite à partir de données satellites SPOT 6 (métrique) sur des données classifiées à la main. Plusieurs méthodes et plusieurs architectures ont été utilisées. Une grande variation des données en entrée des modèles a été appliquée pour étudier leur influence sur l'apprentissage.

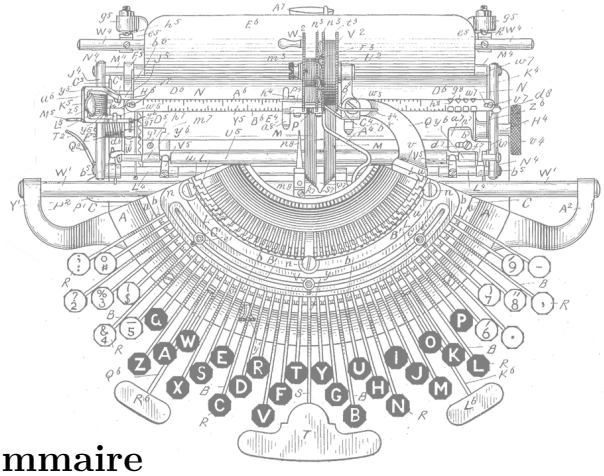
Chapitre 4 - Ce chapitre reprend les avancées faites dans le chapitre précédent pour une résolution spatiale moins fine (décamétrique), rendu possible grâce à l'utilisation de données satellitaires issues du capteur Sentinel-2 (programme européen COPERNICUS). Il permettra une étude temporelle des types de sols, qui ont été réadaptés de la classification manuelle de l'étude précédente.

Chapitre 5 - Ce chapitre présente une méthodologie basée sur la technique de détection d'objets. Le but de cette étude est de pouvoir détecter des palmiers au Brésil automatiquement à l'aide de techniques de Deep learning. Les données de terrain peu abondantes de ce projet sont complétées par des données simulées issues d'un modèle de création de données satellites artificiels. La répercussion et les gains de ces données simulées pour le Deep Learning y sont étudiés.

Chapitre 6 - Les perspectives découlant de cette thèse sont présentées dans ce chapitre.

CHAPITRE 2

LE DEEP LEARNING



"Sarah Connor ?"

Terminator

Sommaire

2.1 L'apprentissage d'un réseau de neurones 23
2.1.1 Les perceptrons, l'ancêtre des réseaux de neurones 24
2.1.2 Calcul de la fonction de coût 25
2.1.3 La descente stochastique du gradient 28
2.1.4 La rétro-propagation 29
2.2 Les réseaux de neurones convolutifs 33
2.2.1 La couche de convolution 34
2.2.2 La fonction ReLU 36
2.2.3 Le *Pooling* 37
2.2.4 La couche entièrement connectée 38
2.2.5 Le *Softmax* 38
2.3 Amélioration de l'apprentissage 39
2.3.1 Le *Drop-out* 39
2.3.2 La *Batch-normalization* 40
2.3.3 Optimisation d'une architecture de Deep Learning 41
2.4 Pixel classification 44
2.4.1 AlexNet 44
2.4.2 ResNet 46
2.4.3 DenseNet 49
2.5 Segmentation sémantique 52
2.5.1 FCN 52
2.5.2 Segnet 55
2.5.3 DeepLab 57

2.1 L'apprentissage d'un réseau de neurones

Le Deep Learning ou « apprentissage profond » en français, désigne des architectures de réseau de neurones artificiels qui s'inspirent de façon très schématique du fonctionnement du cerveau humain. Les notions entourant le fonctionnement interne des architectures en réseau de neurones sont abordées dans ce chapitre d'après plusieurs ouvrages de référence [64, 65, 66, 67].

L'architecture d'un réseau dense est composée de milliers de neurones organisés en une succession de couches (Figure 2.1). Chaque couche prend en entrée les sorties de tous les neurones de la couche précédente et effectue une opération non linéaire simple. Cette opération est accomplie pour chaque couche de l'architecture, et ce jusqu'à la sortie. Les données initiales sont stockées dans la première couche de l'architecture (*Input Layer*) ; la dernière couche effectuant la classification (*Ouput Layer*). Les couches situées entre ces deux dernières sont appelées couches cachées (*Hidden Layer*).

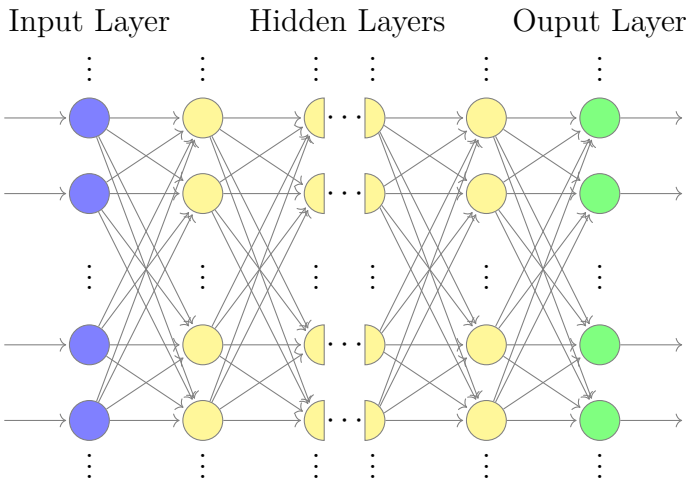


FIGURE 2.1 – Représentation schématique générale d'un réseau de neurones

Le résultat en sortie de l'architecture est alors comparé aux données réelles. La différence entre les données prédites par le modèle et les données observées est calculée par une fonction de coût. Les différents paramètres du réseau sont modifiés dans le but de minimiser cette fonction de coût. L'opération est itérée jusqu'à obtenir un taux de reconnaissance proche des performances humaines, il est dit de cette phase que le réseau de neurones est en train d'apprendre.

2.1.1 Les perceptrons, l'ancêtre des réseaux de neurones

Les réseaux de neurones utilisés en Deep Learning ont pour ancêtre les perceptrons [68], dont un schéma de fonctionnement est présenté ci-dessous (Figure 2.2).

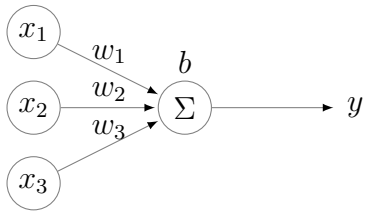


FIGURE 2.2 – Fonctionnement simple d'un perceptron

Soit x_i les entrées, w_i les poids associés à chacune des entrées, b le biais et y la sortie du modèle. La sortie est calculée par la fonction de transfert Σ de la manière suivante :

$$y = \begin{cases} 0 \text{ si } \sum_i w_i x_i + b \leq 0 \\ 1 \text{ si } \sum_i w_i x_i + b > 0 \end{cases} \tag{2.1}$$

La modification des variables de poids et de biais permet d'influer directement sur la sortie du modèle et donc d'ajuster son comportement pour obtenir une première forme de décideur. Cependant, à cause des entrées et des sorties binaires, la moindre variation d'une variable change entièrement le comportement global du décideur. La Figure 2.3 illustre ce problème : pour des valeurs d'entrées x_i égales à 1, le résultat est proche de zéro, donc proche de la valeur de basculement. Si le biais ou l'un des poids est modifié, la réponse de ce modèle pourrait être différente.

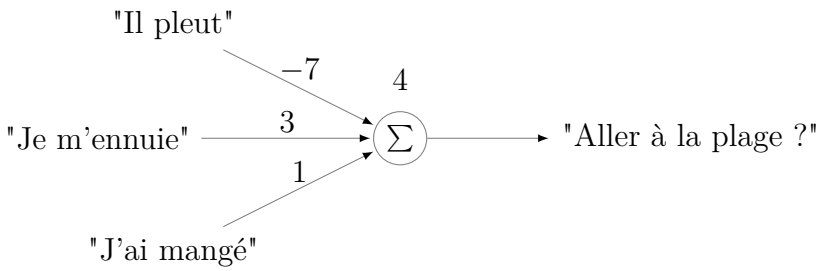


FIGURE 2.3 – Exemple simple de l'utilisation d'un perceptron

Pour un seul perceptron, ce changement est contrôlable, mais pour une architecture plus complexe telle que la Figure 2.1, le nombre de possibilités fait que le modèle n'est plus maîtrisable. Par la suite, la binarité de l'Équation 2.1 sera remplacée avec

une sortie continue comprise entre 0 et 1. La fonction de transfert n'est plus une simple fonction de seuillage, les perceptrons sont devenus des neurones sigmoïdes du nom de la nouvelle fonction de transfert.

$$\text{soit z } = \sum_i w_i x_i + b \tag{2.2}$$

La fonction sigmoïde s'écrit :

$$\sigma(z) \equiv \frac{1}{1 + e^{-z}} \tag{2.3}$$

Il existe d'autres fonctions sigmoïdes pouvant être utilisées (Figure 2.4), plus ou moins efficaces, pour calibrer un réseau de neurones. Les différentes fonctions existantes et le choix de l'une d'entre elles sont discutés plus loin dans ce chapitre.

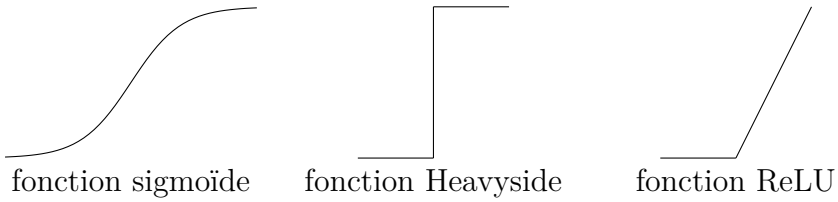


FIGURE 2.4 – Exemple de fonctions de transfert utilisables dans un réseau de neurones.

2.1.2 Calcul de la fonction de coût

À partir d'un jeu de données de chiffres manuscrits de 8 pixels de côté dont un extrait est présenté Figure 2.5, des vecteurs x contenant les 64 pixels composant l'image sont insérés à l'entrée d'un réseau de neurones permettant la reconnaissance de chiffres manuscrits, Figure 2.6.

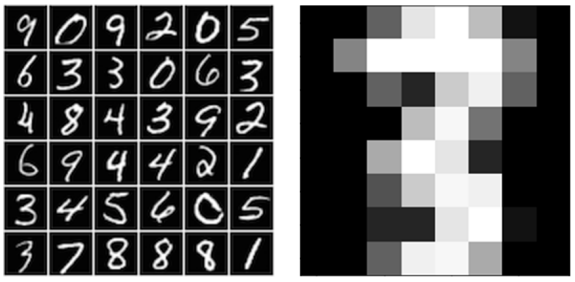


FIGURE 2.5 – Extrait d'un jeu de données de chiffres manuscrits qui doivent être reconnus par un réseau de neurones. À droite, le chiffre 3 de 8 pixels de côté.

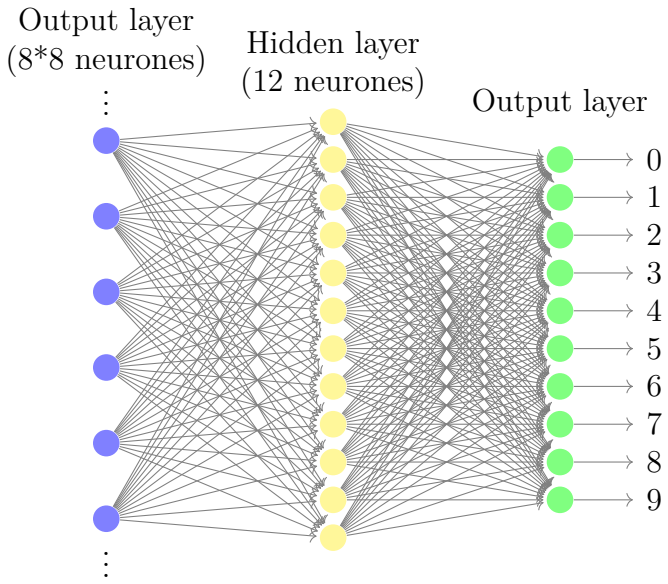


FIGURE 2.6 – Architecture en réseau de neurones reconnaissant les chiffres manuscrits

La fonction de transfert utilisée pour ce modèle est la fonction sigmoïde $\sigma(z)$ définie plus tôt (Équation 2.3). Pour déterminer la précision du modèle, une fonction de coût qui doit être minimisée est définie. Il existe de nombreuses fonctions de coût, la fonction quadratique étant l'une des plus utilisées. Il est aussi possible d'utiliser des fonctions de vraisemblance, qu'il faut maximiser.

Soit n, le nombre d'échantillons passé dans l'architecture, x l'un de ces échantillons, b et w respectivement les biais et les poids internes de l'architecture, $y(x)$ la valeur en sortie du modèle et a la valeur réelle observée. La fonction de coût C s'écrit alors :

$$C(w,b) \equiv \frac{1}{2n} \sum_{x} \|y(x) - a\|^2 \tag{2.4}$$

Plus $y(x)$ est proche de a, plus la fonction de coût sera petite. Le but étant de trouver des valeurs de biais (b) et de poids (w) tels que $C(w,b)$ tend vers 0.

Soit ΔC les changements que subit la fonction de coût. Pour rappel, la motivation dans le choix d'une fonction de transfert tel que la fonction sigmoïde est : qu'un léger changement dans le biais ou le poids d'un neurone change aussi légèrement la valeur de sortie de l'ensemble du réseau. Ces changements peuvent s'écrire comme étant la somme des changements fait aux variables composant chaque neurone.

Soit v la combinaison des variables w et b liées à un neurone et n le nombre total de neurones d'une architecture. Il est alors possible d'écrire ΔC de la manière suivante :

$$\Delta C \approx \frac{\partial C}{\partial v_1}\delta v_1 + \frac{\partial C}{\partial v_2}\delta v_2 + ... + \frac{\partial C}{\partial v_n}\delta v_n \tag{2.5}$$

Or le gradient de la fonction C s'écrit :

$$\nabla C \equiv (\frac{\partial C}{\partial v_1}, \frac{\partial C}{\partial v_2}, ..., \frac{\partial C}{\partial v_n})^T \tag{2.6}$$

Ce qui permet de simplifier l'équation précédente :

$$\Delta C \approx \nabla C.\Delta v \tag{2.7}$$

ΔC représente donc les changements de la fonction de coût, Δv les changements des variables liées à chaque neurone et enfin ∇C, le gradient faisant le lien entre les variations des variables v et les variations en conséquence de C. Le but étant de minimiser la fonction de coût ($\Delta C < 0$). Pour cela, soit η un paramètre de faible valeur positive appelé aussi **pas d'apprentissage**, tel que :

$$v \to v' = v - \eta \nabla C \tag{2.8}$$

Qui s'écrit aussi de la manière suivante :

$$\Delta v = -\eta \nabla C \tag{2.9}$$

L'équation 2.7 s'écrit alors :

$$\Delta C \approx -\eta \|\nabla C\|^2 \tag{2.10}$$

Ici, il faut paramétrer une valeur du pas d'apprentissage assez petite pour obtenir une approximation du minimum satisfaisante et assez grande pour que l'algorithme converge rapidement. Pour cela, soit $\|\Delta v\| = \varepsilon$ avec $\varepsilon > 0$ représentant les petites variations des variables, le pas d'apprentissage s'écrit alors :

$$\eta = \varepsilon / \|\nabla C\| \tag{2.11}$$

Soit C_x le coût moyen d'une seule entrée x tel que :

$$C_x \equiv \frac{1}{2}\|y(x) - a\|^2 \tag{2.12}$$

Et sachant qu'en pratique, pour obtenir le gradient ∇C, il faut calculer les gradients ∇C_x séparément pour chaque entrée d'entraînement x, puis en calculer la moyenne :

$$\nabla C = \frac{1}{n}\sum_x \nabla C_x \tag{2.13}$$

Le temps de calcul se retrouvera d'autant prolongé que le nombre d'entrées sera augmenté. Cette augmentation du temps de calcul n'étant pas négligeable, une technique appelée "descente stochastique du gradient" est utilisée pour améliorer la vitesse d'apprentissage.

2.1.3 La descente stochastique du gradient

Cette technique consiste à extraire un échantillon des données d'entrées et d'en calculer le gradient. L'idée est de prendre un petit nombre m de données d'entraînement. Plus m est important et plus on se rapprochera de la valeur de gradient ∇C. Soit $X_1, X_2, ..., X_m$ ces variables d'entrées, tel que :

$$\frac{1}{m}\sum_{j=1}^{m} \nabla C_{X_j} \approx \frac{1}{n}\sum_x \nabla C_x = \nabla C \tag{2.14}$$

Soit les variables d'origines de poids (w) et de biais (b), tel que, comme pour l'Équation 2.8 :

$$w_k \to w'_k = w_k - \eta\frac{\partial C}{\partial w_k} \tag{2.15}$$

$$b_l \to b'_l = b_l - \eta\frac{\partial C}{\partial b_l} \tag{2.16}$$

Dès lors, en prenant en compte l'équation 2.14, les deux équations précédentes deviennent :

$$w_k \to w'_k = w_k - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial w_k} \tag{2.17}$$

$$b_l \to b'_l = b_l - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial b_l} \tag{2.18}$$

Cette technique, opérée plusieurs fois, permet de réduire considérablement le jeu de données d'entrée en donnant une estimation du gradient, augmentant ainsi la vitesse d'apprentissage.

2.1.4 La rétro-propagation

Par la suite, une fois le résultat de la fonction de coût connu, il faut modifier les différentes variables de l'architecture afin de réduire ce coût. Il faut donc savoir de quelle manière les variables de poids (w) et de biais (b) influencent la fonction de coût.

Soit un réseau de neurones constitué de plusieurs couches. Soit l un entier désignant le numéro d'une de ces couches, j un entier désignant le neurone numéro j de la couche l et enfin k, l'entier désignant le neurone numéro k de la couche $l-1$ lié au neurone j. Car pour rappel, un neurone dispose d'un seul biais, mais de plusieurs poids. Alors l'équation permettant de connaître l'influence des variables sur la fonction de coût s'apparente à savoir les valeurs des dérivées partielles de $\partial C/\partial b^l_j$ et de $\partial C/\partial w^l_{jk}$.

Soit z, défini dans l'équation 2.2 tel que $z = \sum_i w_i x_i + b$. Pour ce calcul, une valeur intermédiaire supplémentaire δ^l_j est définie comme étant l'erreur du neurone j dans la couche l. Cette variable à une influence sur la fonction de coût : $\partial C/\partial z^l_j$, et en considérant que cette influence est proche de zéro, l'heuristique suivante est obtenue :

$$\delta^l_j \equiv \frac{\partial C}{\partial z^l_j} \tag{2.19}$$

Soit a_j^l, l'activation du neurone numéro j dans la couche l. l'activation d'un neurone est lié directement à la couche précédente tel que :

$$a_j^l = \sigma(\sum_k w_{jk}^l a_k^{l-1} + b_j^l) \tag{2.20}$$

En reprenant la notation de l'équation 2.2 :

$$z_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l \tag{2.21}$$

Alors l'équation 2.20 devient :

$$a_j^l = \sigma(z_j^l) \tag{2.22}$$

Soit L le nombre de couche du réseau de neurones. À partir des expressions précédente, les équations suivantes sont obtenues :

Équation de l'erreur de la couche de sortie :

$$\delta_j^L = \frac{\partial C}{\partial a_j^L}.\sigma'(z_j^L) \tag{2.23}$$

Le premier terme $\partial C/\partial a_j^L$ mesure la vitesse à laquelle le coût C change en fonction de la sortie du neurone j. Le second terme $\sigma'(z_j^L)$ mesure la vitesse à laquelle la fonction d'activation σ change en fonction de z_j^L. En reprenant la fonction quadratique pour la fonction de coût, il est aisé de calculer le premier terme de l'équation :

$$\partial C/\partial a_j^L = (a_j^L - y_j) \tag{2.24}$$

L'équation 2.23 s'exprime alors sous forme matricielle :

$$\delta^L = (a^L - y) \odot \sigma'(z^L) \tag{2.25}$$

Équation de l'erreur dépendant de l'erreur de la couche suivante :

$$\delta^L = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) \tag{2.26}$$

Ici il est possible de penser cette équation comme un moyen de déplacer l'erreur sur la couche précédente.

Équation calculant le taux de changement du coût en fonction de n'importe quel biais du réseau :

$$\delta_j^l = \frac{\partial C}{\partial b_l^j} \tag{2.27}$$

Équation calculant le taux de changement du coût en fonction de n'importe quel poids du réseau :

$$\delta_j^l a_k^{l-1} = \frac{\partial C}{\partial w_{jk}^l} \tag{2.28}$$

À partir des équations (2.23)-(2.28) et en considérant que le terme $\sigma(z_j^L)$ atteint une asymptote à proximité des valeurs 0 et 1 à cause de sa nature sigmoïde, tel que $\sigma'(z_j^L) \approx 0$. Cela a pour conséquence que l'apprentissage des poids de la couche final sera d'autant ralenti que la fonction d'activation approchera des valeurs 0 et 1. Dans ce cas, la couche de sortie est dite *saturée* et les poids ont arrêté d'apprendre, pareil pour les biais.

De la même manière avec l'équation (2.26), le terme $\sigma'(z^l)$ signifie que δ_j^l sera d'autant plus petit que l'architecture approchera de la saturation. Le choix d'une fonction d'activation qui évite ces saturations devient alors primordial, par exemple une fonction σ dont la dérivée σ' soit toujours positive et qui ne tend pas vers 0. L'algorithme de la rétro-propagation s'écrirait de la manière suivante :

1. Calcule de l'activation des neurones de la couche d'entrée a^1
2. Pour chaque couche l calcule de $z^l = w^l a^{l-1} + b^l$ avec $a^l = \sigma(z^l)$
3. Erreur de sortie δ^L : Calcul du vecteur $\delta^L = \nabla_a C \odot \sigma'(z^L)$
4. Pour chaque couche calcule de $\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$
5. Gradient de la fonction de coût calculé par $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l$ et $\frac{\partial C}{\partial b_j^l} = \delta_j^l$

Cependant, il est aussi souhaitable d'utiliser la rétro-propagation avec la technique de descente du gradient stochastique. Par exemple, soit un sous ensemble du jeu de données contenant m données d'entraînement, l'algorithme se ré-écrit de la manière suivante :

1. Pour chaque donnée d'entraînement x, calcule des entrées $a^{x,1}$ et applications des étapes suivantes :
   - Pour chaque couche l calcule de $z^{x,l} = w^l a^{x,l-1} + b^l$ avec $a^{x,l} = \sigma(z^{x,l})$
   - Erreur de sortie $\delta^{x,L}$: Calcul du vecteur $\delta^{x,L} = \nabla_a C_x \odot \sigma'(z^{x,L})$
   - Pour chaque couche, calcule de $\delta^{x,l} = ((w^{w,l+1})^T \delta^{l+1}) \odot \sigma'(z^{x,l})$
2. Pour chaque couche, mise à jour des poids et des biais en accord avec les règles :
   - $w^l \to w^l - \dfrac{\eta}{m} \sum_x \delta^{x,l} (a^{x,l-1})^T$
   - $b^l \to b^l - \dfrac{\eta}{m} \sum_x \delta^{x,l}$

2.2 Les réseaux de neurones convolutifs

De nombreuses architectures de Deep Learning existent, le choix de l'une d'entre elles dépendant grandement du type de problématique étudié. Dans le cas de la détection d'objets dans une image ou de la séparation d'un espace en plusieurs classes, ce sont les réseaux de neurones convolutifs (CNN, pour *Convolutional Neural Network*), s'inspirant du cortex visuel des animaux, qui seront privilégiés.

Un CNN est composé majoritairement de couches de convolution avec une fonction de transfert du type ReLU. Elles permettent d'extraire et de traiter de petites zones de l'image qui seront mis en relation dans des couches dites de « pooling ». Cet enchaînement sert d'encodeur dans l'architecture, c'est à dire qu'il permet, à l'aide de l'application de filtres successifs et de sous-échantillonnages de la donnée, d'extraire les informations essentielles de l'image. L'ensemble de ces informations est alors mis en commun dans une dernière étape du réseau de neurones, la classification. Cette étape se fait grâce à un couche « entièrement connectée » (*fully connected* en anglais). Une couche « softmax » est souvent ajouté pour une meilleur visibilité des résultats. En exemple, l'architecture LeNet-5 [69] présentée dans la Figure 2.7 qui utilise l'ensemble des couches nommées précédemment.

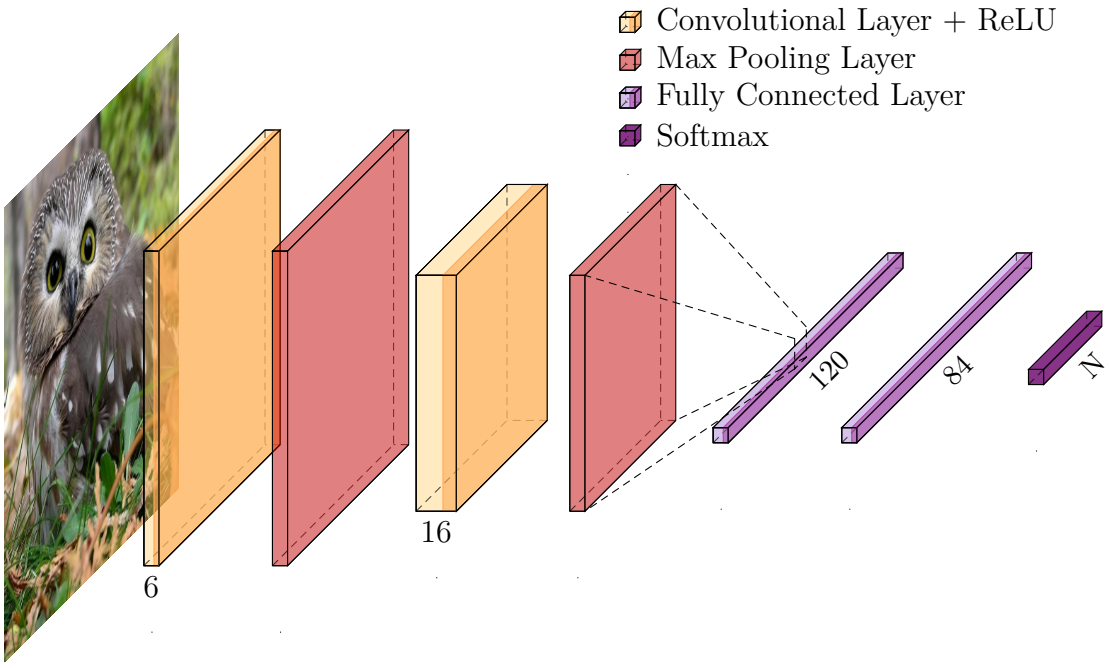


FIGURE 2.7 – Illustration d'un réseau neuronal convolutif, LeNet-5 [69]. Les chiffres indiquent le nombre de filtres des couches de convolution et le nombre de neurones des couches entièrement connectées. N représente le nombre de classes en fin de classification.

2.2.1 La couche de convolution

Une couche de convolution ("convolution layer" en anglais) peut être comparée à l'application d'un noyau de convolution sur l'ensemble d'une image. Ces noyaux de convolution sont comparables à des filtres qui permettraient d'extraire des caractéristiques précises d'une image. Par exemple, dans la Figure 2.8, les contours de l'image sont mis en valeur par l'application d'un filtre simple.

Au lieu de considérer chaque pixel individuel de l'image comme pour les perceptrons, un noyau de convolution, qui peut être de taille variable, permet de prendre en compte le voisinage de chaque pixel. C'est le même noyau qui est appliqué à l'ensemble de l'image, réduisant significativement le nombre de paramètres et donc l'empreinte mémoire. Cette méthode permet aussi une invariance du traitement induit par les translations.

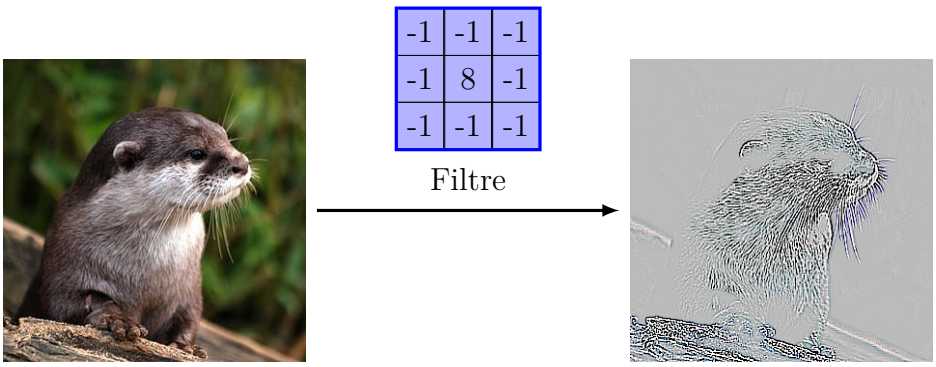


FIGURE 2.8 – Détection du contour à l'intérieur d'une image à l'aide d'un filtre créé à partir d'une matrice convolutionnelle.

Plusieurs paramètres sont à prendre en compte dans la mise en place d'une couche de convolution. La taille des filtres, le nombre de filtres, le pas de décalage du filtre (*stride* en anglais) et la zone de 0 à remplir autour de l'image (*padding* en anglais).

La taille du filtre La première couche de convolution extrait des caractéristiques de bas niveau, telles que les contours, les ensembles unis de l'image ou encore les couleurs. Ce filtre doit être configuré pour correspondre au mieux à la granularité de l'image. La taille des filtres, ou plus précisément la taille de la matrice qui sera appliquée varie énormément selon la littérature, dépendant principalement de la problématique étudiée et des données à disposition. Le choix de ce paramètre

relève d'une connaissance empirique de la problématique et du fonctionnement de l'architecture mise en place. La Figure 2.9 montre l'application d'un filtre de taille 3×3.

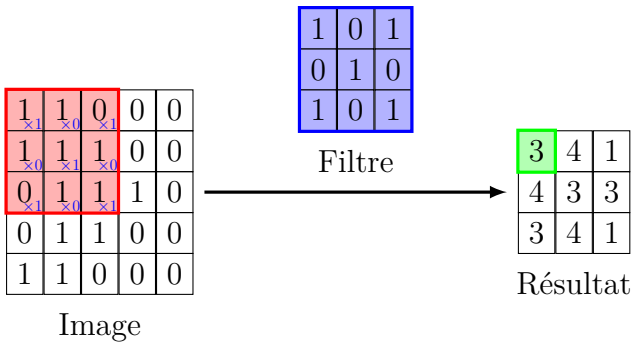


FIGURE 2.9 – Exemple du fonctionnement d'une couche de convolution

Le nombre de filtres Le nombre de filtres dépend directement des capacités de calcul à disposition : plus l'architecture sera profonde plus le temps de calcul sera important. Cela impacte aussi le nombre d'images qu'il est possible de faire passer en même temps dans l'architecture, appelé « Batch-size ». Si la problématique est complexe, par exemple avec de nombreuses classes à prédire, alors le passage d'une large quantité de données est nécessaire pour une bonne représentation du jeu de données. Cependant, les filtres doivent être en nombre suffisant pour répondre à la complexité de la tâche. Le nombre de filtres offre autant de degrés de liberté dans l'interprétation de l'image par l'architecture, un nombre insuffisant impacteraient directement les performances du modèle. En fonction de la mémoire disponible, il faut donc trouver un équilibre entre la taille du *Batch-size* et le nombre de filtres de l'architecture.

Le nombre de filtres varie tout au long de l'architecture. Dans le cas général, la taille de l'image diminue au fur et à mesure de son avancé dans le modèle, il est alors possible d'augmenter en conséquence le nombre de filtres plus l'image progresse dans l'architecture. Cette augmentation progressive permet de conserver les informations extraites dans les couches précédentes tout en ajoutant des degrés de liberté supplémentaire à la couche suivante pour extraire de nouvelles informations. Le nombre de filtres d'une couche à l'autre ne doit pas diminuer pour ne pas perdre d'informations. Tout comme pour la configuration de la taille des filtres, le choix du nombre de filtres à chaque étape de l'architecture relève d'une connaissance empirique de la problématique et de l'architecture de Deep Learning.

Le *stride* et le *padding* Le « stride » désigne le nombre de pixels dont se décalera la fenêtre coulissante du filtre sur l'image. Entre le *stride* et la taille du filtre, il est nécessaire de faire attention à la taille de sortie de l'image après la convolution. Dans le cas d'un filtre 3×3 avec un *stride* de 1 (déplacement du filtre pixel par pixel), une image de 5×5 pixels se transformera en image de 3×3 pixels comme dans la Figure 2.9. Si la diminution de l'image est un comportement non désiré dans l'architecture, un paramètre supplémentaire doit être configuré, le « padding ».

Avec un padding fixé à 1, des valeurs nulles sur une étendue de 1 pixel sont ajoutées tout autour de l'image (champ jaune dans la Figure 2.10). Ce qui résulte à une image aux mêmes dimensions que l'image d'entrée, comme illustré dans la Figure 2.10.

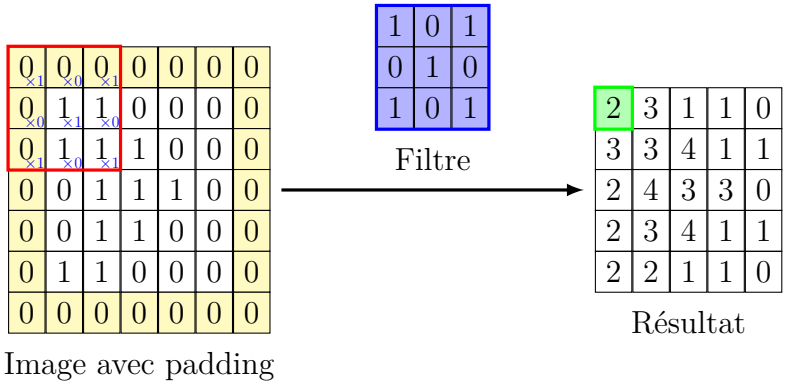


FIGURE 2.10 – Exemple de conservation des dimensions d'une image avec un *stride* et un *padding* de 1.

2.2.2 La fonction ReLU

Dans la section 2.1.1, le rôle et le fonctionnement d'une fonction d'activation a été abordé. Il existe d'autres fonctions non linéaires qui peuvent être utilisées à la place de la fonction sigmoïde $\sigma(x) = (1 + e^{-x})^{-1}$, telle que la tangente hyperbolique saturante $f(x) = tanh(x)$ ou la fonction ReLU (Unité Linéaire Rectifiée). La plupart des architectures actuelles utilisent la fonction ReLU du fait de sa rapidité et de ses performances. En effet, la fonction ReLU est une opération non linéaire très simple :

$$f(x) = max(0, x) \tag{2.29}$$

Cette fonction remplace les valeurs négatives d'une sortie de couche de l'architecture par la valeur 0, augmentant ainsi la non linéarité des calculs tout en n'altérant

pas le champ de décision, la partie positive. La Figure 2.11 illustre l'application de cette fonction d'activation.

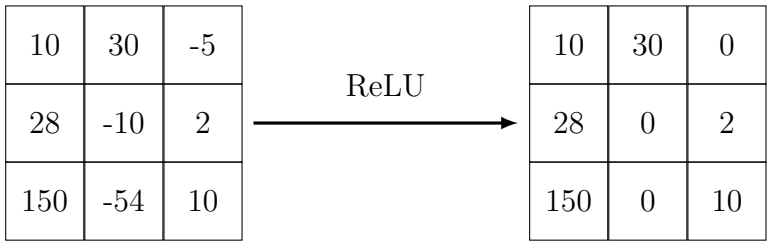


FIGURE 2.11 – Exemple du fonctionnement de la fonction de transfert ReLU

2.2.3 Le *Pooling*

Très souvent mis après une couche de convolution dans un CNN, la couche dite de « Pooling » est responsable du sous-échantillonage de l'image. Tout en conservant les caractéristiques les plus importantes et leur emplacement spatial dans l'image, une couche de pooling réduit les dimensions des données en combinant les sorties de groupes de neurones en une seule entrée pour la couche suivante. Au fur et à mesure que l'image avance dans le modèle, cette dernière est réduite par les couches de pooling successives, réduisant le nombre de paramètres et donc l'empreinte mémoire.

Il existe plusieurs fonctions non linéaires pour mettre en œuvre le pooling : Le *Max Pooling*, qui renvoie la valeur maximale de la portion de l'image couverte par le noyau ; l'*Average Pooling*, qui renvoie la moyenne ; et le *Sum Pooling* qui renvoie à la somme des valeurs. Le *Max Pooling* est le plus souvent utilisé, car en plus d'extraire les caractéristiques principales de l'image tout en réduisant ces dimensions spatiales, permet de réduire le bruit de l'image.

La taille du noyau et le *stride* sont des paramètres à configurer pour chaque couche de *Pooling*. Dans la majorité des cas, c'est un *Max Pooling* avec un noyau 2×2 qui est appliqué avec un *stride* de 2 comme dans l'exemple de la Figure 2.12.

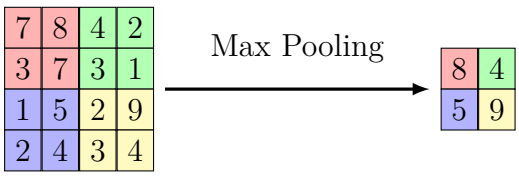


FIGURE 2.12 – Exemple du fonctionnement d'une couche de *Max Pooling* avec un noyau de 2×2 et un *stride* de 2.

2.2.4 La couche entièrement connectée

Après une succession de couches de convolution et de couches de Pooling, l'architecture CNN a réussi à extraire des informations de haut niveau de l'image. La mise en relation et le « raisonnement » final de l'architecture se fait par le biais de couches dites entièrement connectées. Les couches entièrement connectées désignent une couche reliant chaque neurone d'une couche à une autre couche. D'une matrice, nous passons à un vecteur d'informations de haut niveau, facilitant la classification finale. La Figure 2.6 est une architecture constituée uniquement de couches entièrement connectées par exemple.

D'autres méthodes de machine learning peuvent remplacer cette classification finale, comme SVM (Support Vector Machine) [70], Random Forest [71] ou encore XGBoost [72].

2.2.5 Le *Softmax*

Les valeurs de la couche de sortie sont des valeurs propres à l'architecture et sont donc difficiles à interpréter. La régression logistique qu'est la fonction « Softmax », est utilisée juste après la classification finale pour ramener les valeurs de sortie dans la plage $[0, 1]$. De plus la somme des sorties doit être égale à la valeur 1, c'est donc un vecteur de probabilités qui est obtenu à la sortie.

Soit y le vecteur en sortie de taille n tel que $y = (y_1, y_2, ..., y_n)$, la fonction softmax σ s'écrit :

$$\sigma(y)_i = \frac{e^{y_i}}{\sum_{k=1}^{n} e^{y_k}}, \forall i \in \{1, ..., n\} \tag{2.30}$$

Les architectures du type CNN ont la capacité de reconnaître et localiser précisément des objets dans une image. C'est un apprentissage sans supervision, l'architecture met au point ses propres filtres pour discriminer la donnée et capturer l'information essentielle de l'image. Par l'utilisation de couches de convolution, de pooling et de fonction simple comme le ReLU, les CNN parviennent à réduire le nombre de paramètres nécessaires sans perdre les caractéristiques principales de l'image, primordiales pour obtenir une bonne prédiction. C'est une méthode automatique qui permet de traiter un volume massif de données.

2.3 Amélioration de l'apprentissage

Il existe plusieurs façons d'améliorer un réseau de neurones, que ce soit dans le choix d'une meilleure fonction de transfert, l'implémentation de méthodes de régularisation telles que le « Drop-out » ou la « Batch-normalization », ou encore l'application de méthodes heuristiques pour choisir de meilleurs paramètres et mieux initialiser les poids du réseau.

Un problème connu des techniques de machine learning, réduisant les performances d'apprentissage de certains modèles, est le phénomène de *sur-apprentissage*. Ce phénomène apparaît en général en fin du processus d'apprentissage, lorsque le modèle tente de simuler au plus près les données en apprenant le bruit associé au signal. Des techniques de régularisation, telles que le *Drop-out* permettent de limiter ce phénomène.

2.3.1 Le *Drop-out*

L'hypothèse entourant le *Drop-out* est que, lors de l'apprentissage d'un réseau, certains neurones développent une co-dépendance [73]. Pour qu'une architecture fonctionne correctement, chaque neurone doit extraire les composantes essentielles d'une donnée, mais avec la co-dépendance certains neurones ne discriminent plus l'information réduisant les performances de l'architecture globale. Pour éviter ce biais, un nouveau réseau est créé de manière aléatoire en supprimant, une proportion (en général la moitié) des neurones des couches entièrement connectées (Figure 2.13). Les poids des neurones enlevés sont conservés avant d'être réintégrés au passage suivant.

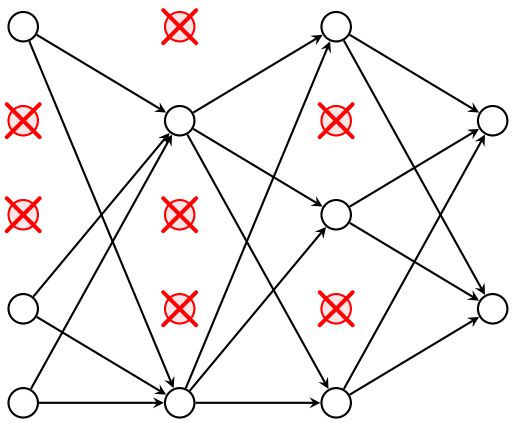


FIGURE 2.13 – Exemple du fonctionnement du *Drop-out*

En créant une multitude de réseau de neurones différents, et en pondérant chaque neurone en fonction de leur présence ou non dans la plupart des réseaux ainsi créés, une architecture globale complète est obtenue avec des neurones non co-dépendants. Le *Drop-out* en plus d'éviter la co-dépendance des neurones et donc de limiter le phénomène de sur-apprentissage [74], permet aussi de diminuer le nombre de paramètres dans les couches entièrement connectées et donc de réduire l'empreinte mémoire. Le modèle dans l'ensemble devient plus robuste, car les neurones apprennent des caractéristiques plus « générales » lors de la classification finale.

2.3.2 La *Batch-normalization*

Durant la phase de propagation de la donnée dans le réseau, celle-ci est altérée à chaque passage dans une couche et donc, les paramètres des couches précédentes influent directement sur les paramètres des couches suivantes lors de la phase de rétropropagation. Dans le cas d'un réseau de neurones profond, le moindre bruit se retrouve amplifié d'autant plus qu'il y a de couches dans le réseau, apprenant à l'architecture des comportements indésirables et entraînant donc par la même occasion un sur-apprentissage.

La *Batch-normalization* [75] ou la « normalisation des lots » en français, est une méthode de régularisation utilisée en Deep Learning pour limiter le décalage interne des covariables et ainsi réduire les phénomènes de sur-apprentissage. La *Batch-normalization* effectue pour cela une normalisation des données.

Soit x la valeur contenue dans le lot B tel que $B = \{x_1, ..., x_m\}$, où m est le nombre d'éléments contenus dans le lot B. Soit μ_B la moyenne du lot. À la sortie de la fonction de transfert, les données sont normalisées selon l'équation suivante :

$$\mu_B = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{2.31}$$

Soit σ_B^2 l'écart type du lot tel que :

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_B)^2 \tag{2.32}$$

Alors la normalisation $\hat{x}_i$ de la valeur x_i s'écrit :

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{2.33}$$

ϵ étant une constante ajoutée à l'écart type du lot pour assurer une stabilité numérique.

Cette normalisation a cependant une influence sur les poids de la couche suivante. La descente du gradient stochastique peut annuler la normalisation, si c'est un moyen pour elle de minimiser la fonction de coût. Pour pallier à ce problème, la méthode de *Batch-normalization* ajoute deux paramètres entraînables à chaque couche permettant de redimensionner et décaler la donnée. Soit γ et β ces deux variables d'ajustement, la sortie y_i d'une variable x_i après l'application de la méthode de *Batch-normalization* s'écriera :

$$y_i = \gamma \hat{x}_i + \beta \tag{2.34}$$

La normalisation des lots peut être vue comme un prétraitement à chaque couche du réseau que ce dernier peut choisir de défaire à l'aide de ces deux variables introduites sans changer tous les poids de l'architecture.

L'utilisation du *Batch-normalisation* a pour but de réduire la co-dépendance qu'il existe entre chaque couche de l'architecture et d'éviter l'amplification de certains décalages internes de l'architecture. En évitant cette amplification, le réseau peut être entraîné avec un pas d'apprentissage plus élevé comme le gradient est mieux "contrôlé", et ainsi améliorer la vitesse de convergence de réseau [76].

L'utilisation conjointe des méthodes de *Batch-normalization* et de *Drop-out* permet alors la formation de modèle robuste et moins sujet au sur-apprentissage. Cependant l'utilisation de ces méthodes n'est pas anodine et ne se fait pas de manière systématique [77], car elles peuvent aussi entraîner des pertes de performances surtout lorsqu'elles sont utilisées ensemble [78, 79].

2.3.3 Optimisation d'une architecture de Deep Learning

Différentes méthodes peuvent être appliquées pour réduire le taux d'erreur, le temps d'apprentissage, les effets de sur-apprentissage ou encore pallier au manque de données.

Le pas d'apprentissage Un grand pas d'apprentissage permet la convergence rapide d'un modèle, mais peut induire une instabilité en fin d'apprentissage où le modèle doit effectuer des ajustements minimes. Le modèle peut osciller autour d'une valeur d'équilibre sans parvenir à se stabiliser. Une réduction du pas d'apprentissage

au fur et à mesure de l'apprentissage conjugue rapidité de convergence du modèle et un ajustement fin du modèle sur les dernières itérations de l'apprentissage. Fixé le pas d'apprentissage d'une architecture de Deep Learning reste un sujet complexe où plusieurs méthodes heuristiques existent [80, 81, 82]

Les fonctions de coût c'est en comparant la sortie du modèle aux données réelles et en calculant la différence entre les deux à l'aide d'une fonction de coût que le modèle parvient à déterminer dans quel sens progresser. Il existe de nombreuses fonctions de coût (l'erreur quadratique moyenne, la somme des carrés des résidus, l'erreur absolue moyenne, etc.) et le choix de l'une d'entre elles est complexe [83] et doit prendre en considération la problématique étudiée [84], au risque sinon de dégrader fortement les performances d'apprentissage. Par exemple, les objectifs d'une classification et d'une régression sont différents et la fonction de coût utilisé doit refléter cet objectif.

La régularisation L1 et L2 Lors du calcul de la fonction du coût il possible d'utiliser la méthode de dégradation des pondérations, en d'autres mots d'ajouter ce qu'on appellera une « pénalité ». Deux types de pénalité peuvent être ajoutés à la fin de la fonction de coût, la régularisation L1 et L2. La différence entre les deux tient dans la fonction de régression qu'utilise l'une et l'autre. Cette régularisation permettent de pénaliser les plus gros coefficients de l'architecture et qui peuvent éclipser les plus petits poids de l'architecture. En ajoutant cette pénalité, cela aide les plus petits paramètres de l'architecture à avoir de plus grande variation tout en minimisant celle des plus gros paramètres, permettant ainsi de limiter les problèmes de sur-apprentissage et ajoutant de la robustesse au modèle [85].

Augmentater des jeux de données Une des principales problématiques entourant le Deep Learning est la disponibilité de données en quantité suffisante pour entraîner une architecture en réseau de neurones. La donnée initiale est très souvent de qualité ou en quantité insuffisante et des techniques d'augmentation artificielle de la donnée ont été mis en point pour pallier ce problème. Par exemple, dans le cas de la reconnaissance dans une image, les images en entrée du modèle peuvent subir des rotations, des changements au niveau de leur contraste et de leur luminosité, ou encore des réajustements au niveau de leur colorimétrie. Ils existent beaucoup d'autres méthodes d'augmentation artificielle de la donnée [86, 87] qui permettent d'obtenir des modèles plus robustes et précis [88, 89, 90].

Il existe d'autres méthodes de régularisation et d'optimisation des architectures de Deep Learning [91, 92] qui n'ont pas été abordées, comme l'initialisation des poids de l'architecture ou l'utilisation d'autres méthodes que la descente stochastique du gradient. Les méthodes et techniques présentées ici sont les principales et les plus utilisées. Leur application est généralement suffisante pour obtenir un modèle robuste et avec des performances convenables.

2.4 Pixel classification

Les architectures de Deep Learning présentées dans cette section sont des CNN permettant de faire de la détection d'objets. C'est-à-dire que les sorties de ces architectures sont un label unique décrivant l'image d'entrée. Dans le cas de la télédétection, cela implique l'identification d'un objet ou d'une structure au sol de telle manière à ce que l'architecture puisse étiqueter l'image simplement, par exemple avec des qualificatifs comme « avion », « voiture », « piscine » ou encore « port ». Dans le cas d'une occupation du sol, c'est le pixel central de l'image qui sera étiqueté.

Dans cette section sont présentées les architectures AlexNet [20], ResNet [93] et DenseNet [94].

2.4.1 AlexNet

L'article présentant l'architecture AlexNet [20] est l'un des articles fondateurs les plus influents sur le sujet du Deep Learning. Cité plus de 57,000 fois au jour de cette thèse, AlexNet (Figure 2.14) s'est fait connaître en 2012 lors du « ImageNet Large Scale Visual Recognition Challenge » (ILSVRC) où elle parvient à achever 15.3% sur le « top-5 erreur » (le pourcentage d'images où la bonne classification n'est pas l'une des 5 classifications les plus probables), soit plus de 10.8 % moins que l'algorithme arrivé en second. Cette compétition permet en effet d'évaluer et de comparer des algorithmes de détection d'objets et de classification de données sur un très grand jeu de données : 15 millions d'images labellisées sur plus de 22,000 classes.

AlexNet est une variante de l'architecture LeNet du type Convolutional Neural Network (CNN) introduit par Yann Lecun[69]. L'architecture d'AlexNet est plus profonde avec ces 8 couches, les 5 premières couches étant des couches de convolution (Partie 2.2.1) dont les paramètres sont visibles Tableau 2.1, avec parfois une couche du type *MaxPooling* (Partie 2.2.3) (Figure 2.14). Les deux dernières couches de l'architecture sont deux couches entièrement connectées (Partie 2.2.4) avant la classification finale opérée avec un *Softmax* (Partie 2.2.5). Une fonction d'activation du type ReLu (Partie 2.1.1) est utilisée pour chaque couche. En plus de l'architecture, des techniques d'augmentation et de *Drop-out* (Partie 2.3.1) ont été utilisées pour limiter les effets de sur-apprentissage.

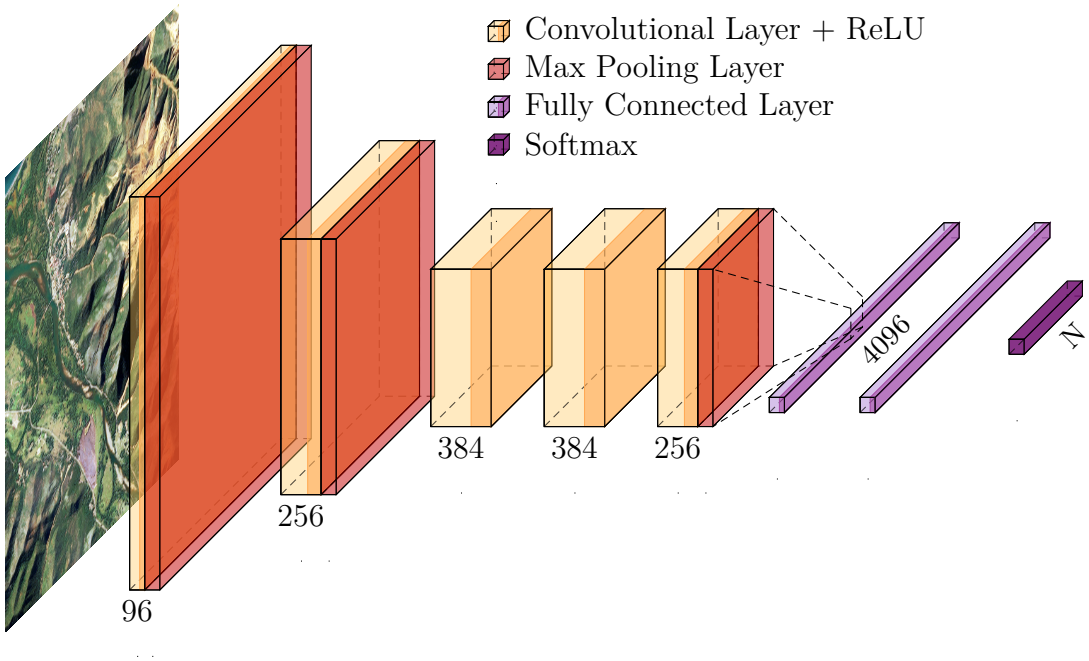


FIGURE 2.14 – L'architecture de Deep Learning AlexNet [20]. Les chiffres indiquent le nombre de filtres de chaque couche de convolution et le nombre de neurones présents dans les couches entièrement connectées. N représente le nombre de classes en fin de modèle. L'ensemble des paramètres sont résumés Tableau 2.1.

TABLEAU 2.1 – Configuration des couches de l'architecture AlexNet. Les paramètres des différentes couches sont explicités dans la Section 2.2.

Couche	**Taille du noyau**	**Nombre de filtres**	**stride**	**padding**
Convolution 1	11×11	96	4	0
Convolution 2	5×5	256	1	2
Convolution 3	3×3	384	1	1
Convolution 4	3×3	384	1	1
Convolution 5	3×3	256	1	1
Max Pooling	2×2	-	2	0

2.4.2 ResNet

Depuis le succès d'AlexNet, un nouveau mot d'ordre est né au sein de la communauté scientifique, les architectures de Deep Learning doivent être de plus en plus profondes. Cependant avec l'ajout de couches viennent de nouvelles contraintes, par exemple le problème de sur-apprentissage avec la multiplication des variables de l'architecture et le problème du « vanishing gradient ». Ce dernier exprime que, du fait des multiplications entre les différentes couches du gradient, lors de la *back propagation* (Partie 2.1.4), le gradient peut devenir infiniment petit. Ce faible gradient impacte le *momentum* de l'architecture, c'est-à-dire le changement des poids de ces couches, le réduisant significativement et faisant ainsi stagner l'apprentissage et pouvant même le dégrader.

ResNet pour Deep Residual Network [93] est une architecture de Deep Learning avec de nombreuses couches utilisant un procédé de « court-circuit ». Ces courts-circuits sont une fonction identité qui permet d'ignorer certaines couches de l'architecture. Ils permettent de lutter contre le problème du « vanishing gradient », car la simplification du réseau de neurones en utilisant moins de couches dans l'apprentissage initial minimise la propagation de celui-ci. De plus, les couches après un court-circuit réutilisent la fonction d'activation de la couche précédente, et ce jusqu'à ce qu'elles parviennent à apprendre et faire varier leurs poids. Les couches ignorées sont graduellement restaurées et intègrent les caractéristiques apprises de l'ensemble de l'architecture à laquelle elles appartiennent. ResNet permet d'entraîner des centaines de couches en limitant les problèmes de sur-apprentissage.

La Figure 2.15 représente une architecture ResNet 34-couches. Il existe plusieurs versions de ResNet avec plus de couches dont l'architecture interne est détaillée dans le Tableau 2.2. Il est à noter que l'architecture ResNet de 1001 couches a été créée et testée dans cette expérimentation [95]. Les courts-circuits pointillés dans la figure font référence aux identités qui ont été multipliées par une projection linéaire pour s'adapter aux nouvelles dimensions de la couche suivante. Le noyau de la première couche de convolution est de dimension 7×7, le reste des couches de convolution ont un noyau de dimension 3×3.

FIGURE 2.15 – L'architecture de Deep Learning ResNet-34 [93]. Les chiffres indiquent le nombre de filtres des couches de convolution et le nombre de neurones dans la couche entièrement connectée. N représente le nombre de classe en fin du modèle. Les flèches passant au dessus des couches représentent les « court-circuits » de l'architecture, celles en pointillées ont en plus une projection linéaire pour conserver les dimensions de l'architecture. L'ensemble des paramètres des différentes architectures ResNet sont présentés Tableau 2.2.

TABLEAU 2.2 – Table des paramètres des couches des différentes architectures de ResNet [93]

Couches	18-couches	34-couches	50-couches	101-couches	152-couches
Convolution 1	7×7 conv, 64, stride 2				
Pooling	3×3 max pool, stride 2				
Convolution 2x	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
Convolution 3x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
Convolution 4x	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
Convolution 5x	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
Pooling	global average pool				
Fully-connected	1000D fully-connected, softmax				

2.4.3 DenseNet

À partir du raisonnement autour de l'architecture ResNet, le principe des courts-circuits a été poussé plus loin dans l'architecture DenseNet [94]. Ici les courts-circuits ont été multipliés de telle manière que les ponts entre les couches se fassent avec toutes les couches suivantes, comme illustrés dans la Figure 2.16. De ce fait, l'entrée de toutes les couches de convolution est le résultat de toutes les couches de convolution précédentes. Pour des raisons d'empreinte mémoire, cette méthode ne peut pas être appliquée sur l'intégralité du réseau, c'est donc des modules entiers comme présentés dans la Figure 2.16 qui seront utilisés dans la nouvelle architecture et qui seront appelés « Dense Block ».

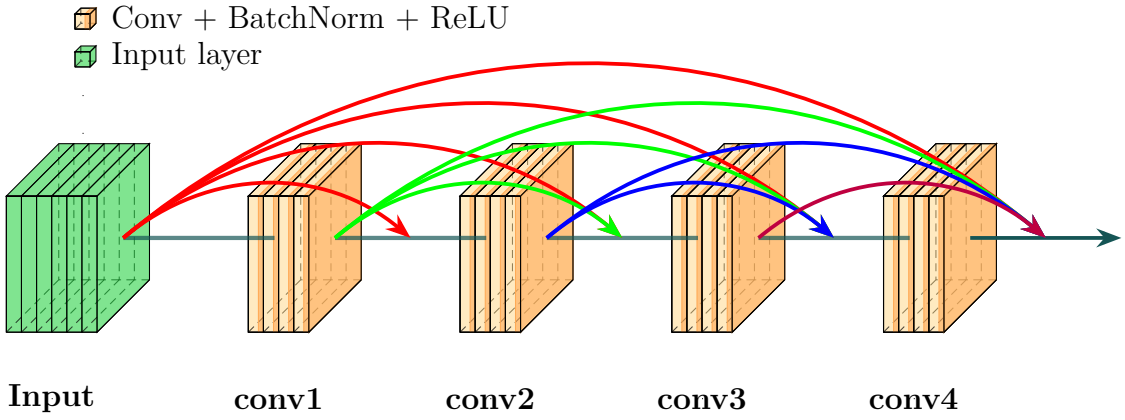


FIGURE 2.16 – Représentation d'un « Dense Block » de 5 couches où la première couche est distribuée à l'entrée des 4 groupes de couches suivants, la sortie du deuxième groupement de couches est distribuée aux 3 groupes suivants et ainsi de suite.

L'architecture se compose alors d'une succession de couches de convolutions, de couche *Max Pooling* et de *Dense Block*. L'architecture se conclut par une couche d'*Average Pooling*, avant une couche entièrement connectée qui finit par un *Softmax*. Il existe plusieurs versions de l'architecture DenseNet où les paramètres et le nombre des couches de convolutions changent dans les *Dense Block*. L'architecture représentée Figure 2.17 est l'architecture DenseNet-121 dont la composition interne est détaillée dans le Tableau 2.3.

DenseNet compile les différents avantages de ResNet, elle réduit les problèmes du « vanishing gradient » et avec ces courts-circuits diminuent significativement la complexité du modèle en réduisant le nombre de paramètres et donc l'empreinte mémoire. Cette technique encourage chaque couche à créer ses propres caractéristiques et évite ainsi les problèmes de sur-apprentissage.

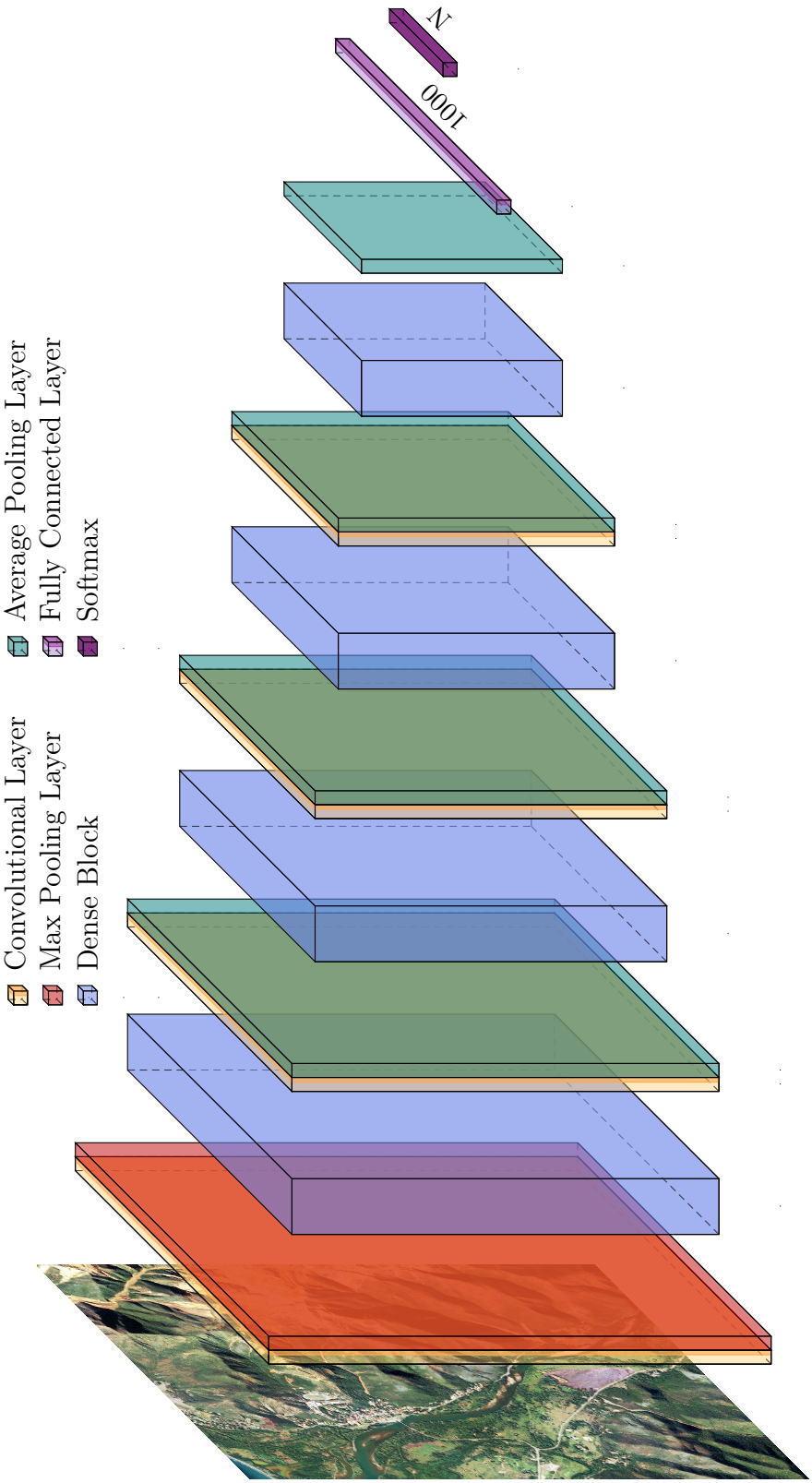


FIGURE 2.17 – L'architecture de Deep Learning DenseNet-121 [94]. Le chiffre correspond au nombre de neurones dans la couche entièrement connectée et N le nombre de classes en sortie de modèle. Les paramètres de l'architecture sont présentés Tableau 2.3

TABLEAU 2.3 – Table des paramètres des couches des différentes architectures de DenseNet [94]

Couches	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	7×7 conv, stride 2			
Pooling	3×3 max pool, stride 2			
Dense Block	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Convolution	1×1 conv			
Pooling	2×2 average pool, stride 2			
Dense Block	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Convolution	1×1 conv			
Pooling	2×2 average pool, stride 2			
Dense Block	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Convolution	1×1 conv			
Pooling	2×2 average pool, stride 2			
Dense Block	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Pooling	7×7 global average pool			
Fully-connected	1000D fully-connected, softmax			

2.5 Segmentation sémantique

La segmentation d'images consiste à rassembler entre eux, plusieurs pixels proches spatiallement dans une image et partageant des caractéristiques communes. Ces groupements de pixels partitionnent l'image en plusieurs segments qui mis en commun peuvent former des objets. L'étiquetage de ces objets relève d'une compréhension de haut niveau de l'image, car il faut réussir à mettre en relation les pixels composant un objet avec le contexte qui l'entoure pour déterminer la nature de l'objet. Ce partionnement suivi d'un étiquetage scrupuleux de chaque élément délimité est appelé « segmentation sémantique ». Ce type de problématique de détection est appliqué à de nombreux domaines du traitement de l'image et notamment à la télédétection.

Plusieurs architectures de Deep Learning existent pour réalisé cette tâche. Dans cette section ce sont les architectures de FCN [96], SegNet [97] et DeepLab [98] qui seront présentées. À partir d'une image d'entrée, le but de ces architectures est de produire une image de même dimension que celle d'entrée. Ces architectures peuvent être décomposées en deux parties, une partie encodeur et une partie décodeur. La partie encodeur, qui s'inspire très fortement des CNN, se charge comme son nom l'indique d'encoder l'information pour en extraire les caractéristiques principales. Cependant avec l'encodage de la donnée, une dégradation graduelle de l'image s'est réalisé. Le rôle du décodeur est de récupérer les caractéristiques principales produites par l'encodeur et de les reprojeter à la résolution initiale de l'image pour obtenir une classification dense. Plusieurs méthodes de ré-échantillonage de la donnée existent et sont détaillées par la suite.

2.5.1 FCN

Les architectures du type Fully Convolutional Networks (FCNs) [96] portent leur nom du fait qu'elles ne comportent que des couches de convolution dans leur structure. Figure 2.18, l'encodeur de FCN consiste en une succession de couches de convolution et le décodeur d'une couche de « déconvolution » permettant retrouver les dimensions initiales de l'image, suivie d'un softmax classifiant l'intégralité des pixels de l'image selon le nombre de classes voulu.

Si la couche de déconvolution de l'architecture FCN est réalisée uniquement avec les données de la couches précédentes en entrée, alors une perte d'informations importante est attendue du fait d'une perte de résolution de facteur 32. Une étape supplémentaire est réalisable dans la phase de décodeur de l'architecture comme

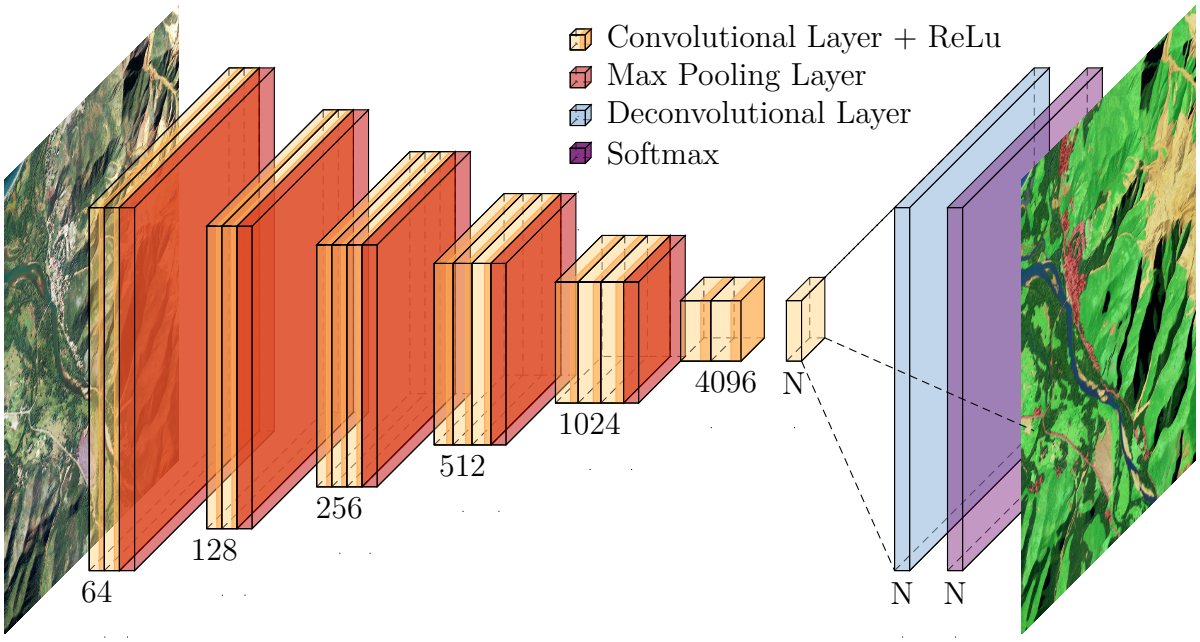


FIGURE 2.18 – L'architecture de Deep Learning FCN [96]. Les chiffres indiquent le nombres de filtres de chaque couche de convolution. N correspond au nombre de classes à prédire. Le détail du ré-échantillonage de la donnée est explicité Figure 2.20.

présentée Figure 2.20. Les données en sortie des couches de *Pooling* sont réintroduit à l'entrée de la couche de déconvolution, ajoutant ainsi une information spatiale pour la reconstitution de l'image finale. En fonction de la combinaison des différentes couches de *Pooling* entre elles, une série de variations de l'architecture FCN est obenue. Un exemple des différences existant entre les différentes architectures FCN est présenté Figure 2.19. Pour une image présentant de grands ensembles l'architecture FCN-32s sera adaptée. En revanche, pour la détection de petits objets aux limites clairement définies, du fait de l'importance de l'information spatiale de chaque objet, l'architecture FCN-8s doit être privilégiée.

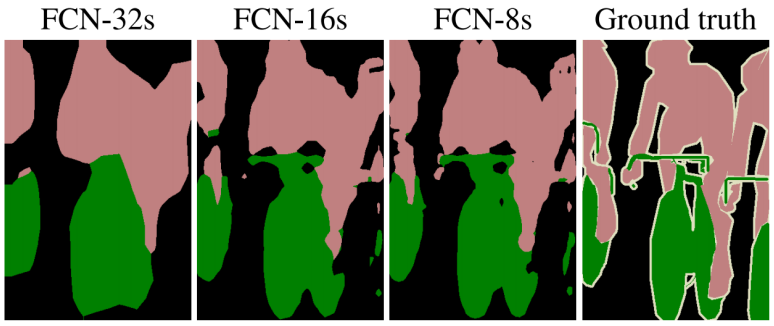


FIGURE 2.19 – Exemple des sorties des différentes architectures FCN issu du papier [96].

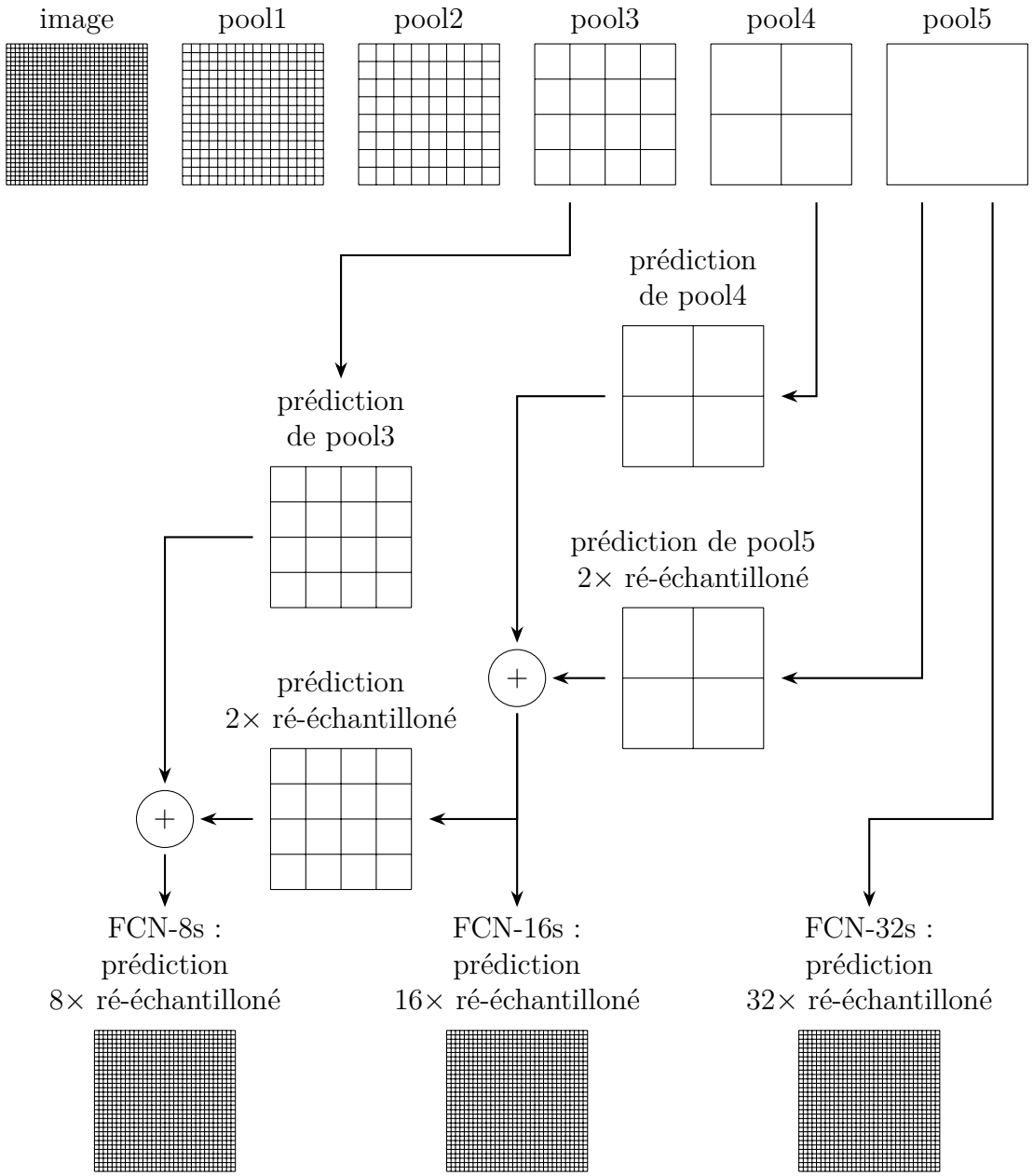


FIGURE 2.20 – Détail des variations d'entrée de la dernière couche de déconvolution de l'architecture FCN déclinant les variations possibles de cette dernière.

2.5.2 Segnet

SegNet [97] est un réseau de neurones du type encodeur-décodeur du même type que les architectures DeconvNet [99] et U-Net [100]. L'encodeur dans l'architecture est une série de couche de convolution et de couches de *Max Pooling*, comme pour l'architecture FCN [96]. À chaque *Pooling*, l'information de la couche est conservée avant d'être réintroduite à la couche de déconvolution correspondante dans la partie décodeur de l'architecture, liens représentés par des flèches Figure 2.21. De cette manière l'information spatiale perdue dans l'encodage et conservé à chaque étape du décodeur. Enfin La dernière couche de l'architecture, le *Softmax*, classifie l'ensemble des pixels de l'image d'entrée.

Cette architecture, utilisée initialement pour détecter en temps réel les différents éléments nécessaires à la conduite autonome, peut être adapté à l'observation par satellite. En effet, c'est à partir de cette architecture que l'architecture DeepNet-ForEO [101] s'inspire pour répondre à des problématiques de télédétection. Hormis des paramétrages internes différents, les deux architectures sont identiques. Dans les études suivantes c'est l'architecture SegNet qui a été testée.

FIGURE 2.21 – L'architecture de Deep Learning SegNet [97]. Les chiffres indiquent le nombre de filtres de chaque couche de convolution. Les flèches représentent les ponts effectués dans l'architecture pour restituer l'information en sortie de couche de Pooling à la couche de déconvolution équivalente.

2.5.3 DeepLab

L'architecture DeepLab [102] est passée par plusieurs versions avant d'arriver à celle qui est utilisée dans le cadre de cette thèse, l'architecture DeepLab V3+ [98]. La particularité de l'architecture DeepLab par rapport aux autres architectures est l'utilisation d'une couche de convolution dites « Atrous » [103] dans le décodeur de son architecture. Cette couche, illustrée Figure 2.22, permet d'extraire l'information essentielle d'une couche de convolution sans réduire sa résolution, remplaçant ainsi efficacement les couches de Déconvolution.

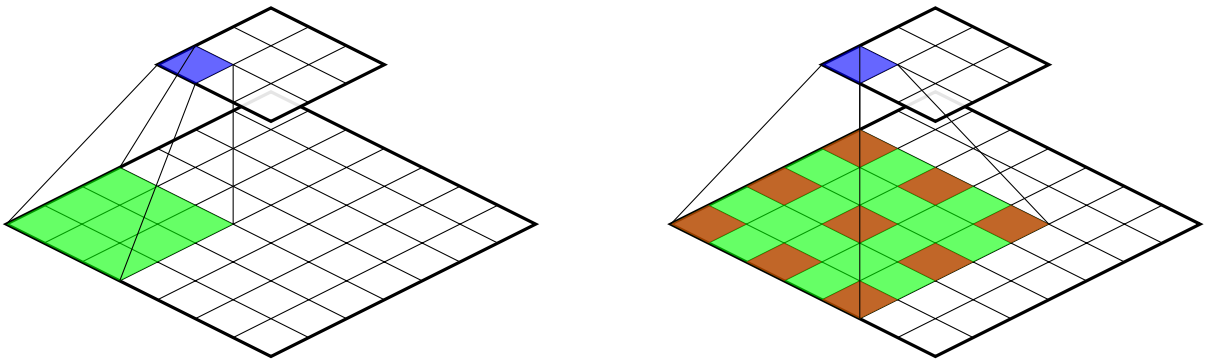


FIGURE 2.22 – Exemple de fonctionnement d'une couche Atrous avec pour paramètres à gauche r=1 et à droite r=6 pour l'équation 2.35.

Soit le filtre w, avec K le nombre de cellules de la matrice de convolution. Soit x la zone de l'image où est appliquée le noyau du filtre et y le résultat de cette opération. Soit r le taux de séparation entre chaque cellule, alors le résultat de l'application d'une couche « atrous convolution » s'écrit :

$$y_i = \sum_{k}^{K} x_{i+r.k} w_k, \forall k \in \{1, ..., K\} \tag{2.35}$$

Une architecture ResNet-101 [95] ou une architecture Xception [104], dont les dernières couches de *Pooling* ont été remplacées par une couche de convolutions *Atrous*, sont utilisées comme encodeur dans l'architecture DeepLab. À la sortie d'un de ces modèles, une pyramide de couche de convolution *Atrous* appelée Atrous Spatial Pyramid Pooling (ASPP), comme illustrée dans la Figure 2.23, est accolée. Les différents taux des couches de convolution *Atrous* de la pyramide permettent de prendre en compte des objets de plusieurs échelles et ainsi augmenter la précision du modèle.

Après un ré-échantillonnage de l'ensemble de l'image, une segmentation sémantique est obtenue. Les techniques employées dans cette architecture permettent d'élargir efficacement le champ de vision des filtres sans augmenter le nombre de paramètres et donc l'empreinte mémoire.

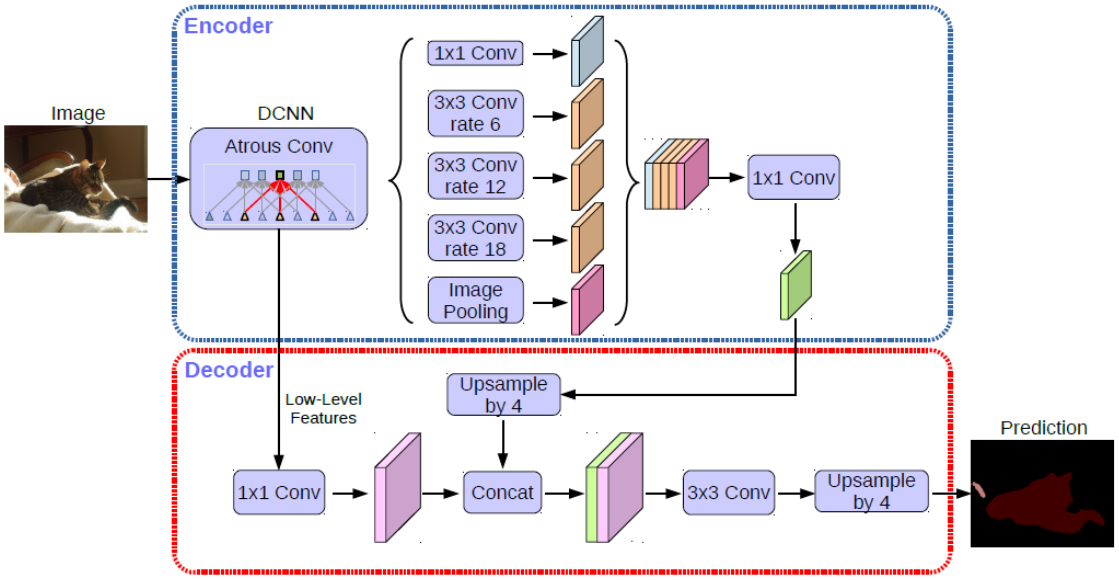
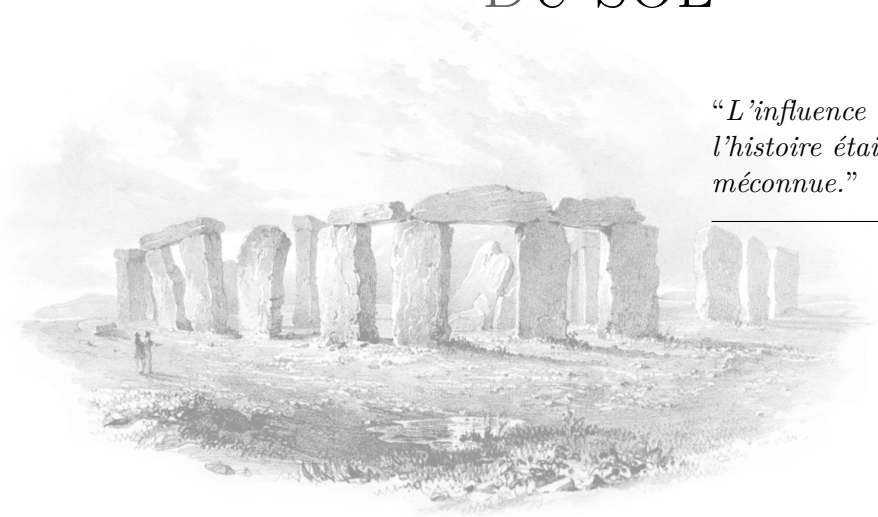


FIGURE 2.23 – L'architecture de DeepLearning de DeepLabV3+ issu du papier Chen et al. [98]

Les différentes architectures de Deep Learning présentées de manières succinctes dans ce chapitre ne sont pas exhaustives. De nombreuses autres architectures existent et sont suffisamment polyvalentes pour s'adapter à de nombreuses problématiques (Inception-4 [105], VGG19 [106], ENet [107], Recursive Convolutionnal Neural Network (R-CNN) [108], etc.), et même de générer de façon autonome ces propres jeux d'apprentissage (Réseaux Antagonistes Génératifs (GAN) [109]). Au jour de cette thèse, de nouvelles architectures performantes font leurs preuves chaque année, augmentant toujours plus le champ d'applications possibles des techniques de Deep Learning.

CHAPITRE 3

CLASSIFICATION DES OCCUPATIONS DU SOL



"L'influence de la géographie sur l'histoire était largement méconnue."

Frank Herbert

Sommaire

3.1 Introduction 61
3.1.1 Caractérisation de la zone d'étude 61
3.1.2 Préambule : l'occupation du sol 63
3.1.3 Objectif Général de l'Étude 65
3.2 Les Données 66
3.2.1 Les Données Satellites SPOT 6 66
3.2.2 Nomenclatures des classes d'occupation de sol 68
3.2.3 Création des données de couverture et d'usage du sol . . . 71
3.3 Méthodes 77
3.3.1 Les néocanaux 78
3.3.2 Les modèles 80
3.3.3 Gestion des données 82
3.4 Analyses et Performances des modèles 85
3.4.1 Les performances de la méthode XGBoost 85
3.4.2 Détection des modes d'occupation du sol 91
3.4.3 Influence des entrées sur l'apprentissage 93
3.5 Discussions et conclusions 97
3.5.1 Détails autour des résultats de la couverture du sol 97
3.5.2 Détails autour des résultats de l'usage des sols 100
3.5.3 Conclusions 103

3.1 Introduction

La notion de mode d'occupation des sols est complexe et dépendante de la zone géographique considérée. Cette section présente la zone d'étude ainsi qu'une définition de l'occupation du sol afin de mieux définir, par la suite, le contexte et l'objectif de cette étude.

3.1.1 Caractérisation de la zone d'étude

La Nouvelle-Calédonie est un territoire français d'outre-mer situé dans le Pacifique Sud et fait partie de la Mélanésie, l'un des trois grands groupements d'îles formant l'Océanie. Elle couvre une superficie de 18 575 km^2 et possède une zone économique exclusive de 1,4 million de kilomètres carrés. Le territoire, composé d'une grande île principale appelée « Grande Terre » et d'une constellation de plus petites îles l'entourant, est divisé en trois collectivités territoriales : la province des Îles, la province Nord et la province Sud (Figure 3.1).

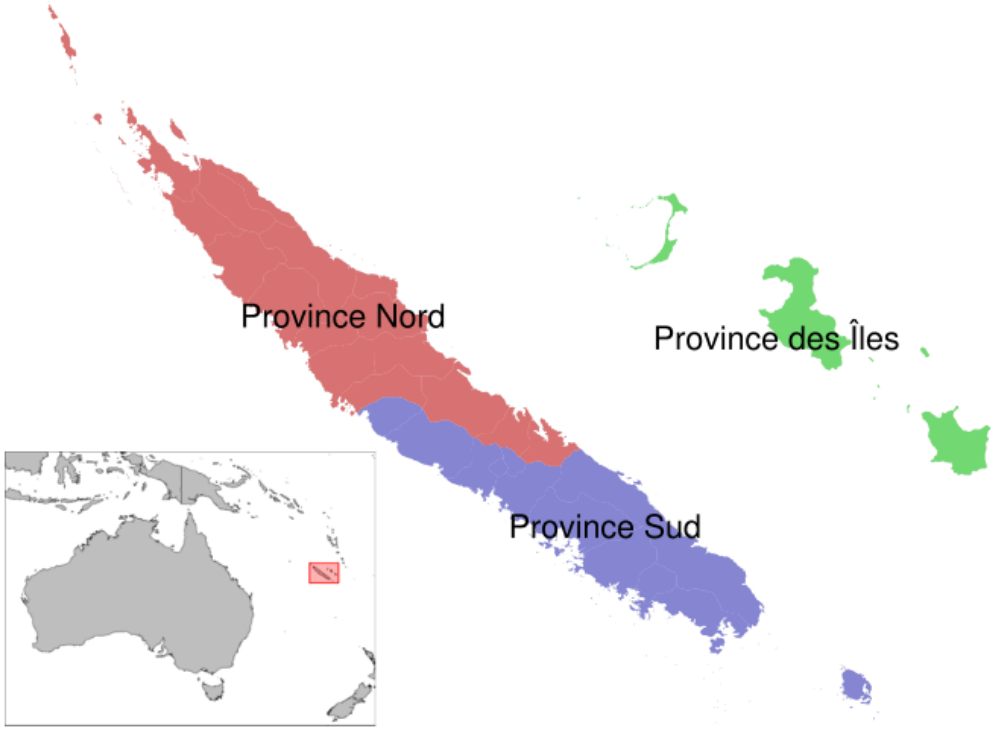


FIGURE 3.1 – Carte de la Nouvelle-Calédonie et de ses provinces

La Nouvelle-Calédonie possède un climat tropical tempéré marqué par deux saisons distinctes : la saison chaude qui couvre la période de mi-novembre à mi-avril avec des températures comprises entre 28 °C et 32 °C, et la saison fraîche qui couvre la période de mi-mai à mi-septembre avec des températures comprises entre 15 °C et 25 °C. Une forte activité cyclonique peut-être observée de décembre à mars dans la région.

La Grande Terre est une île montagneuse de 400 km de long sur 50 km de large, exposée aux alizés. En plus de la composition des sols très différente le long de la Grande Terre (Figure 3.2), cette topologie naturelle créée deux environnements très différents entre la côte Est et la côte Ouest de l'archipel. La côte Est plus humide et plus exposée aux vents, est donc plus luxuriante et abrite une forêt tropicale dense. La côte Ouest, préservée par la chaîne de montagnes qui est au cœur de l'île a un climat plus sec. Elle est principalement composée de larges espaces de savanes et le vestige d'une forêt sèche autrefois abondante.

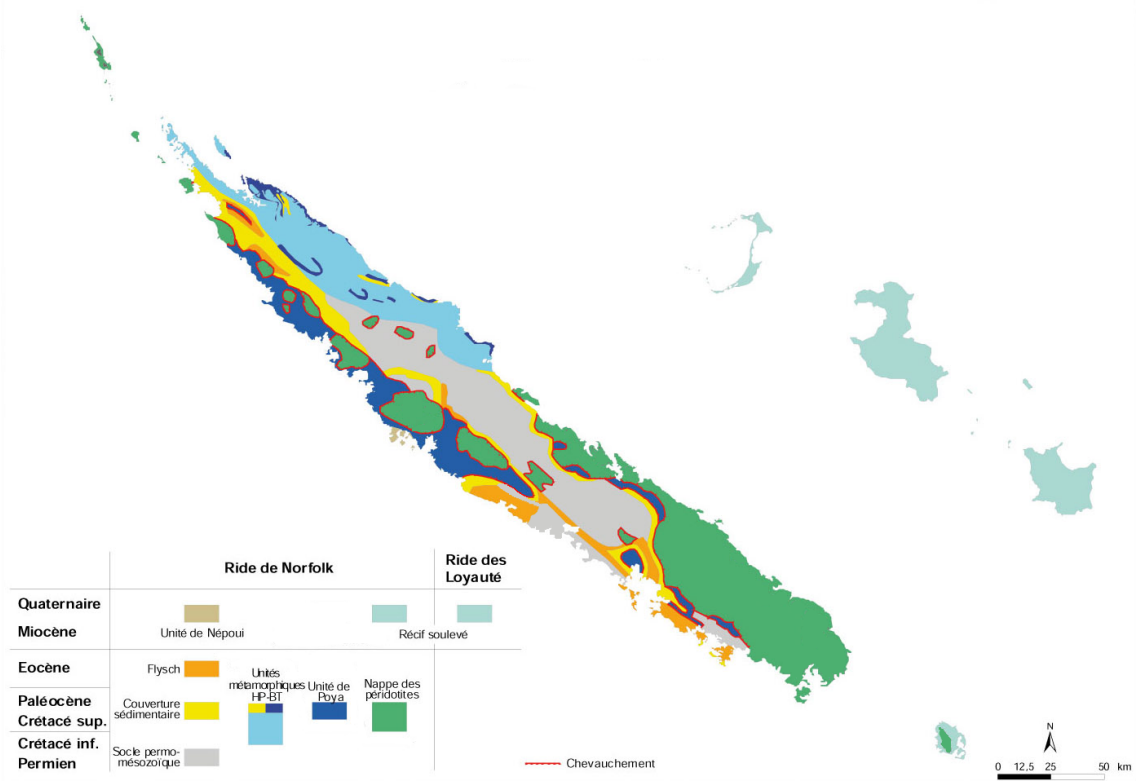


FIGURE 3.2 – Carte géologique simplifiée de la Nouvelle-Calédonie au 1 000 000ème (DIMENC/SGNC-BRGM, 2009) extrait du site internet *croixdusud.info*.

Sur la province Sud, recouverte à 90 % d'espace naturel, de nombreux milieux naturels différents sont observables, allant des maquis miniers aux mangroves du littoral. Elle possède l'un des plus hauts degrés d'endémisme au monde pour sa flore (74, 3 % [110]). Les unités de végétation primaire (forêts humides particulièrement denses et forêts sclérophylles) ont régressé au profit des unités secondaires (savanes et arbustes sclérophylles). Environ 46 % des zones d'étude sont composées de forêts humides préforestières, de forêts sèches et de mangroves. Environ 44 % sont occupées par des arbustes et/ou de la végétation herbacée, les zones avec peu ou pas de végétation occupant le reste. Les terres agricoles viennent en deuxième position avec 5 %. Les surfaces étanches ou stériles, y compris les opérations urbaines et minières, n'occupent que 3 % de la province du Sud. Les zones humides, telles que les zones submergées, partagent les 2 % restants avec les plans d'eau [111].

Il s'agit donc d'un paysage complexe et diversifié qui doit être représenté dans un ensemble de données comprenant toutes ces surfaces naturelles et ces espaces artificiels, comme les zones minières fortement présentes en Nouvelle-Calédonie. La province Sud est le centre de l'étude menée dans la cadre de cette thèse. Le territoire comptant une population municipale de 271 407 selon le Décret $n°$ 2020-157 du 25 février 2020, il s'agit de la province la plus densément peuplée de la Nouvelle-Calédonie avec 74, 8 % de la population totale y résidant. C'est donc la zone la plus sujette aux activités humaines et à l'expansion de la tache urbaine.

3.1.2 Préambule : l'occupation du sol

Dans le cadre de l'aménagement d'un territoire (construction, reboisement, etc.) et plus particulièrement lorsque celui-ci est riche en espèces endémiques, un bilan de la biodiversité existante doit être mené de façon systématique [112]. L'usage des sols et leurs changements dans l'organisation d'un territoire ont en effet des impacts à la fois directs et indirects sur la biodiversité [113]. Très souvent la multiplication des espaces artificiels, en lieu et place d'un environnement naturel et sauvage, engendre des impacts comme de l'érosion [114, 115, 116], des changements au niveau du coefficient de perméabilité du sol [117] et de la fragmentation forestière [118]. Ces impacts, qu'ils soient locaux ou globaux, de court à long terme, ne sont pas toujours bien évalués par manque de compréhension et d'informations sur les processus sous-jacents, mais sont néanmoins un enjeu majeur d'un développement durable [119].

La construction de modèles spatialisés de suivi de l'environnement est basée sur des informations géoréférencées. Or ces informations existent rarement ou, quand elles existent, ne sont pas complètement adaptées au modèle ciblé. Ces informations, collectées sur le terrain, nécessitent la mobilisation d'un opérateur sur des périodes plus ou moins longues (d'un à plusieurs mois en fonction de la complexité du terrain). Par voie de conséquence, elle est coûteuse et ne permet pas de couvrir de très grandes zones de recherche. L'analyse des images de télédétection, quant à elle, permet la récolte de grande quantité de données bien que dépendante des contraintes techniques et météorologiques. À partir de ces données, il est possible de créer des modèles pouvant établir un bilan environnemental (variation des ressources naturelles, bilan carbone des écosystèmes, etc.) d'une zone géographique et de son évolution dans le temps [120]. Si ces modèles permettent un tel suivi et aident ainsi la compréhension des processus de changements et de leurs évolutions, ils sont néanmoins entachés d'erreurs. En effet, ces erreurs sont inhérentes aux techniques employées (méthodes semi-automatiques et heuristiques imparfaites) et aux biais provenant des données utilisées (variations radiométriques des images satellitaires, corrections atmosphériques, etc.) pour établir ces modèles [121, 122].

Les cartes d'occupation du sol font partie intégrante d'un modèle spatialisé de suivi de l'environnement. La première étape de leur construction est l'établissement d'une typologie des classes répondant au mieux aux besoins identifiés. Cette tâche s'avère compliquée, car les demandes des potentiels utilisateurs sont variées et quelquefois contradictoires [123]. D'ailleurs, la communauté scientifique peine à converger vers un consensus sur une typologie et de nombreuses définitions d'un même milieu naturel apparaissent. En fonction de la filière d'étude ou du sujet d'étude, les indices mesurés ne sont pas les mêmes et intègrent des caractéristiques qui leur sont propres, les rendant alors spécifiques à certains contextes et inexploitables pour d'autres problématiques. Par exemple, dans la définition d'une lisière forestière, il est difficile, même pour un expert, d'en définir avec précision les limites. Ainsi, pour un même lieu géographique, il est courant de trouver une multitude de définitions pour décrire une même classe d'occupation du sol.

Pour simplifier la lecture de telle cartographie, une nomenclature hiérarchisée est alors proposée pour décrire ces milieux de la façon la plus générale possible. Il est a noter qu'il existe deux nomenclatures qui se distinguent dans leur définition et pouvant désigner toutes les deux l'occupation du sol créant ainsi des confusions dans leur utilisation [124] : la couverture terrestre (« land cover » en anglais) et l'utilisation des terres (« land use » en anglais).

La couverture terrestre désigne la couverture biophysique recouvrant la surface de la Terre (exemple : asphalte, eau, herbes, arbres, etc.). L'utilisation des terres désigne l'exploitation ou l'usage fait par l'homme d'un espace terrestre (ex. : zones urbaines, zones agricoles, voies d'accès, zones industrielles, etc.). À partir de ces deux nomenclatures hiérarchiques et des données satellites, une cartographie de l'occupation du sol de l'ensemble d'un territoire est alors faisable.

3.1.3 Objectif Général de l'Étude

La Nouvelle-Calédonie est définie comme étant une « île haute tropicale » dont la biodiversité est l'une des plus riches au monde [125], avec un haut degré d'endémisme de sa faune et sa flore. Elle est néanmoins soumise à de fortes pressions anthropiques et naturelles [126] et les décideurs doivent donc prêter un soin particulier à la préservation de son environnement. La disponibilité de cartes de la couverture et de l'usage du sol à jour devient alors un enjeu majeur dans l'organisation et la gestion du territoire.

L'utilisation de la résolution spectrale des images satellitaires (enregistrées dans différentes longueurs d'onde du spectre électromagnétique), couplée à diverses résolutions spatiales, s'avère efficace pour la surveillance de l'environnement [127, 128]. Parmi les techniques d'utilisation de ces informations, le Deep Learning s'impose comme référence dans l'interprétation automatique d'images [65]. En effet, dans le domaine de la télédétection optique par satellite, l'apprentissage en profondeur est devenu un outil standard qui semble bien adapté aux variations des conditions d'imagerie (éclairage, propriétés des capteurs, composition de l'atmosphère, etc.) [129, 130].

Ainsi, l'objectif de cette étude est de mettre en place un système opérationnel et automatique de surveillance de la couverture et de l'usage du sol sur l'ensemble de la Nouvelle-Calédonie, milieu insulaire complexe, à partir de l'utilisation d'imagerie satellitaire au travers d'une méthodologie basée sur le Deep Learning.

3.2 Les Données

Des données scrupuleusement étiquetées doivent être disponibles et en nombre suffisant pour entraîner des modèles de machine learning. Pour cela, sur 5 zones de la province Sud, un travail de labellisation manuelle pour les deux types de classification (la couverture et l'usage du sol) est effectué. Ce travail chronophage, associé à des données satellites SPOT 6 de très hautes résolutions ($1,5\,m$) a été mis à disposition de la communauté scientifique via un dépôt sur une plateforme collaborative de partage de donnée (doi : 10.23708/PHVOJD) [62]. Le détail des données satellites, des nomenclatures et des données labellisées à la main est présenté dans cette section.

3.2.1 Les Données Satellites SPOT 6

L'usage d'images satellites s'inscrit dans de nombreux projets visant à exploiter ces images pour la surveillance de l'environnement et comprendre les mécanismes climatiques [131]. Elles sont une source d'appui dans de nombreuses prises de décision, que ce soit dans le cadre de planification à long terme ou de situations d'urgences, comme des phénomènes naturels intenses ou de l'aide humanitaire [132, 133, 134].

La famille des satellites SPOT (Système Probatoire d'Observation de la Terre ou Satellite pour l'Observation de la Terre) est composée d'une série de capteurs et d'infrastructures terrestres qui ont été développées principalement par le CNES (Centre National d'Études Spatiales). Tous les capteurs SPOT possèdent les mêmes caractéristiques spectrales (plage spectrale et largeur de la bande), permettant aux données ainsi capturées d'être comparables entre-elles pour toute la gamme de satellites SPOT. Les méthodologies développées dans cette thèse pourront donc s'appliquer aux autres images de la constellation SPOT, en prenant en compte une phase d'adaptation des modèles du fait des résolutions spatiales différentes. Par ailleurs, la couverture spatiale des satellites SPOT est mondiale avec une répétitivité quasi journalière lors de l'agrégation des données de tous les satellites.

SPOT 6 a été lancé le 9 septembre 2012 et doit assurer sa mission jusqu'en 2024. En tandem avec SPOT 7, les deux satellites agissent ensemble pour assurer une continuité des données à haute résolution spatiale de $1,5\,m$ au maximum (fusion des canaux panchromatiques et multispectraux). Les satellites ont une fauchée de $60\,km$ de large sur $600\,km$ de long, soit une capacité d'acquisition de 3 millions de km^2 par jour et par satellite. Le satellite Spot 6 à lui seul possède une fréquence de revisite de 2 à 3 jours pour une zone géographique spécifique.

SPOT 6 dispose de 5 canaux :

- un canal panchromatique (0,455-0,745 μm) ;
- des canaux de couleurs bleu (0,455- 0,525 μm) vert (0,530-0,590 μm) et rouge (0,625-0,695 μm) ;
- et d'un canal dans le proche infrarouge (0,760-0,890 μm).

Les données multispectrales (bandes de 1 à 4) sont à $6\,m$ de résolution, mais peuvent être améliorées à $1,5\,m$ de résolution à l'aide d'un traitement ou d'une fusion avec le canal panchromatique, qui est à une résolution de $1,5\,m$. L'image résultante possède ainsi 4 canaux à $1,5\,m$ de résolution spatiale.

Les données SPOT 6 enregistrées courant 2013-2014 ont été corrigées (étapes de corrections des effets atmosphériques et des biais radiométriques inhérents au capteur) et assemblées en une mosaïque couvrant une superficie totale de $18\,575\,km^2$ de la Nouvelle-Calédonie. Les canaux Rouge, Vert, Bleu et Proche Infrarouge (PIR) sont disponibles à une résolution de $1,5\,m$. Les données ont été fournies sous forme de tuiles orthorectifiées à partir du MNT de 10 mètres (DTSI : Direction des Technologies et des Services de l'Information, gouvernement de la Nouvelle-Calédonie), pour un total de 12 images, qui sont détaillées dans le tableau B.1 en annexe.

La région de la province Sud de la Nouvelle-Calédonie concernée par les images satellites est soumise à l'influence océanique avec des vents dominants au sud-est (alizés). C'est une zone particulièrement humide de par la présence de grandes zones hydromorphes étendues possédant une série de lacs qui engendre une couverture nuageuse présente toute l'année. Il est donc très difficile d'obtenir des images satellitaires sans nuages. La mosaïque livrée par la DTSI possède ainsi un taux de couverture nuageux de $\approx 10\,\%$. De plus, les images ont été acquises tôt le matin et la province du Sud est une zone à haut relief, ce qui implique que les versants orientés sud-sud-ouest présentent des ombres importantes empêchant l'observation sur certains versants de montagne. La Nouvelle-Calédonie fait face également à de longues périodes de sécheresse provoquant des différences radiométriques importantes sur les séries d'images satellitaires. Cette variation est considérée comme négligeable pour cette étude, car les méthodes de Deep Learning s'adaptent normalement bien aux distributions de données multimodales [135].

C'est sur la base de ces images satellites que les classifications manuelles de l'utilisation des terres et de la couverture terrestre sur cinq zones d'intérêt ont été effectuées. C'est aussi à partir de ces images satellites que l'intégralité des jeux de données d'apprentissage des architectures de Deep Learning sera constituée.

3.2.2 Nomenclatures des classes d'occupation de sol

Deux nomenclatures caractérisant les modes d'occupation du sol nous intéressent particulièrement : la nomenclature désignant la couverture terrestre et celle désignant l'usage des sols. Les classes choisies pour constituer ces deux nomenclatures sont directement inspirées de travaux antérieurs réalisés par l'Observatoire de l'Environnement de la Nouvelle-Calédonie [43]. Travaux qui s'inspiraient eux-mêmes de l'inventaire biophysique européen de l'occupation des sols et de son évolution : CORINE Land Cover (CLC) [136].

Les nomenclatures de la couverture du sol et de l'usage des sols sont utilisées conjointement, car elles fournissent deux informations différentes et complémentaires qui sont parfois manquantes avec l'utilisation d'une seule nomenclature. Par exemple, une zone industrielle est définie par la composition de plusieurs types de couvertures biophysique : c'est un ensemble de bâtiments, de sols nus et de zones herbeuses. Cette double définition enlève alors l'ambiguïté d'un tel type de sol pour les architectures de Deep Learning et permettra donc une labellisation plus fine des sols tout en permettant un traitement automatique par des techniques de machine learning.

Nomenclature de la couverture du sol

Par définition, la nomenclature de la couverture biophysique des sols est basée sur la nature même des objets observés (type de sols nus comme asphalte, terre humide ou sèche, type de roches...). Or, ces informations n'ont pas été disponibles dans le cadre de cette thèse. Cette nomenclature a donc été réduite aux 5 classes suivantes (Table 3.1).

Tableau 3.1 – Nomenclature de la couverture des sols

N1	Désignation	Description
1	Bâtiments	Surfaces composant le toit des bâtiments
2	Sols nus	Ni de la végétation ni des bâtiments
3	Forêts	Groupement d'arbres
4	Végétation de faible densité	Surfaces végétales en dehors des forêts
5	Eau	Surfaces couvertes par l'eau

Il reste cependant, certaines zones naturelles qui sont difficiles à classer dans ce type de nomenclature. Entre autres, les mangroves, appartenant à la classe forêt quand elles ont une forte densité et sont classées sinon dans la végétation de faible densité. La distinction entre forêts faiblement denses et végétation de faible densité peut ici faire l'objet de confusion. Chaque cas particulier sera alors associé à une de ces 5 classes selon la subjectivité de l'opérateur humain.

Nomenclature hiérarchisée de l'usage du sol

En s'inspirant toujours de la nomenclature hiérarchisée de l'OEIL (Observatoire de l'Environnement de Nouvelle-Calédonie), une nouvelle nomenclature hiérarchisée de l'usage des sols en Nouvelle-Calédonie est proposée Table 3.2. Cette nouvelle nomenclature se décompose en deux niveaux, les colonnes N1 et N2. La colonne C1, quant à elle, désigne le numéro des classes que prendront les pixels une fois qu'ils auront été injectés dans l'architecture Deep Learning.

Tableau 3.2 – Nomenclature hiérarchisée de l'usage des sols

N1	N2	Désignation	C1
1		Territoires artificialisés	
	11	Zones urbanisées	1
	12	Zones industrielles ou commerciales et équipements	2
	13	Décharges, chantiers, extraction de matériaux, mines	3
	14	Réseaux de communication	4
	15	Pistes	5
2		Formations naturelles	
	21	Végétation de forte densité, forêts	6
	22	Végétation de moyenne densité, arbustes	7
	23	Végétation de faible densité, herbes	8
	24	Roches nues	9
	25	Zones incendiées	10
	26	Sols nus	10
3		Zones humides	
	31	Surfaces d'eau	11
	32	Engravements	12
4		Ombres et Nuages	
	41	Nuages	0
	42	Ombres	0

Le premier niveau de la nomenclature sépare quatre grandes classes :

- les territoires artificialisés, comportant le tissu urbain, les routes et tous les aménagements faits par l'homme ;
- les formations naturelles, qui comportent, comme son nom l'indique, toutes les zones naturelles à l'exception des zones humides. Les zones incendiées sont classées comme des surfaces naturelles, même s'il s'agit d'un feu non accidentel (culture sur brûlis par exemple) ;
- les zones humides, comportant l'engravement, car en plus d'être un produit des rivières, ces zones peuvent être considérées comme inondables ;
- les ombres et nuages ne sont pas des surfaces terrestres par leur nature. Elles doivent être incluses dans la nomenclature néanmoins, afin d'identifier la surface observée d'une part, et pour pouvoir les écarter de l'apprentissage d'autre part.

Le deuxième niveau de la nomenclature subdivise donc chacune de ces 4 classes en plusieurs sous-classes. Ces sous-classes ont été identifiées en particulier pour leur importance dans la gestion et la conservation des terres en Nouvelle-Calédonie. Par exemple, le contexte minier de cet archipel donne une importance particulière à cette nomenclature, car l'impact environnemental de cette activité nécessite une surveillance régulière. Leur identification est importante pour des tâches telles que la surveillance de la fragmentation des forêts ou la prévention de l'érosion des sols.

Comme mentionné précédemment, sont inclus également des classes qui n'appartiennent pas dans les nomenclatures de la couverture et de l'usage des sols : les nuages et les ombres. Le masque nuageux est un produit de télédétection fourni avec les données SPOT 6 et les ombres peuvent être détectées avec une simple heuristique délimitant les régions « presque » noires (à valeurs spectrales faibles). Pour les architectures de Deep Learning, si un soin particulier n'est pas apporté pour identifier ces régions, car non pertinentes pour l'étude, un masque doit être néanmoins appliqué sur ces régions.

Un numéro est attribué à chaque classe de la nomenclature, ici de 0 à 12 (colonne C1). La classe 0 contient les nuages et les ombres, car elle a été ignorée pendant le processus d'apprentissage. Si deux classes comportent le même chiffre alors elles ne sont pas distinguées dans l'apprentissage et dans les résultats finaux des modèles, même si elles ont été distinguées dans la labellisation à la main.

On peut remarquer que dans le Tableau 3.2, les classes « Zones incendiées » et « Sols nus » ont le même numéro de classe. Il a été décidé de fusionner ces classes à cause du trop petit échantillon de données représentant les zones incendiées. De plus, à cause de la temporalité, certaines de ces zones n'étaient pas au même stade de récupération de la végétation : certaines présentent encore des traces d'incendies alors que d'autres n'étaient pas distinguables d'une zone de sols nus ou de végétation de faible densité.

3.2.3 Création des données de couverture et d'usage du sol

En tenant compte des régions inaccessibles, de la grande diversité des milieux et de la superficie du territoire, il est difficile, voire impossible, de collecter des informations de terrain. Les données déjà existantes de télédétection documentant la Nouvelle-Calédonie ne sont pas en nombre suffisant pour établir une base d'apprentissage et certaines d'entre elles ne sont pas adaptables à la problématique de cette étude. Il a été donc décidé pour cette étude, d'étiqueter, à dire d'expert, manuellement plusieurs zones du territoire de la province Sud par le biais d'une photo-interprétation de données satellites SPOT 6.

Sélection des zones d'intérêts

Le choix des zones d'intérêts doit répondre aux critères suivants : (i) représenter toute la diversité du territoire ; (ii) concerner une surface suffisamment grande pour illustrer chacune des classes des nomenclatures d'usage et de couverture du sol, dans le but de satisfaire des modèles d'apprentissage automatique ; (iii) et représenter une surface suffisamment petite pour pouvoir effectuer une labellisation à la main dans des temps raisonnables.

Une sélection de 5 zones en particulier a été au final réalisée (Figure 3.3). Nommées à partir des communes où elles sont situées, elles sont réparties dans toute la province Sud pour une étendue cumulée de $128,4\,km^2$ et contiennent une variété d'environnements, notamment des zones minières, des zones urbaines et des forêts tropicales. La description, l'étendue et la surface des cinq zones sont décrites dans le tableau 3.3. Les données sont projetées selon le système de coordonnées de référence RGCN91-93 / Lambert Nouvelle-Calédonie (EPSG :3163). En plus des zones indiquées sur la figure 3.3, la surface et les coordonnées en haut à gauche (x_1,y_1) et en bas à droite (x_2,y_2) sont décrites dans le tableau 3.3.

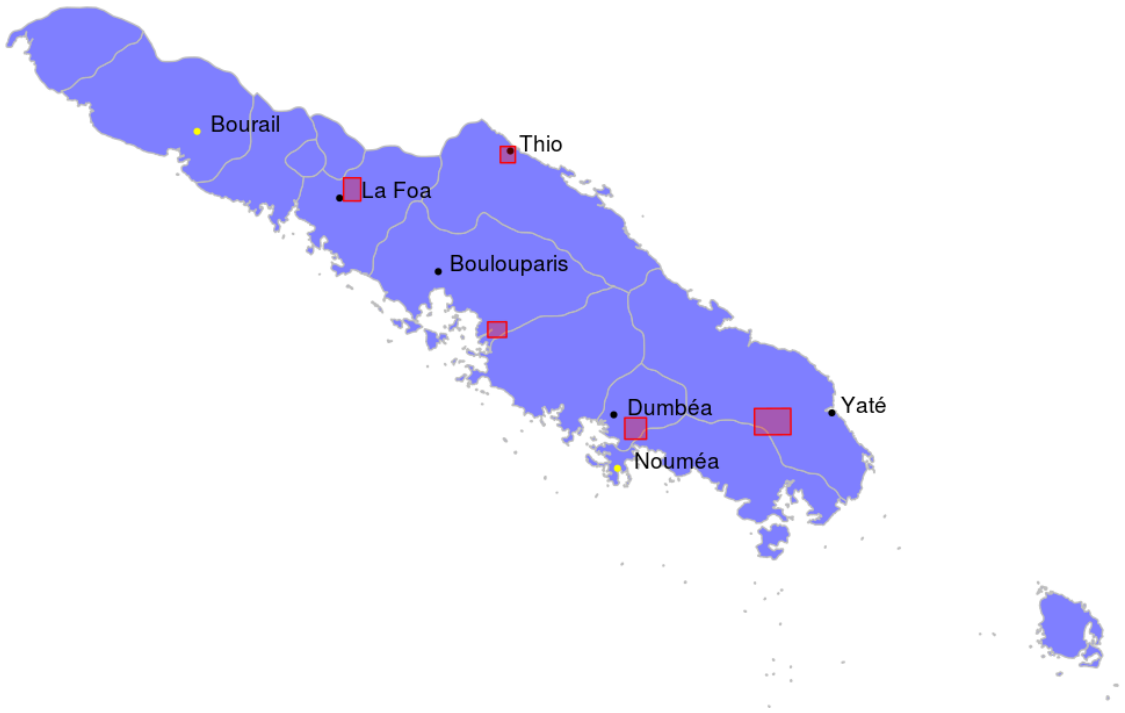


FIGURE 3.3 – Carte de la province Sud de Nouvelle-Calédonie situant les 5 zones sélectionnées pour l'étude dont le détail est présenté Tableau 3.3.

TABLEAU 3.3 – Détails des 5 zones d'études sélectionnées, présentant les points en haut à gauche (x_1,y_1) et en bas à droite (x_2,y_2) de chaque emprise dans le système de coordonnées EPSG :3163.

Nom des zones	(x_1,y_1)	(x_2,y_2)	**Surface**
Thio	(419828,285110)	(423328,288915)	$13,3km^2$
La Foa	(382983,276240)	(387031,281621)	$21,8km^2$
Boulouparis	(416813,244503)	(421228,248117)	$25,3km^2$
Dumbéa	(448857,220743)	(453938,225718)	$52,1km^2$
Yaté	(479224,221634)	(487729,227756)	$15,9km^2$

Photo-interprétation des images SPOT 6

Une segmentation à la main sous SIG (*QGIS*) de l'espace géographique est effectuée à partir des images satellites SPOT 6 (Figure 3.4). Une fois l'espace segmenté, deux classes ont été attribuées pour chaque polygone, une classe correspondant à la couverture terrestre et l'autre à l'usage du sol, en concordance avec les deux nomenclatures définies ci-devant dans les tableaux 3.1 et 3.2. Les deux classes sont attribuées en même temps pour garantir des résultats cohérents et permettre d'économiser du temps, puisque les deux nomenclatures se confondent sur une grande partie du territoire (en particulier sur les terres naturelles et non cultivées).

Les figures 3.5 et 3.6 montrent respectivement la segmentation de l'espace géographique de la couverture terrestre et de l'usage des sols des 5 zones sélectionnées pour l'étude. C'est à partir de ces classifications que les jeux de données d'entraînement, de validation et de test ont été constitués.

Mise à disposition des données à la communauté scientifique

La communauté scientifique dispose à ce jour de très peu de jeux de données de télédétection à très haute résolution spatiale (métrique ou submétrique) pour les tâches d'apprentissage machine. Un jeu de données de référence a donc été créé à partir des données décrites précédemment et sera disponible à la communauté scientifique sur une plateforme de partage de l'IRD (Institut de Recherche pour le Développement). Ce jeu de données comporte les extractions des images satellites SPOT 6 (celles à partir desquelles le travail de classification manuelle a été mené) et les couches vectorielles d'expertise qui ont servi aux classifications et leurs légendes (les nomenclatures). Ces données vectorielles ont été stockées dans un format standard de données géospatiales : format geopackage.

L'ensemble de ces données a été contrôlé et publié sur l'entrepôt de données DataSuds, créé par l'IRD. Ces données sont accessibles sous l'intitulé « SPOT6 satellite imagery and land cover/land use classification of 5 areas in the South Province of New Caledonia » à partir du doi : 10.23708/PHVOJD [62].

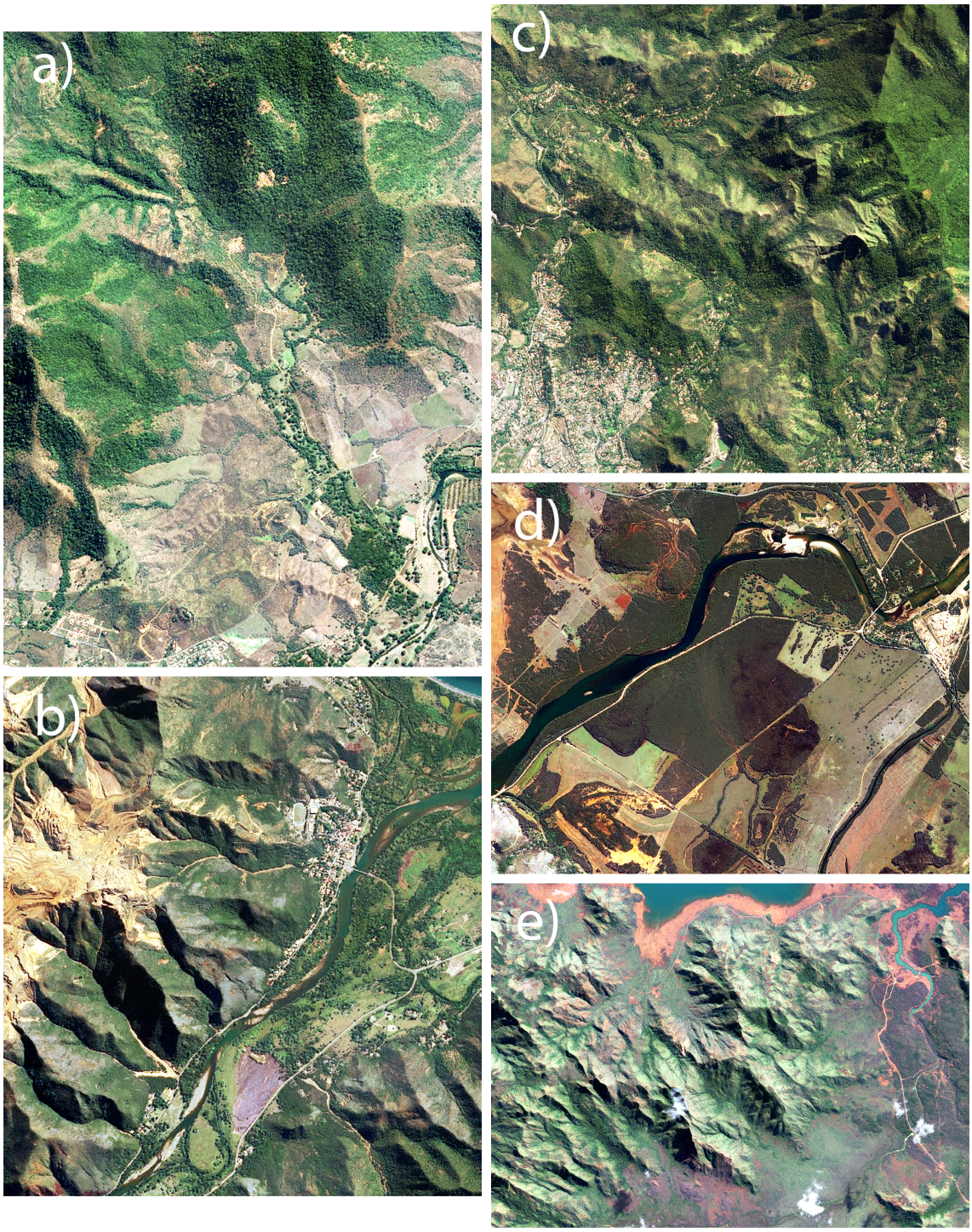


FIGURE 3.4 – Images satellites SPOT 6 des zones sélectionnées dans les communes de La Foa (a), Thio (b), Dumbéa (c), Boulouparis (d) et Yaté (e).

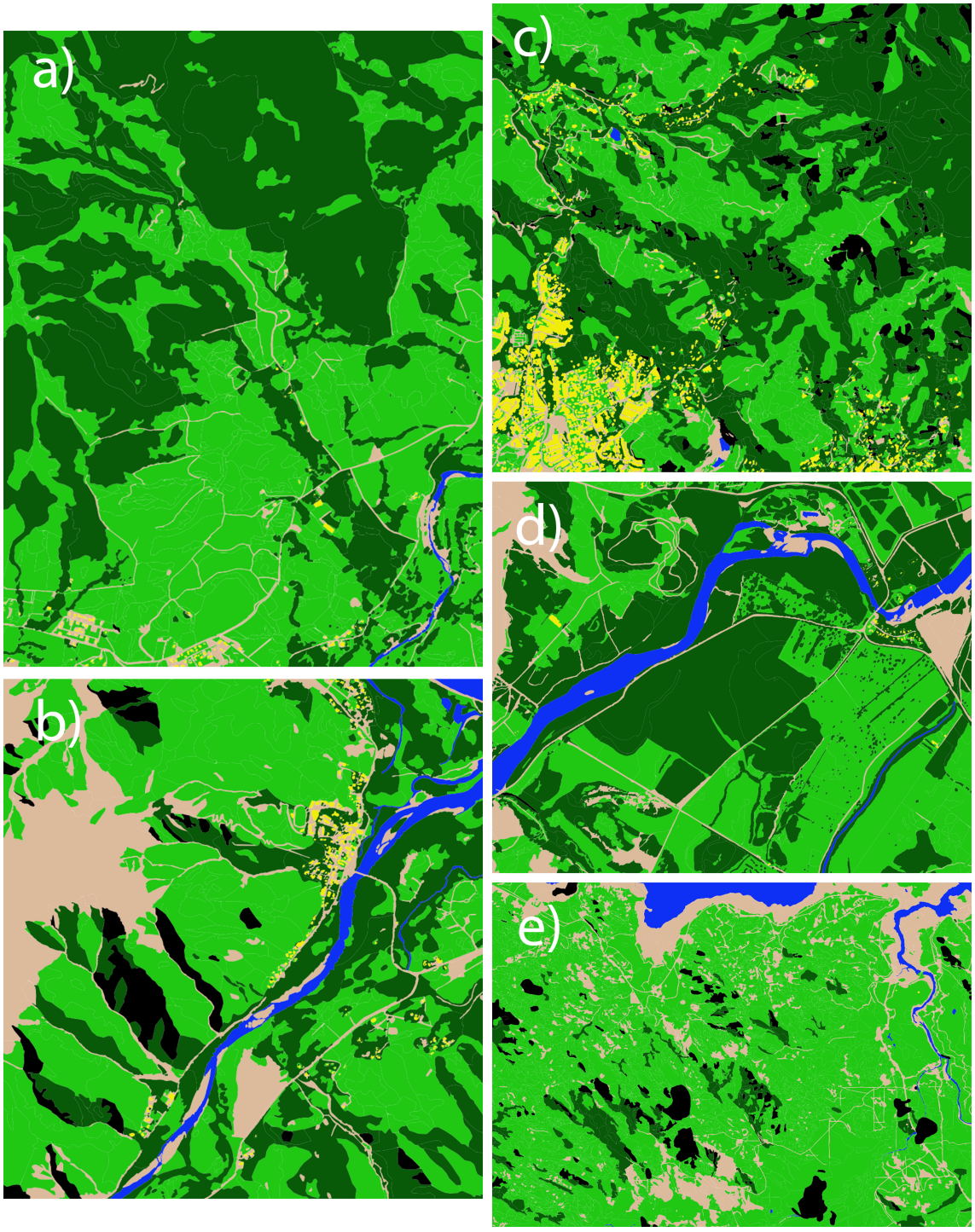


FIGURE 3.5 – Cartographie de la couverture du sol selon la nomenclature présentée Table 3.1 pour les zones sélectionnées dans les communes de La Foa (a), Thio (b), Dumbéa (c), Boulouparis (d) et Yaté (e).

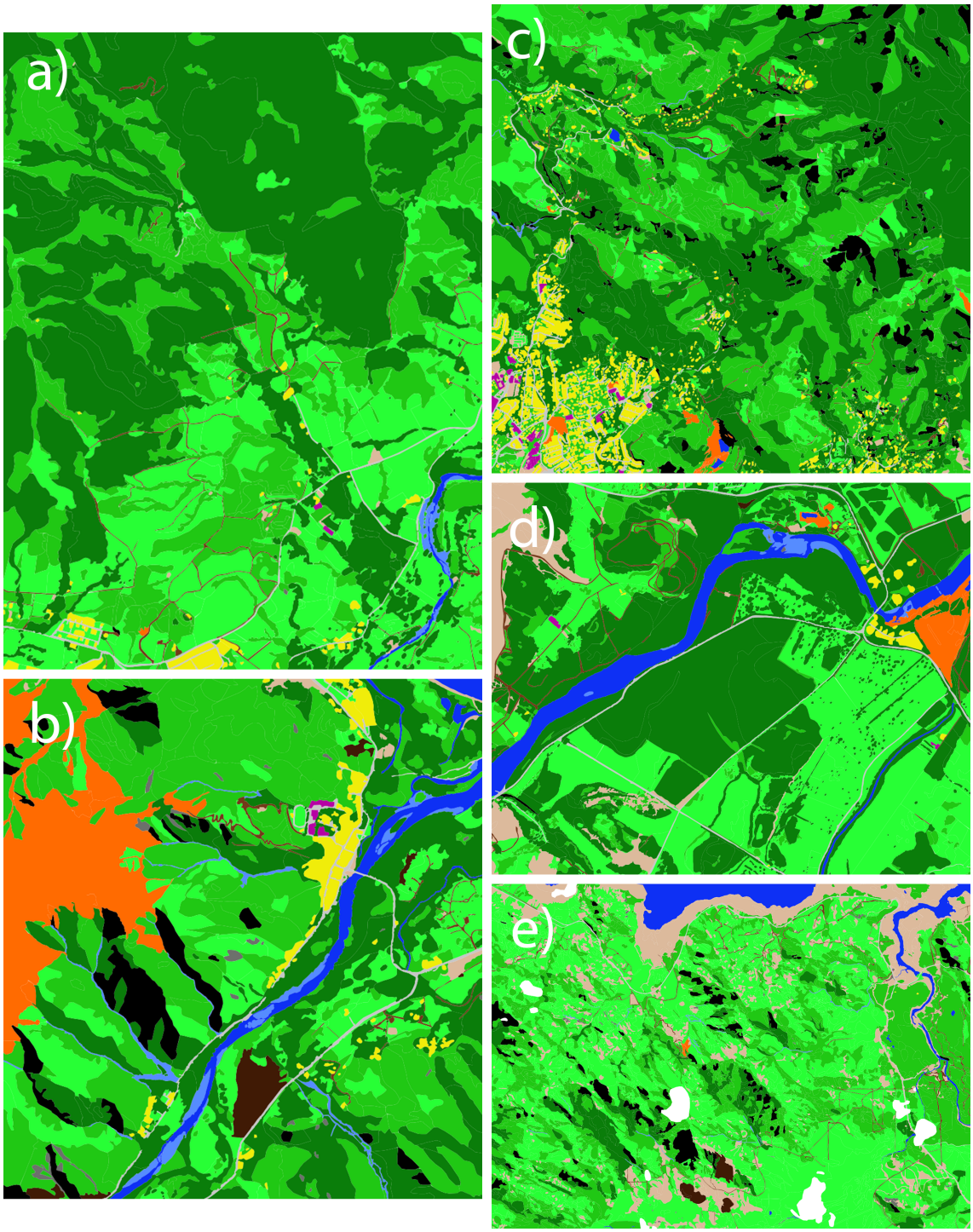


FIGURE 3.6 – Cartographie de l'usage du sol selon la nomenclature présentée Table 3.2 pour les zones sélectionnées dans les communes de La Foa (a), Thio (b), Dumbéa (c), Boulouparis (d) et Yaté (e).

3.3 Méthodes

Dans ce chapitre, plusieurs architectures de Deep Learning pour la classification de la couverture du sol et de l'usage du sol ont été comparées. Cette comparaison s'est fait sur 5 régions spécifiques de la province Sud en Nouvelle-Calédonie.

Il existe de nombreuses et puissantes architectures de Deep Learning dans le domaine de la télédétection qui utilisent deux stratégies possibles : « l'étiquetage du pixel central » ou « la segmentation sémantique » (décrites respectivement dans les sections 2.4 et 2.5). L'objectif est de sélectionner l'architecture d'apprentissage profond la mieux adaptée au cas de la Nouvelle-Calédonie, tout en modifiant le moins possible son architecture interne et sa configuration.

Un certain nombre d'architectures de Deep Learning ont été comparées entre elles, mais aussi par rapport à une technique de machine learning plus classique, XGBoost, qui a montré de très bonnes performances dans plusieurs travaux de détection de la couverture et de l'usage du sol [137] et d'une manière générale de classification de données tabulaires [138]. Comme les « néocanaux » dérivés par des combinaisons arithmétiques des canaux d'images brutes sont depuis longtemps populaires en télédétection, nous évaluons également la nécessité d'une telle ingénierie manuelle des caractéristiques dans le contexte du Deep Learning, même si les architectures de Deep Learning sont capables de créer leurs propres combinaisons non linéaires à partir des données brutes.

La cartographie de l'usage du sol est une tâche plus complexe que la cartographie de la couverture terrestre. La cartographie de la couverture du sol est généralement obtenue par traitements (photo-interprétation, classifications automatiques...) d'images satellites et/ou aériennes. La cartographie des usages nécessite généralement l'intégration de données exogènes (carte du réseau routier, zones inondables, zones agricoles, etc.) et de relevés au sol, ces informations n'existant pas sur les images satellitaires. Cependant, pour cette étude, ces sources de données exogènes ne sont pas disponibles et les informations sur l'usage du sol ont été obtenues par expertise à partir de photo-interprétation.

En outre, dans cette étude, la nomenclature de l'utilisation des terres comporte un nombre plus élevé de classes distinctes que la nomenclature de la couverture du sol. Mais les résultats de cette étude montreront que, lorsque le résultat de la classification de la couverture du sol est utilisé comme étiquette en entrée de la tâche de classification de l'usage du sol, les performances de la classification de l'usage du sol sont améliorées.

3.3.1 Les néocanaux

Les 4 canaux bruts de SPOT 6 utilisés dans cette étude sont enregistrés dans les longueurs d'onde R (rouge), V (vert), B (bleu) et PIR (proche infrarouge), respectivement. À partir de ces canaux bruts, des néocanaux sont calculés pour permettre de meilleures discriminations des objets observés tels que les infrastructures urbaines [139, 140]. L'un des néocanaux calculés, l'indice NDVI [141], est utilisé pour mettre en évidence l'activité photosynthétique de la couverture végétale. Certains de ces néocanaux utilisent même des combinaisons non linéaires des canaux bruts pour souligner certaines caractéristiques de l'image (la luminance, la couleur c_3, etc.).

Les néocanaux sont potentiellement importants pour des méthodes de machine learning telle que XGBoost, car ils ajoutent de nouvelles informations aux modèles, permettant ainsi de meilleures performances. Les néocanaux utilisés dans cette étude sont décrits dans le tableau 3.4.

TABLEAU 3.4 – Détails des différents néocanaux utilisés

Description du néocanal	**Équation**
La luminance **L**, appartenant au système de couleurs HSL (Hue, Saturation, Luminance)	$\frac{max(R,V,B)+min(R,V,B)}{2}$
La couleur $\boldsymbol{c_3}$, issue du système de couleurs $c_1c_2c_3$ [142]	$arctan\left(\frac{B}{max(R,V)}\right)$
Le **NDVI** (Normalized Difference Vegetation Index [141])	$\frac{PIR-R}{PIR+R}$
Le **MSAVI** (Modified Soil-Adjusted Vegetation Index [143])	$\frac{(PIR-R)\times(1+L)}{PIR+R+L}$
Le **ExG** (Excess Green index [144])	$\frac{2\times V-R-B}{R+V+B}$
Le **MNDWI** (Modified Normalized Difference Water Index [145])	$\frac{V-PIR}{V+PIR}$

Les indices NDVI, MSAVI et ExG [146] rehaussent les caractéristiques de la végétation dans une image de télédétection L et c_3 fournissent des informations sur le contraste et l'illumination [142]. L'utilisation de ces néocanaux, qui sont pour la plupart des rapports spectraux, permet de réduire l'effet de variation de l'illumination solaire causé par la topographie. Par exemple, une forêt sur le versant d'une montagne n'aura pas la même réflectance en fonction de l'orientation du soleil. L'utilisation de rapports plutôt que de valeurs brutes des canaux d'origines permet de limiter ce phénomène. La Figure 3.7 présente la sortie de quelques néocanaux pour l'image SPOT 6 de la région de Thio.

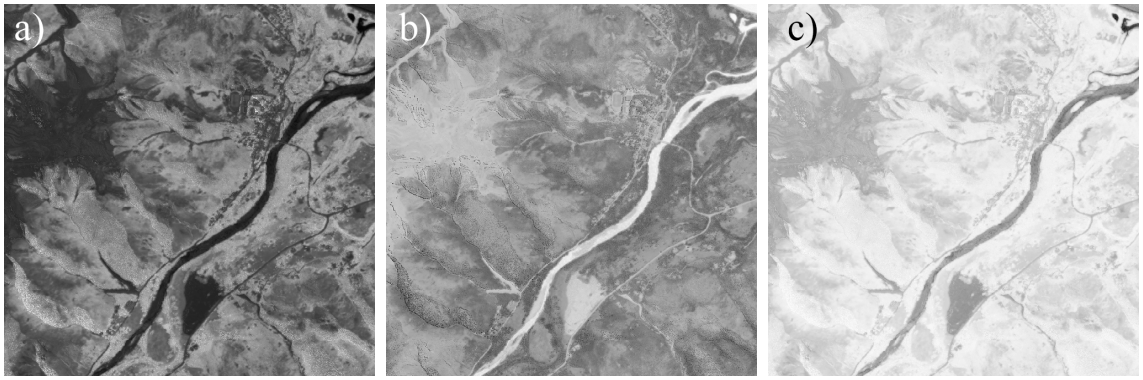


FIGURE 3.7 – Exemple des néocanaux NDVI (a), MNDWI (b) et MSAVI (c) appliqués à l'image satellite SPOT 6 de la commune de Thio.

À partir de ces néocanaux, il est possible d'appliquer un traitement supplémentaire pour obtenir de nouvelles caractéristiques de l'image [147]. Ainsi, l'analyse de la texture, issue d'un calcul qui prend en compte la valeur du pixel donné ainsi que les valeurs des pixels voisins, permet d'obtenir des informations sur l'arrangement spatial des objets dans une image. Par exemple, la différence de granularité entre une forêt et une plaine herbeuse dans l'espace géographique d'une image peut être détectée en appliquant une succession de filtres de texture différents sur les canaux bruts ou même les néocanaux. Dans la littérature, l'utilisation de ces textures conjointement avec les canaux initiaux d'une image satellitaire augmente la performance des modèles [148, 149].

Haralick propose, parmi toutes les textures possibles, 14 descripteurs statistiques résumant l'information texturale disponible dans une image [150]. Parmi ces textures d'Haralick, un petit nombre d'entre elles (les textures de contraste, corrélation, dissimilarité, entropie et homogénéité) sont particulièrement utilisées dans le milieu du traitement d'image [151]. Un exemple de ces textures est présenté Figure 3.8.

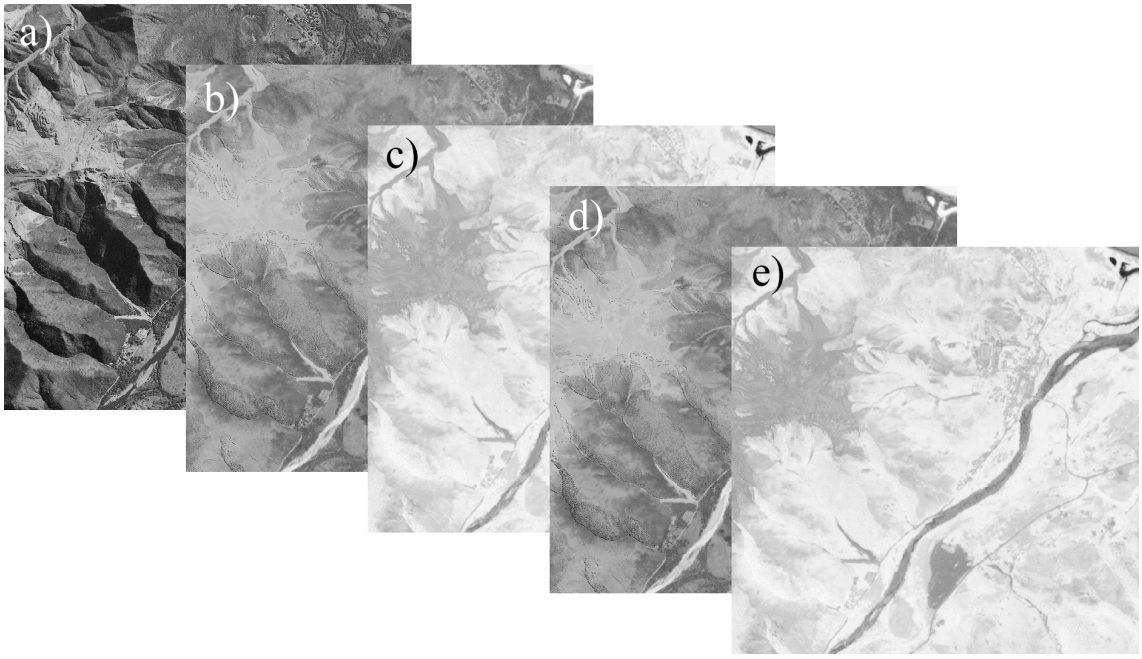


FIGURE 3.8 – Exemple de textures de contraste (a), corrélation (b), dissimilarité (c), entropie (d) et d'homogéinité (e) appliquées à l'image satellite SPOT 6 de la commune de Thio.

Les canaux bruts, les néocanaux et les différentes textures sont utilisés comme entrée pour l'entraînement de certains modèles de machine learning comme XGBoost [152, 153].

3.3.2 Les modèles

Deux types de modèles de machine learning sont utilisés dans cette étude : une technique de machine learning classique XGBoost et des architectures de Deep Learning.

XGboost

XGBoost [154] est un algorithme d'arbres de *Gradient Boosting*, qui consiste à agréger des arbres de décision de faible capacité d'apprentissage (comportant très peu de branches de décision) pour produire un modèle de décision plus robuste. À chaque itération, l'algorithme calcule la performance du modèle intermédiaire et utilise les données résiduelles, issues de la différence entre la donnée prédite et la donnée observée, comme cible d'apprentissage du modèle suivant. Cette approche itérative s'effectue jusqu'à ce que le modèle global converge, c'est-à-dire lorsque le

modèle ne parvient plus à réduire la fonction de coût d'une itération à une autre.

Dans les faits, XGBoost construit itérativement plusieurs ensembles d'arbres sur des sous-ensembles de données en pondérant leurs prédictions individuelles en fonction de leurs performances. Une prédiction d'ensemble est ensuite calculée en prenant la somme pondérée de tous les apprenants. La bibliothèque est conçue pour être très efficace au niveau du temps de calcul et fournit des algorithmes parallèles pour le renforcement des arbres (également connu sous le nom de GBDT, GBM) qui sont particulièrement bien adaptés aux grands volumes de données comme les images satellitaires submétriques. XGboost est une technique de pointe en matière de classification supervisée qui a montré de très bonnes performances sur une grande variété de tâches de référence [154].

Pour exécuter un modèle XGBoost, des néocanaux et de multiples filtres de texture ont été utilisés. Les filtres sont : dissimilarité, entropie, homogénéité et moyenne. Des images de 64×64 sont utilisées comme entrée pour l'étiquetage du pixel central. Cette taille de fenêtre correspond à la taille de la fenêtre utilisée pour les architectures de Deep Learning de telle sorte que les bases de données d'apprentissage soient strictement équivalentes afin de pouvoir comparer les performances des modèles entre eux.

Les architectures de Deep Learning

Dans le cadre de la classification des données satellitaires en plusieurs classes (labellisées l), la première couche d'une architecture de Deep Learning reçoit en entrée, généralement, une image de k canaux de $n \times m$ pixels. Les couches convolutionnelles agissent comme des filtres qui extraient les caractéristiques pertinentes de l'image. La mise en commun des couches permet ensuite de sous-échantillonner les réponses des filtres à une résolution plus faible afin d'extraire des caractéristiques de plus haut niveau avec un contexte spatial plus large. Les dernières couches mettent en correspondance les caractéristiques résultantes avec les différentes classes à prédire.

Il y a deux stratégies possibles : L'étiquetage du pixel central, où une couche entièrement connectée met en correspondance les caractéristiques calculées sur l'ensemble d'une zone de l'image avec un vecteur de pseudo-probabilités de $1 \times l$ pour le seul pixel central de la zone ; ou la segmentation sémantique, où les caractéristiques de haut niveau sont interprétées comme un codage latent et décodées pour obtenir une carte de probabilités par pixel de $1 \times n \times m$ pour l'ensemble de la zone, où le décodeur est une autre séquence de couches de ré-échantillonnage et de convolution.

Les architectures utilisées dans cette partie sont donc les 3 architectures de labellisation du pixel central de l'image (Section 2.4) : AlexNet (Sous-section 2.4.1), ResNet (Sous-section 2.4.2) et DenseNet (Sous-section 2.4.3) ; et les 3 architectures de segmentation sémantique (Section 2.5) : FCN (Sous-section 2.5.1), SegNet (Sous-section 2.5.2) et DeepLab (Sous-section 2.5.3).

Les paramètres internes des architectures CNN utilisées n'ont pas été modifiés par rapport à ceux proposés à l'origine. L'apprentissage des différentes architectures a été entrepris avec une descente de gradient stochastique en utilisant un protocole similaire : un momentum de $0,9$ et un pas d'apprentissage initial de 10^{-2}. Pour toutes les 10 itérations de l'intégralité du jeu de données, le pas d'apprentissage est divisé par 10 jusqu'à atteindre une valeur de 10^{-6}.

3.3.3 Gestion des données

Les données du satellite SPOT 6 des 5 zones d'étude ont été prétraitées afin d'être intégrées dans les différentes architectures de Deep Learning et le modèle XGBoost. Les données ont été divisées en trois parties : un ensemble d'apprentissage, un ensemble de validation et un ensemble de test totalement indépendant des deux précédents.

Quatre des cinq zones sont utilisées pour l'apprentissage et la validation. La dernière zone, la commune de Thio, est utilisée comme ensemble de test. Cette dernière image contient toutes les classes des deux nomenclatures : soit, les 5 classes de couverture du sol et les 12 classes d'utilisation des sols. De plus, cette image est représentative du contexte environnemental calédonien et comprend : l'environnement urbain, le minier, les zones montagneuses et forestières et s'étendant du littoral jusqu'aux zones montagneuses intérieures. C'est sur cette scène que la matrice de confusion finale et les mesures de qualité ont été calculées.

Prétraitements des données satellites et échantillonnage

La constitution du jeu de données est une étape importante, car elle influe grandement les résultats finaux des modèles de Deep Learning. Que le jeu de données soit équitablement réparti ou non, la résolution des images, leur découpage, et la sur-représentation/sous-représentation des données ont un impact direct sur l'apprentissage des architectures de Deep Learning [155].

Le jeu de données d'apprentissage en entrée du modèle doit être au format « raster » (grille de valeurs), or celles-ci sont sous forme « vecteur » (polygone, polyligne

ou points). Une transformation de format vecteur en raster est donc nécessaire au préalable de l'application du modèle.

L'image satellite et la classification du sol sont découpées en une série d'images plus petites et de taille fixe. Une fenêtre coulissante, dont la taille et le pas sont fixés, échantillonne l'intégralité de l'image. Le pas est fixé à une valeur inférieure à la taille de la fenêtre coulissante, de cette manière une même région de l'image est échantillonnée plusieurs fois, comme présenté Figure 3.9. Cela permet de garder une continuité de l'espace dans l'apprentissage et d'éviter les effets de bords aux limites des images.

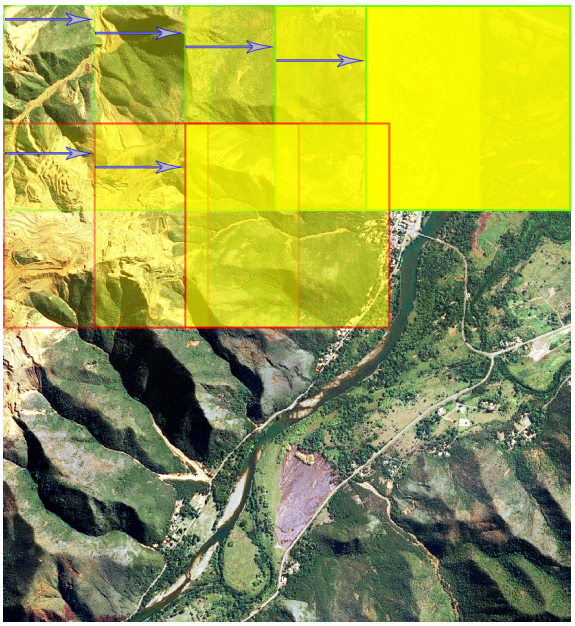


FIGURE 3.9 – Exemple d'une fenêtre coulissante échantillonnant l'image satellite SPOT 6 de la commune de Thio.

Pour la création des jeux de données, la méthode d'échantillonnage stratifié aléatoire est appliquée. Cette méthode possède plusieurs étapes, elle consiste d'abord à placer aléatoirement un nombre minimum d'observations dans chaque catégorie (ou classe). Ensuite d'homogénéiser la taille des classes (nombre de pixels par classe) en se basant sur la taille de la classe minoritaire. Dans le cas de la segmentation sémantique, où l'ensemble de la fenêtre est classifiée et pas seulement le pixel central, la méthode d'équilibrage de la fréquence médiane [156] est appliquée. Les fréquences de chaque classe sont insérées comme paramètres de poids dans les architectures de Deep Learning. Ce paramètre permet de favoriser, dans le calcul du coût, les classes sous-représentées.

Création des jeux de données

Plusieurs combinaisons possibles de canaux d'entrée ont été testées pour les méthodes XGBoost et Deep Learning. Pour la classification de l'utilisation et de l'occupation des sols, un premier ensemble de données, constitué uniquement des 4 canaux bruts de SPOT 6 (canaux rouge, vert, bleu et proche infrarouge), a été utilisé comme base. Les autres jeux de données étaient composés, en plus de ces canaux bruts, de 6 néocanaux supplémentaires correspondants aux indices NDVI, MSAVI, MNDWI, L, c_3 et ExG. En plus de ces entrées, le résultat de la classification de la couverture du sol peut être utilisé comme canal supplémentaire en entrée d'une tâche de classification de l'usage du sol. Les différentes variantes de jeux de données sont résumées dans le tableau 3.5.

Tableau 3.5 – Synthèse des différents jeux de données utilisés par des méthodes d'apprentissage machine.

Jeux de données	**Description**
RVBPIR	Jeu de données contenant uniquement les canaux bruts : R, V, B et PIR
NEO	Jeu de données contenant uniquement des néocanaux : L, c_3, NDVI, MSAVI, MNDWI et ExG
LCE	Couverture du sol produite par un opérateur humain
LCM	Couverture du sol produite par une méthode d'apprentissage machine

Les jeux de données ainsi constitués seront stockés dans un fichier de type *hdf*5 qui permet un stockage physique important de la donnée en optimisant la place mémoire. En effet, le format *hdf*5 offre la possibilité de charger et décharger rapidement la donnée lors d'un apprentissage profond et donc manipuler de larges jeux de données rapidement.

3.4 Analyses et Performances des modèles

Après avoir fixé les paramètres de XGBoost et des différentes architectures de Deep Learning, les modèles sont exécutés sur l'ensemble de la zone de test. Une cartographie complète de la couverture et de l'usage du sol de la zone de la commune de Thio est donc obtenue.

Pour les architectures utilisant une méthode de labellisation centrale des pixels (2.4), la cartographie se fait pixel par pixel. La méthode XGBoost appartient à cette catégorie. Pour éviter les effets de bords, un tampon de 32 pixels est appliqué tout autour de la zone de test. Les pixels dans cette zone tampon sont fixés à 0, la valeur nulle des modèles. Pour les architectures de labellisation sémantique, une fenêtre coulissante avec un pas de 16 (choisie empiriquement) est appliquée. Agrandir ce pas détériore les résultats et le réduire n'apporte pas d'améliorations significatives et augmente le temps de calcul. Là où la fenêtre coulissante se chevauche, une moyenne des probabilités résultantes par classe est appliquée. La classe ayant obtenu le score de probabilité le plus élevé est retenue comme label du pixel.

Des matrices de confusion sont extraites de ces résultats et quatre mesures de qualité y sont utilisées : la précision globale (PG), la précision du producteur (PP), la précision de l'utilisateur (PU) et le F1-score (F1s). La précision globale est calculée par la somme de la diagonale de la matrice de confusion divisée par le nombre de classes. La précision utilisateur prend le nombre d'individus bien classés divisé par la somme des lignes. La précision du producteur prend le nombre d'individus bien classé, divisé par la somme de la colonne de la matrice de confusion. Enfin, le F1-score est calculé par la moyenne harmonique de la précision et du rappel, c'est à dire ici de la PP et la PU. Notez que les zones d'ombres et les nuages ne sont pas pris en compte dans la matrice de confusion.

3.4.1 Les performances de la méthode XGBoost

Les performances de la méthode XGBoost serviront de base de comparaison pour les architectures de Deep Learning. Il est à savoir également que la méthode de XGBoost permet, une fois le modèle créé, d'estimer les variables les plus discriminantes de l'information dans l'arbre de décision. Cette valeur est calculée en prenant en compte le gain de performances du modèle lorsque l'arbre se scinde grâce à cette variable et le nombre d'observations que cette variable influence dans le modèle.

Spécifiquement pour les jeux de données de XGBoost, un jeu de données supplémentaire est créé. Ce jeu de donnée, nommé *All*, intègre toutes les entrées possibles discutées précédemment, c'est-à-dire tous les canaux bruts, tous les néocanaux et toutes les textures appliqués aux néocanaux. Le premier modèle de XGBoost est utilisé sur ce jeu de données pour obtenir le graphe d'importance des variables. À partir de ces graphes, de nouveaux jeux de données prenant en compte les variables d'importance sont créés. Soit n le nombre des n premières variable pris dans le graphe dans leur ordre d'importance afin de créer un jeu de données qui sera nommé : $Opti_n$.

XGBoost et la classification de la couverture des sols

Pour la tâche de classification de la couverture des sols, la base de données contenant toutes les variables (canaux bruts, néocanaux et textures) a été testée en premier afin d'obtenir le graphe d'importance des variables (Figure 3.10). Les variables ayant plus de 0,4 et 0,6 d'importance dans le modèle sont utilisées pour créer les jeux de données $Opti_6$ et $Opti_9$ respectivement. Les résultats de ces jeux de données ainsi que des autres mentionnés précédemment sont présentés dans le tableau 3.6.

TABLEAU 3.6 – Résultats de la méthode XGboost pour la tâche de classification de la couverture du sol.

Jeux de données	**PG**	**PU**	**PP**	**F1s**
RVBPIR	73,88	69,99	71,53	70,75
RVBPIR+NEO	73,39	68,78	73,42	71,02
NEO	71,76	67,73	72,68	70,12
All	77,33	72,43	74,64	73,52
$Opti_6$	73,28	69,3	75,59	72,31
$Opti_9$	**77,55**	**72,39**	**76,93**	**74,6**

Le modèle XGBoost $Opti_9$ a les meilleures performances (résultat en gras dans le tableau 3.6), le modèle utilisant toutes les données affiche des performances juste un peu plus dégradées. Le modèle n'utilisant que les néocanaux (*NEO*) présente les plus mauvaises performances. Il est visible par ces résultats que le simple ajout de néocanaux aux données brutes n'améliore pas sensiblement les résultats et que les néocanaux ne peuvent pas se substituer à l'information contenue dans les canaux bruts.

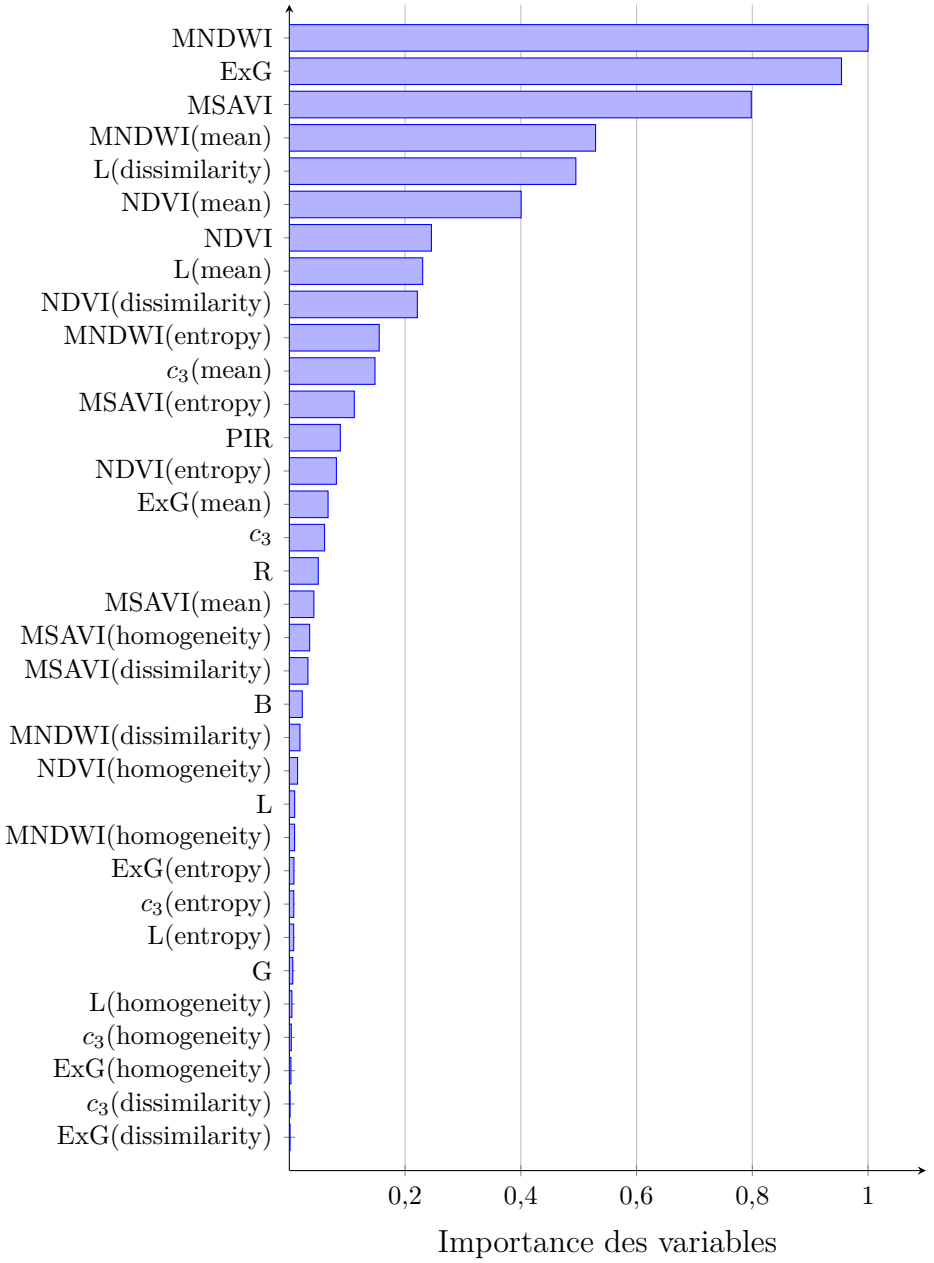


FIGURE 3.10 – Importance des différentes variables dans l'apprentissage d'un modèle XGBoost pour une tâche de classification de la couverture du sol sur le jeu de données *All*.

De plus, il est à noter que la suppression de trop de variables dans le modèle XGboost, détériore les performances du modèle. Au contraire, la sélection du bon nombre de variables permet au modèle d'être plus robuste (meilleures performances au niveau de la PU et de la PP), mais permet aussi d'améliorer la vitesse d'apprentissage, car le modèle à moins de variables avec lesquels choisir. Enfin, la figure 3.10 montre que les variables les plus corrélées aux données de couverture des sols sont les néocanaux et leurs variantes avec les textures.

XGBoost et la classification de l'usage des sols

De nouveau pour la tâche de classification de l'usage des sols, le modèle XGboost utilisant toutes les variables est utilisé pour dresser la liste des variables d'importances (Figure 3.11). La figure montre que les variables les plus importantes pour le modèle sont les néocanaux et leurs variantes de texture, excepté pour l'utilisation du canal PIR en toute première position. Les résultats des différents jeux de données sont présentés dans le tableau 3.7.

TABLEAU 3.7 – Résultats de la méthode XGboost pour la tâche de classification de l'usage du sol.

Jeux de données	**PG**	**PU**	**PP**	**F1s**
RVBPIR	**53,21**	39,19	36,51	37,8
RVBPIR+NEO	52,24	39,89	37,12	38,45
NEO	52,22	40,45	37,77	39,06
All	44,82	36,52	37,85	37,17
Opti_6	51,56	**42,61**	**38,27**	**40,32**
Opti_{10}	46,15	38,22	37,32	37,76

Ici, en ne prenant en compte que la PG, l'utilisation des canaux bruts pour le modèle semble être la solution la plus optimale. Or en prenant un compromis entre la PU et la PP, c'est-à-dire en prenant le F1s, c'est le modèle $Opti_6$ qui présente les meilleures performances. Il est cependant à noter les faibles performances globales du modèle XGBoost pour la tâche de classification de l'usage des sols, avec un maximum de $53,21\,\%$ pour la PG et $40,32\,\%$ pour le F1s.

Comme pour la classification de la couverture du sol, l'addition de néocanaux n'améliore pas les résultats, même si ces canaux semblent être les variables les plus importantes. Par contre, l'utilisation d'un nombre restreint de variables semble être particulièrement important, générant une différence d'environ $3\,\%$ dans le F1-score.

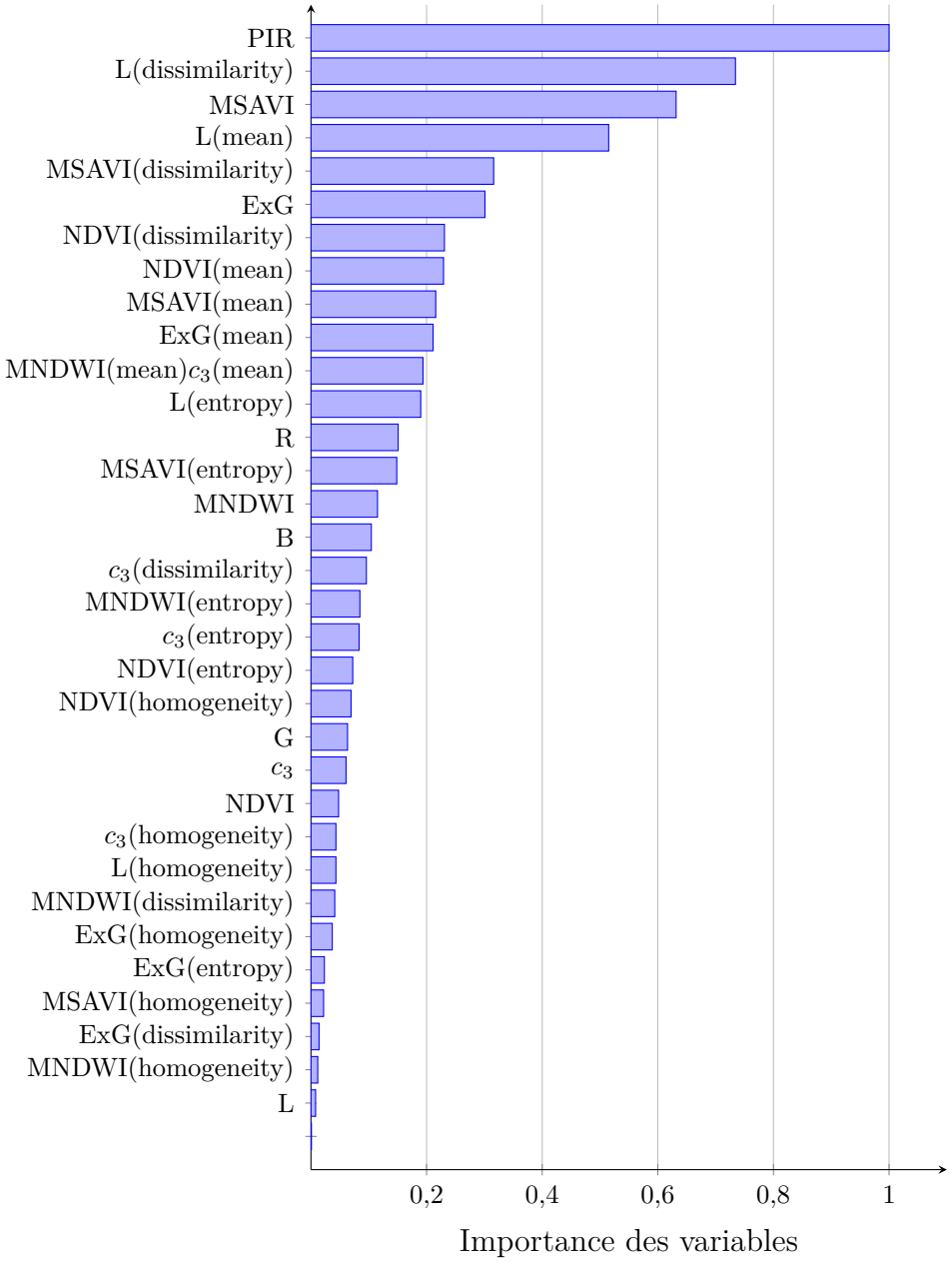


FIGURE 3.11 – Importance des différentes variables dans l'apprentissage d'un modèle XGBoost pour une tâche de classification de l'usage du sol sur le jeu de données *All*.

L'importance des néocanaux et des textures dans XGboost

Les figures 3.12 et 3.13 montrent l'importance des variables dans l'apprentissage du modèle XGBoost avec un jeu de données contenant uniquement les canaux bruts et les néocanaux. Il est observé que les néocanaux, pour les deux cas, sont les deux variables les plus importantes. Plus particulièrement, la variable ExG (« Excess Green Index ») ressort comme particulièrement importante pour les deux types de classification, mais aussi dans les variables importantes des modèles précédents (figure 3.10 et 3.11).

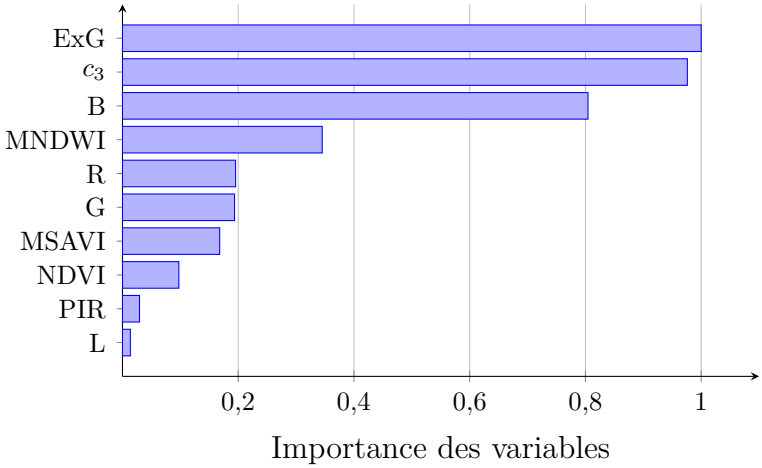


FIGURE 3.12 – Importance des différentes variables dans l'apprentissage d'un modèle XGBoost pour une tâche de classification de la couverture du sol sur le jeu de données *RVBPIRNEO*.

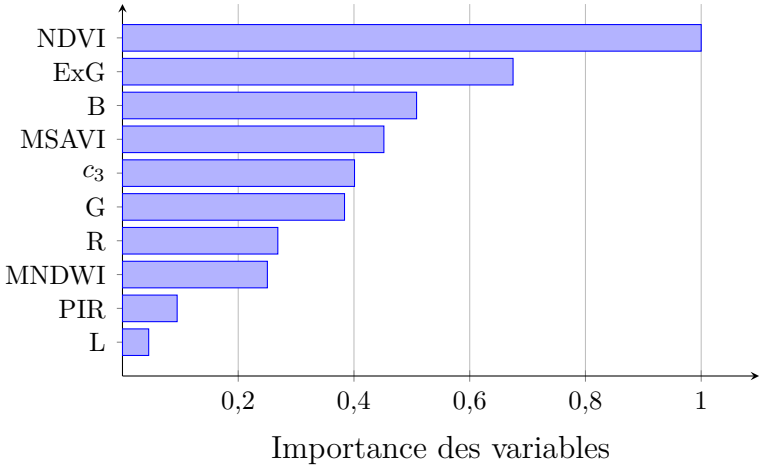


FIGURE 3.13 – Importance des différentes variables dans l'apprentissage d'un modèle XGBoost pour une tâche de classification de l'usage du sol sur le jeu de données *RVBPIRNEO*.

In fine, il est difficile de déterminer quelle combinaison de variables est la plus pertinente dans le cadre de la détection de la couverture et de l'usage du sol par méthode Xgboost. Le canal brut représentant le Proche Infrarouge, par exemple, est la variable la plus importante pour la tâche de classification de l'usage des sols. Or dans les modèles n'utilisant que les canaux bruts et les néocanaux, elle est la moins discriminante du modèle. Une étude systématique des variables, pour le modèle XGBoost, doit donc être effectuée pour déterminer celles qui sont les plus discriminantes.

Enfin, même si XGBoost offre des performances convenables pour la classification de la couverture du sol (74, 6 % pour le F1s), ces performances chutent drastiquement ($\approx -35\,\%$) dans le cadre de la classification de l'usage du sol (40,32 % pour le F1s), de fait de la complexité de cette problématique.

3.4.2 Détection des modes d'occupation du sol

Les différentes architectures appliquées sont de deux types : les architectures faisant de la labellisation du pixel central (Section 2.4) et les architectures faisant de la labellisation sémantique (Section 2.5).

Classification de la couverture du sol

Les résultats des différentes architectures de Deep Learning pour la tâche de détection de la couverture du sol sont présenté Tableau 3.8. Les meilleures performances de chaque type d'architecture sont mises en valeur dans le tableau. Le meilleur résultat obtenu par le modèle XGBoost pour la tâche de classification de la couverture du sol est reporté dans ce tableau.

Tableau 3.8 – Résultats des modèles de Deep Learning et de XGBoost pour la tâche de classification de la couverture terrestre avec pour entrée le jeu de données *RVBPIR*.

Modèles	**PG**	**PP**	**PU**	**F1s**
XGboost	77,55	72,39	76,93	74,60
AlexNet	73,29	66,77	78,98	72,36
ResNet	79,13	**73,18**	82,55	77,58
DenseNet	**79,55**	73,10	**83,13**	**77,79**
SegNet	79,54	73,00	82,16	77,31
DeepLab	**81,41**	**75,15**	**84,37**	**79,49**
FCN	80,48	74,66	83,99	79,05

Toutes les architectures de Deep Learning, hormis l'architecture AlexNet, supplantent les résultats de XGBoost de 3 % aussi bien en PG qu'en F1s. La méthode la plus performante, DeepLab, augmente cette différence avec un écart de 5 %. Le modèle de référence XGboost présente cependant des performances comparables aux meilleures méthodes de Deep Learning sur la PP. Il est aussi observé que les architectures de labellisation sémantique (SegNet, DeepLab et FCN) dominent légèrement les architectures de labellisation du pixel central. Cependant DenseNet talonne les capacités de ces architectures.

La plupart des modèles ont montré une PU supérieure à la PP, c'est-à-dire que le rappel (pourcentage du total des résultats pertinents correctement classés) est plus élevé que la précision (pourcentage des résultats pertinents). En termes plus généraux, une PU importante signifierai que pour les pixels attribués à une classe par le modèle, il y a une très forte probabilité que ces pixels soient bel et bien correctement classifiés. Cependant si la PP est faible, cela veut dire que le modèle a oublié de nombreux pixels et les a faussement classé dans d'autres classes. À l'inverse, une PP élevée signifie qu'un très grand nombre de pixels de la classe a bel et bien été trouvé par le modèle, et un PU faible, que bon nombre de pixels classifiés dans cette classe n'appartiennent pas réellement à la classe. Par exemple, dans le cas extrême où le modèle classifierait la totalité des pixels dans une seule classe. La PU de cette classe serait de 100 % et la PP de 0 %.

En prenant le F1s, qui est un bon compromis entre la PP et la PU, toutes les méthodes testées atteignent un score compris entre 72 % (AlexNet /« Labellisation du pixel central ») et 79 % (Deeplab/« Labellisation Sémantique »).

Classification de l'usage des sols

Avec les mêmes canaux d'entrée que dans la section précédente, les mêmes modèles sont entraînés pour la détection de l'usage des sols. Pour réaliser cette tâche, les algorithmes doivent ici différencier 12 classes, comme décrit dans la nomenclature du tableau **??**. En outre, pour la classification de l'usage de sols, le contexte et le voisinage des pixels deviennent très importants, car un pixel ayant la même radiométrie peut avoir des utilisations différentes. Les résultats sont présentés dans le tableau 3.9.

Comme pour la tâche de détection de la couverture des sols, toutes les architectures de Deep Learning hormis AlexNet, surpassent le modèle XGBoost avec des écarts de 10 % pour les architectures les plus performantes. Cette fois-ci, les archi-

TABLEAU 3.9 – Résultats des modèles de Deep Learning et de XGBoost pour la tâche de classification de l'usage du sol avec pour entrée le jeu de données *RVBPIR*.

Modèles	**PG**	**PP**	**PU**	**F1s**
XGboost	51,56	42,61	38,27	40,32
AlexNet	45,79	33,93	38,26	35,97
ResNet	55,96	43,89	49,58	46,56
DenseNet	**59,59**	**46,18**	**55,00**	**50,21**
SegNet	58,36	37,16	40,48	38,75
DeepLab	**61,45**	**49,77**	**51,04**	**50,40**
FCN	56,07	49,59	47,22	48,38

tectures DeepLab et DenseNet, en prenant en considération la plupart des indices et surtout le F1s, dominent les autres architectures. Cela est surtout visible avec l'architecture SegNet, qui bien que présentant une bonne PG, a un très mauvais F1s. D'où l'importance dans les résultats de ne pas se fier à un seul indice.

Dans la globalité, un net gain au niveau des performances de la classification de l'usage des sols est observé avec l'utilisation des architectures de Deep Learning. Cependant ces performances, $50,40\,\%$ au F1s pour la meilleure architecture, restent faibles. Une vue plus en détails de la matrice de confusion et du classement de chaque classe est nécessaire pour déterminer si les modèles proposés sont fiables. Cette étude sera menée plus loin dans ce chapitre.

3.4.3 Influence des entrées sur l'apprentissage

Les deux architectures les plus performantes sont conservées pour une étude de l'influence du jeu de données sur l'apprentissage : DenseNet pour les architectures labellisant le pixel central et DeepLab pour les architectures de labellisation sémantique.

Influence des néocanaux sur le Deep Learning

Les résultats pour la tâche de classification de la couverture du sol sont réunis dans le tableau 3.10, et les résultats pour la tâche de classification de l'usage du sol sont réunis dans le tableau 3.11. Les deux tableaux sont découpés en trois parties. La première partie comporte les résultats obtenus pour le jeu de données *RVBPIR* présentés dans les tableaux 3.8 et 3.9 ; les résultats obtenus par l'ajout de néocanaux à la donnée brute (*RVBPIR+NEO*) ; et les résultats en augmentant artificiellement

les données de *RVBPIR* : *RVBPIRAug*. Cette augmentation de données est effectuée en altérant aléatoirement la teinte, la saturation et la luminosité des images, augmentant par 4 la taille du jeu de données. La deuxième partie des tableaux présente les résultats obtenus par les deux architectures en ajoutant un néocanal en plus de la donnée brute. Cette partie permet d'observer plus en détail l'influence de chaque néocanal sur l'apprentissage. Enfin la troisième partie introduit des combinaisons des canaux bruts et des néocanaux.

TABLEAU 3.10 – Résultats des architectures DenseNet et DeepLab pour la tâche de classification de la couverture du sol sur des combinaisons de canaux bruts et de néocanaux.

	DenseNet				**DeepLab**			
Jeux de données	**PG**	**PP**	**PU**	**F1s**	**PG**	**PP**	**PU**	**F1s**
RVBPIR	80,07	73,58	83,54	78,24	81,32	74,86	84,43	79,36
RVBPIR+NEO	79,64	72,74	82,97	77,52	80,98	74,82	84,32	79,29
RVBPIRAug	76,09	69,57	77,38	73,27	78,94	73,12	80,53	76,65
RVBPIR+L	80,66	73,57	84,30	**78,57**	80,72	74,74	83,79	79,01
RVBPIR+c_3	79,46	72,81	82,57	77,38	80,19	74,13	83,22	78,41
RVBPIR+ExG	78,37	72,24	81,84	76,74	80,44	74,84	83,83	79,08
RVBPIR+NDVI	79,81	72,93	83,62	77,91	80,24	74,44	83,75	78,82
RVBPIR+MSAVI	79,08	72,81	82,25	77,24	80,42	74,5	83,58	78,78
RVBPIR+MNDWI	80,34	73,06	83,79	78,06	80,96	74,93	84,24	79,31
NEO	79,85	72,29	83,37	77,44	80,38	73,91	83,96	78,62
RVBPIR+L+c_3	80,12	73,2	83,64	78,07	81,32	75,27	84,36	**79,56**
RVBPIR+c_3+ExG	80,08	73,41	83,31	78,05	80,8	74,71	84,38	79,25
RVBPIR+L+ExG	79,5	73,2	82,77	77,69	80,81	74,9	83,82	79,11
RVBPIR+L+c_3+ExG	79,45	72,38	82,79	77,24	80,7	75,01	84,26	**79,37**
L+c_3+ExG	75,94	67,21	73,81	70,36	75,96	69,06	72,73	70,85

Comme attendu, du fait de la capacité des architectures d'apprentissage profond à élaborer ses propres combinaisons de canaux, y compris en tenant compte des voisinages des pixels considérés, il est constaté que l'ajout de néocanaux n'apporte pas de différence significative. Plus surprenant, il en va de même pour l'augmentation artificielle des jeux de données, qui semble perturber le modèle, particulièrement pour l'architecture DenseNet, alors que dans le domaine du traitement d'image, cette technique permet d'améliorer en général les performances [86]. Que ce soit par l'ajout ou le remplacement de certaines variables par des néocanaux, les architectures de Deep Learning, à l'inverse d'XGBoost, ne présente pas d'amélioration significative.

TABLEAU 3.11 – Résultats des architectures DenseNet et DeepLab pour la tâche de classification de l'usage du sol sur des combinaisons de canaux bruts et de néocanaux.

	DenseNet				DeepLab			
Jeux de données	**PG**	**PP**	**PU**	**F1s**	**PG**	**PP**	**PU**	**F1s**
RVBPIR	59,38	45,54	54,76	49,73	60,9	50,04	51,23	50,63
RVBPIRNEO	59,48	46,67	53,91	**50,03**	59,83	47,55	50,08	48,78
RVBPIRAug	52,85	43,00	48,68	45,66	56,65	48,25	48,92	48,58
RVBPIR+L	59,02	47,00	53,35	**49,97**	60,94	49,32	51,54	50,41
RVBPIR+c_3	61,78	48,22	55,14	**51,45**	60,32	48,51	51,1	49,77
RVBPIR+ExG	59,65	45,77	51,55	48,49	59,93	48,03	49,95	48,97
RVBPIR+NDVI	59,76	46,62	54,98	**50,46**	59,3	48,89	50,67	49,76
RVBPIR+MSAVI	57,78	46,1	53,92	49,70	59,54	49,25	52,51	**50,83**
RVBPIR+MNDWI	58,28	45,64	52,9	49,00	57,73	49,36	51,47	50,39
NEO	57,03	45,26	52,21	48,49	57,53	46,04	51,74	48,72
RVBPIR+L+c_3	57,99	46,84	52,86	49,67	61,51	50,25	52,44	**51,32**
RVBPIR+c_3+ExG	60,89	47,28	55,71	**51,15**	60,77	47,31	49,81	48,53
RVBPIR+L+ExG	59,52	46,72	53,8	**50,01**	60,86	47,63	50,49	49,02
RVBPIR+L+c_3+ExG	57,68	46,35	49,93	48,07	61,56	48,58	50,12	49,34
L+c_3+ExG	53,08	41,28	45,19	43,15	54	42,83	45,39	44,07

Sur le F1-score, une légère amélioration des résultats est observable dans les tableaux 3.10 et 3.11 (en gras). Cette amélioration n'est cependant pas significative, mais nous constatons que ces légères augmentations de performances sont quasi systématiquement liées aux néocanaux L et c_3, les seuls néocanaux non linéaires utilisés. On peut supposer qu'il est très aisé pour une architecture de Deep Learning de retrouver les néocanaux linéaires classiques à l'aide de ces filtres, mais plus difficile de retrouver « tôt » dans la partie encodeur ceux non linéaires.

Influence de l'ajout de la couverture du sol

Les architectures les plus performantes sont sélectionnées et utilisées pour une tâche de détection de l'usage du sol en utilisant la couverture terrestre comme entrée en plus des canaux bruts. Deux variantes sont utilisées : le *LCM* (Land Cover Model) qui utilise les classifications de la couverture terrestre d'une architecture de Deep Learning ; et le *LCE* (Land Cover Expert) qui utilise les valeurs de vérité de terrain, c'est-à-dire que cette version sert de limite supérieure de la performance de la classification de l'usage du sol si un « oracle » parfait pour la couverture du sol était disponible. Les résultats sont présentés dans le tableau 3.12, où les résultats précédents pour le *RVBPIR* des deux architectures ont été aussi rapportés.

TABLEAU 3.12 – Détection de l'usage des sols par les architectures DenseNet et DeepLab en ajoutant la couverture terrestre en plus des données brutes en entrée.

	DenseNet				DeepLab			
Jeux de données	**PG**	**PP**	**PU**	**F1s**	**PG**	**PP**	**PU**	**F1s**
RVBNIR	59,59	46,18	55,00	50,21	61,45	49,77	51,04	50,40
RVBNIR+LCE	74,90	58,55	61,36	59,92	73,56	56,09	61,06	58,47
RVBNIR+LCM	62,08	50,31	53,86	**52,02**	63,61	50,74	51,36	**51,05**

En raison de la proximité entre certaines classes de la couverture terrestre et de l'usage des sols, la forêt par exemple, le *LCE* atteint des résultats proches de 75 % de la PG pour les deux architectures. Notez qu'il s'agit d'une limite supérieure de la quantité d'informations que peut extraite le modèle de la couverture terrestre *connu*, ce qui laisse entrevoir la complexité du problème et les limites de la dérivation de l'usage du sol uniquement à partir de données de télédétection.

Plus intéressant encore, une amélioration de la classification de l'usage du sol est observée lorsque des étiquettes provenant de la prédiction de la couverture terrestre d'une architecture sont introduites. Dans le meilleur des cas, le fait d'ajouter *LCM* aux canaux bruts permet de faire un bon de 2 % dans les modèles, que ce soit pour la PG ou le F1s. Une classification en plusieurs niveaux semble alors à ce stade une évolution possible de classification de l'usage des sols. En reprenant la nomenclature initiale de l'usage des sols (tableau 3.2), cela équivaudrait à effectuer une première classification au niveau N1 avant de descendre au niveau N2 de la nomenclature.

3.5 Discussions et conclusions

Cette section présente les matrices de confusion de la couverture et de l'usage des sols de la meilleure architecture de Deep Learning. Une discussion autour de ces matrices et des résultats visuels est menée avant de conclure cette étude.

3.5.1 Détails autour des résultats de la couverture du sol

Pour rappel, toutes les techniques de Deep Learning citées précédemment sont utilisées en l'état, c'est-à-dire en ne modifiant pas les paramètres internes des architectures : la taille des convolutions, le nombre de couches convolutionnels etc. Seules les variables d'apprentissage générales (taux d'apprentissage, momentum, etc.) sont paramétrées pour chacune des architectures. La figure 3.14 représente l'application du modèle le plus performant (DeepLab) dans la tâche de classification de la couverture terrestre sur la commune de Thio. Le jeu de données utilisé contient uniquement les canaux bruts. La matrice de confusion issue de cette figure est présentée ci-dessous, Tableau 3.13.

Tableau 3.13 – Matrice de confusion montrant la précision producteur de la couverture du sol produite par le modèle DeepLab à partir du jeu de données *RVBPIR*.

	Prédictions				
	Bâtiments	Sols nus	Forêts	Herbes et arbustes	Surfaces d'eau
Bâtiments	43	0	0	0	0
Sols nus	24	84	2	4	7
Forêts	18	2	78	12	5
Herbes et arbustes	15	14	19	84	1
Surfaces d'eau	0	0	1	0	87

Dans ce tableau, une précision globale de $81,32\,\%$ est atteinte et ce, principalement grâce aux scores obtenus pour la classification des surfaces en eau, des sols nus et de la végétation. La reconnaissance de la classe de bâtiment est la tâche la plus difficile pour le modèle ($43\,\%$). Le modèle a tendance à surestimer l'étendue des bâtiments, créant de nombreux faux positifs sur les autres classes, notamment les sols nus. Ce phénomène peut s'expliquer par la proximité des autres classes autour de la classe représentant les bâtiments, comme on peut le constater sur la Figure 3.15 par un « débordement » des pixels des bâtiments tout autour.

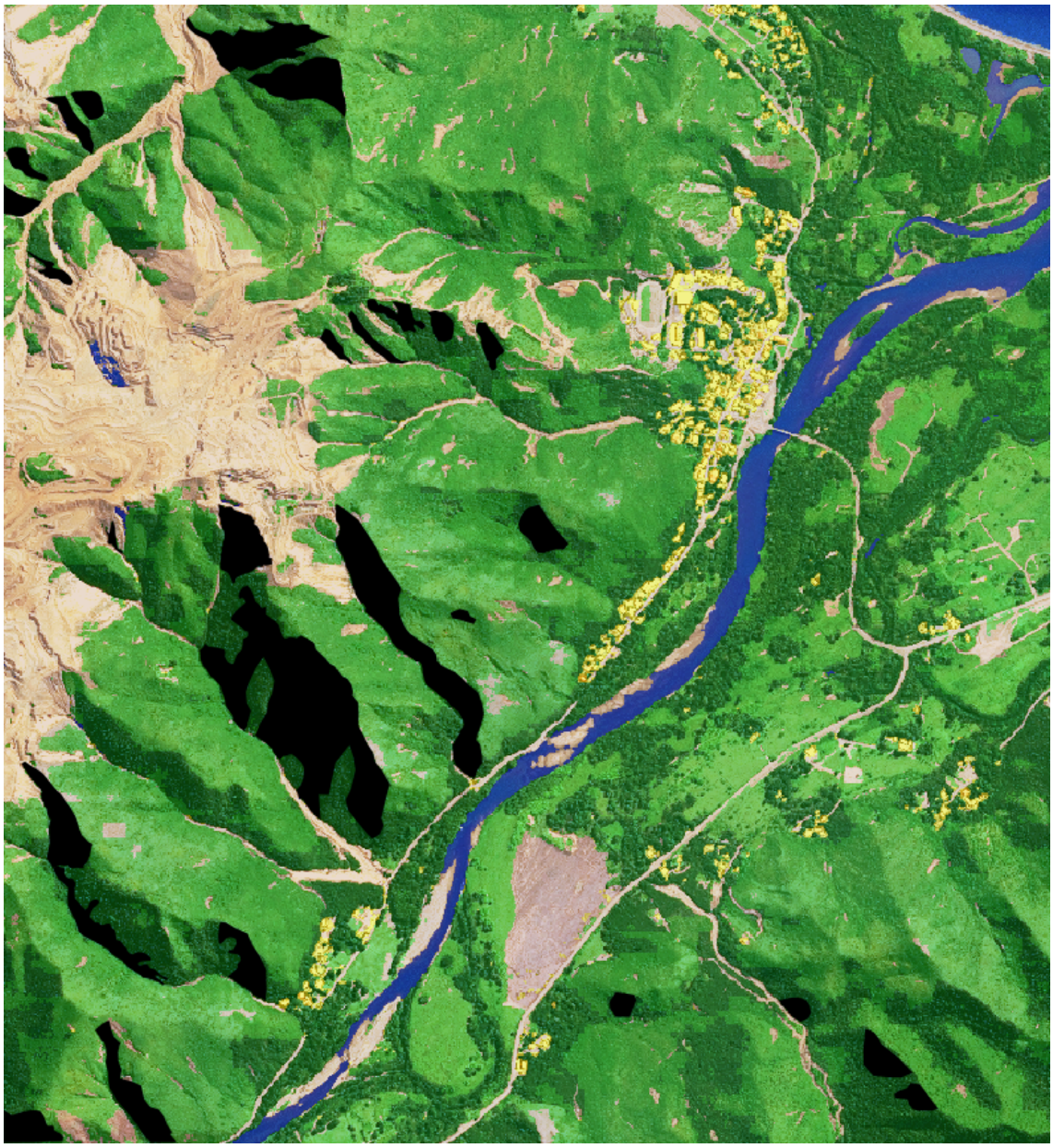


FIGURE 3.14 – Résultats de l'application de l'architecture DeepLab sur l'image SPOT 6 de la commune de Thio pour la tâche de classification de la couverture du sol dont la nomenclature est présentée Tableau 3.1.

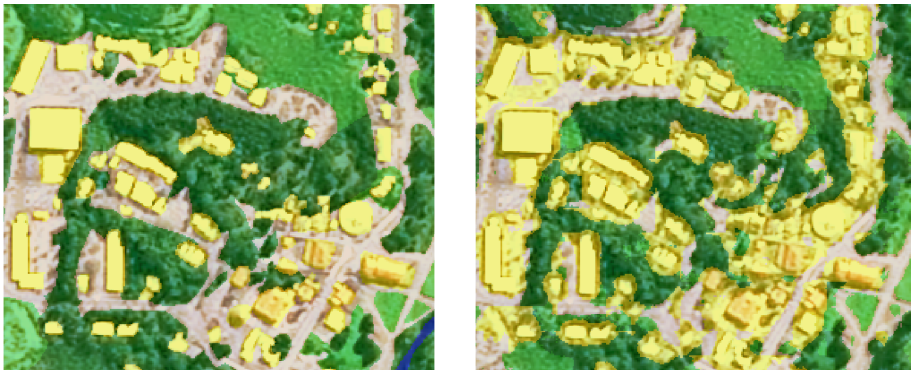


FIGURE 3.15 – Illustration du débordement de la détection de bâtiments sur un fond d'image SPOT 6 de la commune de Thio. À gauche la vérité terrain et à droite la classification de l'architecture DeepLab.

Il est observé aussi une légère confusion entre les sols nus et la classe représentant la végétation de faible densité, ainsi que pour cette dernière avec la forêt. Cette confusion peut s'expliquer par le fait que ces classes se côtoient dans l'espace et que la lisière entre ces classes est floue et entachée d'erreurs avec des frontières qui dépendent de l'appréciation de l'opérateur humain. Il n'existe pas en effet de règles précises pour définir cette frontière, ce qui entraîne une forte confusion entre les différentes densités de végétation, comme montrée Figure 3.16.

FIGURE 3.16 – Illustration de la confusion dans la classification de la couverture du sol entre les végétations de différentes densités. À gauche l'image satellite SPOT 6, au milieu la vérité terrain et à droite la classification de DeepLab.

Dans cette figure, la rivière aussi est très mal détectée par le modèle du fait de la couverture de la forêt juste au-dessus d'elle. Les autres confusions qui existent entre les surfaces d'eau et le sol nu vient du fait de la présence des berges autour des rivières, de l'engravement au fond des rivières, et des rivières asséchées.

Il est difficile de déterminer la réponse correcte entre la classification rigoureuse pixel par pixel d'une méthode automatique, et le trait de délimitation créé subjectivement par un opérateur humain. Par exemple toute une zone classifiée par l'opé-

rateur humain comme étant une zone de végétation de faible densité Figure 3.17, sera classifiée scrupuleusement par un modèle plus précis comme le Deep Learning.

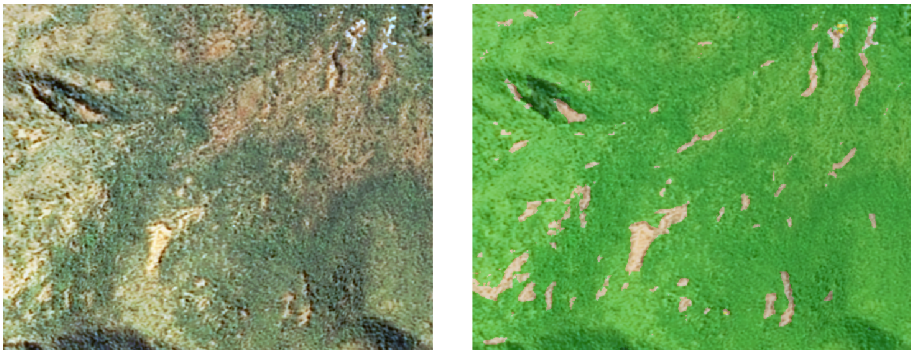


FIGURE 3.17 – Illustration de la précision des modèles de machine learning par rapport à l'étiquetage plus grossier d'un opérateur humain qui aurait classé l'intégralité de l'image dans une seule classe. À gauche l'image satellite et à droite la classification du modèle.

3.5.2 Détails autour des résultats de l'usage des sols

La figure 3.18 représente l'application du modèle le plus performant (DeepLab) dans la tâche de classification de la couverture terrestre sur la commune de Thio. Cela correspond à l'architecture Deep Learning avec le jeu de données contenant uniquement les canaux bruts. La matrice de confusion issue de cette figure est présentée ci-dessous, Tableau 3.14.

Dans cette matrice de confusion, 4 classes sont particulièrement bien détectées par le modèle : les zones urbaines, les zones minières, les forêts et les surfaces d'eau. La zone industrielle est en partie confondue avec la zone urbaine et une partie du réseau routier. Cette zone est en effet souvent constituée de parkings (confondus avec le réseau routier) et des bâtiments qui s'ils sont de trop petites tailles peuvent être considérés comme appartenant à une zone urbanisée. Une grande confusion est également constatée entre les forêts et les zones arbustives. Cette dernière est aussi souvent confondue avec les zones herbacées, continuation du problème de la détection de la couverture du sol précédente. L'ensemble de ces erreurs sont visibles Figure 3.19.

Les roches nues, les sols nus et les pistes sont les moins bien détectés par le modèle, très souvent confondus avec la zone minière qui est un produit de toutes ces classes. Cette erreur peut s'expliquer par la taille de la fenêtre coulissante utilisée par le modèle. Il existe plusieurs études qui discutent de l'influence de la taille de l'image sur la performance du Deep Learning [157, 158]. Pour une résolution de 1,5 mètre

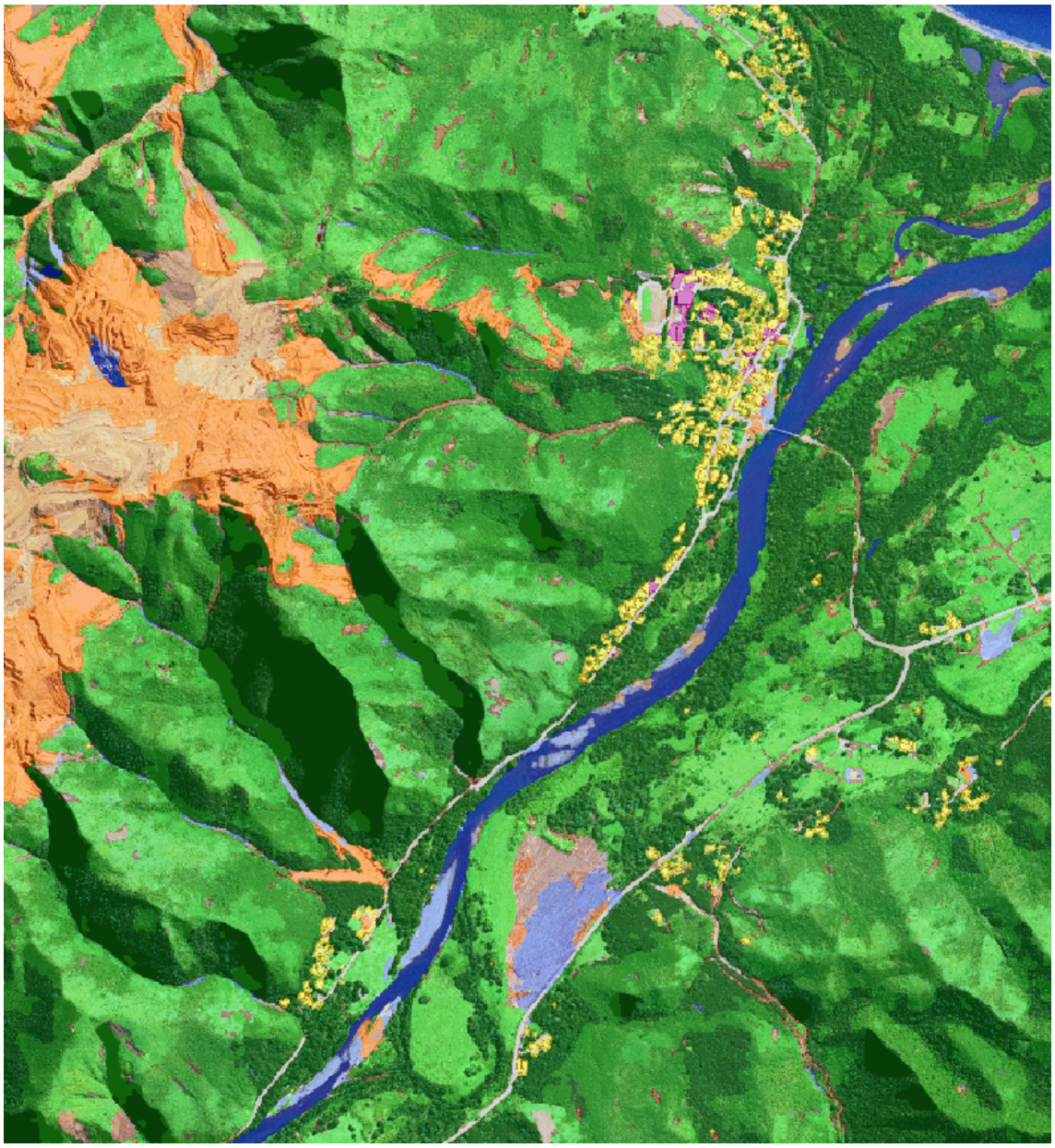


FIGURE 3.18 – Résultat de l'application de l'architecture DeepLab sur l'image SPOT 6 de la commune de Thio pour la tâche de classification de l'usage du sol dont la nomenclature est présentée Tableau 3.2 ;

TABLEAU 3.14 – Matrice de confusion de l'usage du sol produite par le modèle DeepLab à partir du jeu de données *RVBPIR + LCM*.

	Predictions											
	Zones urbanisées	Zones indust.	Zones minières	Réseaux routier	Pistes	Forêts	Arbustes	Herbes	Roches nues	Sols nus	Eau	Engravements
Zones urabinées	72	32	0	9	1	2	1	0	1	0	0	0
Zones indust.	3	43	0	0	0	0	0	0	0	0	0	0
Zones minières	0	0	86	2	15	0	1	2	27	69	2	5
Réseaux routiers	5	18	1	58	7	0	0	0	0	2	0	4
Pistes	1	1	2	3	8	0	0	1	2	1	0	1
Forêts	6	3	0	6	7	77	19	2	5	1	4	3
Arbustes	9	2	3	7	25	17	69	46	27	3	1	5
Herbes	4	1	3	4	16	2	8	47	9	6	0	3
Roches nues	0	0	0	1	4	0	1	1	12	0	0	1
Sols nus	0	0	2	5	4	0	0	1	2	14	0	4
Eau surfaces	0	0	0	0	0	1	0	0	1	0	88	1
Engravements	0	0	3	5	13	1	1	0	14	4	5	37

et pour une image de 64×64 pixels, la surface couverte est d'environ $0,01\,km^2$. À cette résolution et pour la surface étudiée, il est possible pour un opérateur humain d'identifier le type de terrain et son utilisation. Cependant il est difficile de déterminer précisément la classe d'appartenance d'un objet si celui-ci fait partie d'un objet de plus grande surface. Par exemple, une zone de sol nu peut être confondue avec une zone minière. Il y a donc un impact direct de la taille d'une fenêtre, lié à la perception globale qu'un opérateur peut avoir d'une zone géographique, sur la labellisation d'un point en particulier.

Des tailles alternatives de 32×32 pixels ou 128×128 pixels ont été étudiées, mais peu de variations dans le résultat ont été observées. En raison des limitations matérielles de la carte graphique, impliquant une réduction drastique de la taille du lot (*batch-size*), nous n'avons pas poursuivi la configuration de 128×128 pixels. Le nombre plus important d'échantillons individuels à la taille 32×32 ne semble pas entraîner d'amélioration. Nous avons donc opté pour des pixels de 64×64, en trouvant un compromis entre la taille de l'image et le nombre d'échantillons.

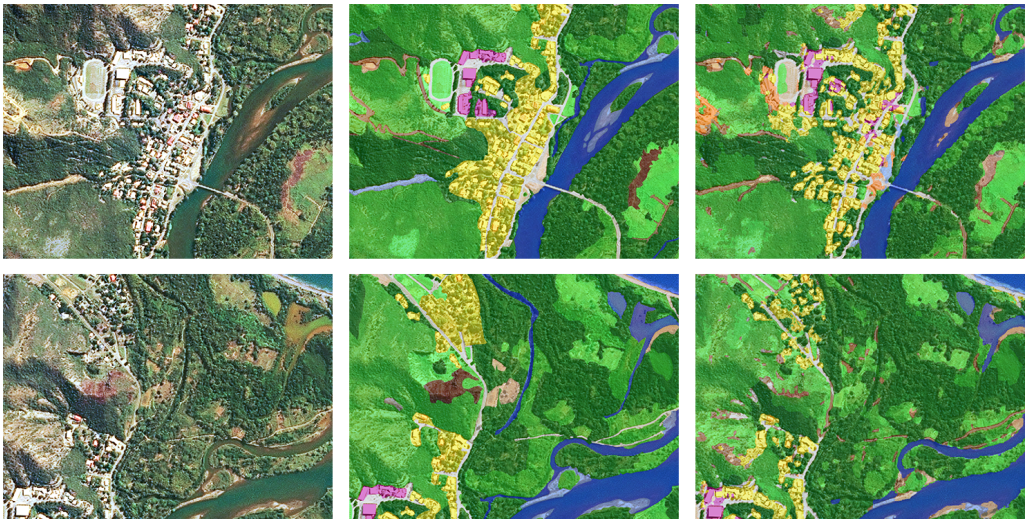


FIGURE 3.19 – Résultats de l'architecture DeepLab sur la commune de Thio pour la tâche d'usage des sols. À gauche l'image satellite, au milieu l'étiquetage manuel de l'usage du sol et à droite l'application de l'architecture de Deep Learning.

3.5.3 Conclusions

Dans cette étude, des techniques de machine learning telles que le Deep Learning et XGBoost ont été comparées pour la classification de la couverture et de l'usage du sol dans un environnement insulaire tropical. À cette fin, un ensemble de données spécifique basé sur les données du satellite SPOT 6 a été créé et mis à la disposition de la communauté scientifique : cinq zones représentatives de la Nouvelle-Calédonie labellisées manuellement, quatre à utiliser comme ensemble d'apprentissage et la cinquième comme ensemble de tests. Les performances de XGBoost sont à la hauteur de celles du Deep Learning pour les classifications de la couverture du sol, mais comme dans de nombreuses applications dans le domaine du traitement d'images, les meilleures architectures de Deep Learning affichent les meilleures performances. L'approche XGBoost est clairement moins performante dans des domaines d'utilisation plus complexes qui nécessitent un niveau plus élevé de conceptualisation de l'environnement, comme la classification de l'usage de sol.

Aucune des 4 zones d'entraînement ne contient toutes les classes de la nomenclature pour la couverture et l'usage des sols. Seule la zone de Thio (Tableau 3.3) comprend toutes les étiquettes possibles. Il existe de grandes inégalités dans la répartition des classes, la classe de végétation couvrant plus de 90 % du territoire calédonien. L'ensemble de la zone de test a été utilisée dans l'évaluation des résultats de classification, sans aucun équilibrage des classes, afin de correspondre à une tâche de cartographie réaliste et complète. En effet, les réseaux neuronaux fonctionnent

moins bien lorsqu'ils sont formés avec des ensembles de données déséquilibrés [159]. Dans cette étude, certaines classes de la couverture et de l'usage des sols étaient sur-représentées. En conséquence, les architectures testées n'ont pas pu apprendre à différencier correctement chaque classe et une solution pour surmonter le problème réside en un sous-échantillonnage permettant d'équilibrer les ensembles de données [159].

En ce qui concerne spécifiquement l'approche du Deep Learning, les deux méthodes : la labellisation sémantique et la labellisation au pixel fournissent des performances équivalentes pour les architectures les plus efficaces : DeepNet et DeepLab, dont la structure interne n'a pas été modifiée. Si les différences sont plutôt faibles pour la classification de la couverture terrestre, des gains importants peuvent être obtenus pour l'usage des sols en choisissant la bonne architecture. La nature plus difficile de la classification de l'usage du sol est confirmée par les faibles performances de XGBoost et AlexNet. En revanche, ces méthodes, qui sont rapides et faciles à mettre en œuvre et à entraîner, ont atteint des performances (relatives) satisfaisantes pour la tâche de classification de la couverture terrestre plus simple.

En prenant en compte les capacités de calcul disponibles, il pourrait être intéressant d'utiliser une approche de labellisation sémantique pour une application à large échelle du modèle à des images de télédétection à très haute résolution. À l'inverse si on constate un manque de données, l'approche de la classification par pixel est appropriée, car il est plus aisé d'échantillonner des points sur une image par photo-interprétation que de la segmenter manuellement.

Comme prévu, les architectures CNN construisent leur propre ensemble de caractéristiques et ne nécessitent pas d'informations supplémentaires en entrée autre que les canaux bruts des images satellites, même si ce résultat peut être remis en question lorsqu'on y ajoute des néocanaux fortement non linéaires tels que c_3 qui peuvent légèrement améliorer la précision. Pour notre ensemble de données, l'augmentation des données (rotation, décalage, modification de la teinte, du contraste et de la saturation) n'a pas amélioré les performances. Ces techniques, largement utilisées dans le domaine du traitement des images, pourraient néanmoins être utiles lorsque les données ne sont pas suffisamment représentatives. Nos conclusions ont également montré que l'ajout de la sortie de la couverture terrestre comme entrée pour la classification de l'usage des sols peut améliorer la précision, ce qui suggère qu'une approche hiérarchique pourrait être intéressante pour effectuer la tâche complète de cartographie de la couverture et de l'usage des sols.

La précision de la classification de la couverture terrestre est globalement élevée,

avec une précision globale de $81,41\,\%$ sur l'ensemble de tests du meilleur modèle. La classe « bâtiment » est la plus difficile à détecter, avec une grande confusion avec la classe « sol nu ». Une meilleure détection de cette classe nécessite des informations exogènes. La classification de l'usage des sols reste une tâche très difficile avec un score maximum de $63,61\,\%$ sur la base de test de la meilleure architecture de Deep Learning, avec une nette amélioration par rapport à XGBoost.

Les nomenclatures de l'utilisation des terres et de la couverture des terres sont assez similaires en raison des nombreuses zones non soumises à une utilisation humaine directe. Les principales différences se situent au niveau de la division du tissu urbain. Il est beaucoup plus complexe de qualifier ce type de zone et l'expertise humaine est souvent nécessaire, mais aussi subjective. D'autres types d'erreurs sont à noter en raison de la nature complexe des terres en Nouvelle-Calédonie. Les milieux sont variés et entre les côtes est et ouest de la Nouvelle-Calédonie, on constate des dynamiques environnementales profondément différentes. De plus, l'étude n'a été réalisée que sur la partie sud de la Nouvelle-Calédonie. Les dynamiques de la province Nord et de la province des Îles ont également leurs propres spécificités qui devront être prises en compte.

Dans les travaux futurs, nous visons donc à ajuster le modèle au reste de la Nouvelle-Calédonie, c'est-à-dire à la province Nord et la province des Îles. Même si une application du modèle pour la classification de la couverture terrestre (Figure 3.20) et la classification de l'usage des sols (Figure 3.21) est possible, elles sont entachées d'erreurs. Des informations complémentaires seront nécessaires pour couvrir les conditions spécifiques de ces nouvelles régions. En termes de zone cartographique, il peut être intéressant d'inclure également l'environnement maritime, en particulier les nombreux îlots et récifs du lagon de Nouvelle-Calédonie. Le chapitre suivant décrit une extension de ces approches en utilisant les données Sentinel-2 au lieu de SPOT 6. Bien que ce capteur ait une résolution spatiale inférieure, sa résolution spectrale est meilleure et il offre un temps de revisite de seulement 5 jours. Enfin une autre piste d'amélioration consisterait à inclure la détection des nuages et des ombres dans le classificateur et à utiliser des filtres de post-traitement [160] et des heuristiques (règles orientées objet...).

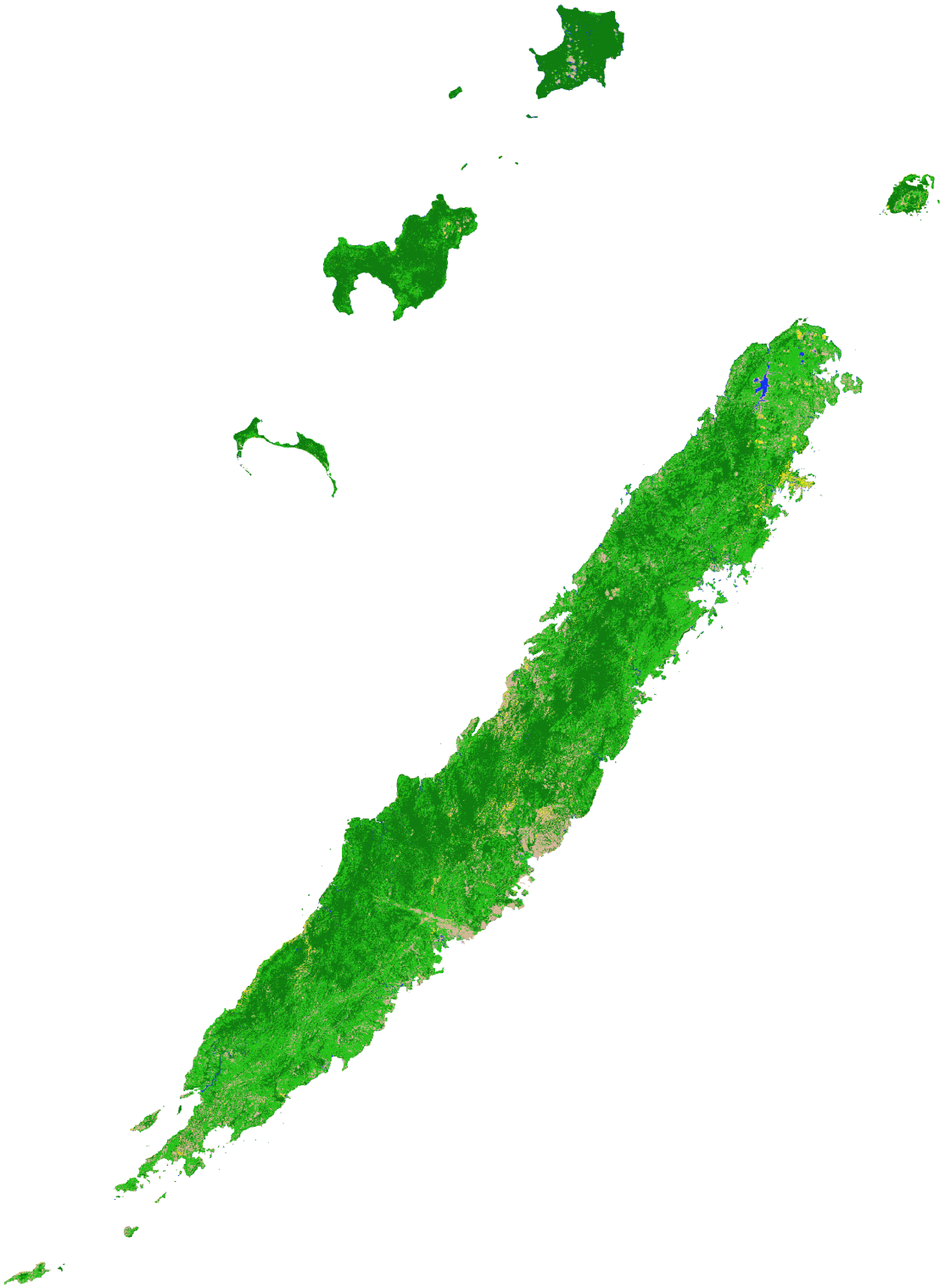


FIGURE 3.20 – Résultat de l'architecture DeepLab pour la tâche de classification de la couverture du sol appliquée sur l'intégralité du territoire de la Nouvelle-Calédonie.

FIGURE 3.21 – Résultat de l'architecture DeepLab pour la tâche de classification de l'usage des sols appliquée sur l'intégralité du territoire de la Nouvelle-Calédonie.

CHAPITRE 4

VERS UNE DÉTECTION DE CHANGEMENT DU SOL

"Quand on cherche, on trouve généralement quelque chose, mais ce n'est pas toujours exactement ce qu'on voulait."

J.R.R. Tolkien - Bilbo le Hobbit

Sommaire

4.1 Introduction 109
4.1.1 La détection de changement 109
4.1.2 Objectif général de l'étude 110
4.2 Adaptation et création de données 111
4.2.1 Les données du satellite Sentinel-2 111
4.2.2 Adaptation du MOS 112
4.2.3 Création de données d'apprentissage pour le modèle de Deep Learning 120
4.3 Analyse et discussion autour des résultats 123
4.3.1 Le MOS comparé à la réalité 123
4.3.2 Le Deep Learning appliqué au MOS 126
4.4 Conclusions 130

4.1 Introduction

La Nouvelle-Calédonie est une île tropicale qui possède une faune et une flore dont la diversité est exceptionnelle [161, 162]. Pour préserver ce patrimoine naturel, une surveillance régulière de son environnement est nécessaire. Cependant, en raison de l'étendue du territoire et de sa faible densité de population, il existe peu d'informations homogènes à l'échelle du territoire. Notamment, le relief montagneux et les nombreuses régions difficiles d'accès ne permettent pas une surveillance sur le terrain [163]. De plus, la couverture nuageuse importante dans le secteur du Pacifique Sud [164] réduit les possibilités d'une observation régulière par satellite ($\approx 12\,\%$ des données satellites sont exploitables).

La surveillance environnementale par satellite permet d'obtenir des informations à large échelle [165] lorsque l'accès aux données est fourni à un coût raisonnable et que la couverture satellitaire est suffisamment importante pour une surveillance régulière. Le satellite Sentinel-2, qui répond aux critères d'accessibilité et de couverture pour le territoire de la Nouvelle-Calédonie, peut être utilisé pour créer un indice environnemental très utile aux gestionnaires : la couverture du sol. Une mosaïque d'images satellites avec peu de nuages (moins de $10\,\%$) et un traitement fiable et précis pour déterminer la couverture du sol du territoire calédonien permettrait de mesurer, par exemple, le recul des forêts [166, 167] ou encore la progression des zones urbaines [168, 169].

4.1.1 La détection de changement

La détection de changement est une problématique importante dans le domaine de la télédétection. Elle permet de quantifier et de comprendre les changements d'état d'une surface ou d'un objet dans le temps [170]. Cette série d'observations se fait à partir de l'exploitation d'images satellites et/ou aéroportées qui ont été traitées pour répondre à la thématique souhaitée.

En fonction de la thématique et des moyens mis en œuvre, cette détection de changement peut se faire dans un temps de l'ordre du jour (ex. : surveillance météorologique), du mois (ex. : feux de forêt, inondations) ou de l'année (ex. : carte d'occupation du sol). Cette détection est bien entendu dépendante des outils d'observations utilisés, c'est-à-dire la résolution spatiale et spectrale des capteurs utilisés et le temps de revisite du capteur sur une même région d'observation.

De nombreuses études utilisent aujourd'hui des données satellites haute résolution pour faire de la détection de changement, tellement nombreuse qu'il est difficile parfois d'établir un cadre dans l'application stricte de méthodes de détection de changements [171]. Ces thématiques touchent aussi bien le suivi de l'évolution du tissu urbain [172], l'évolution météorologique [173], l'évolution de catastrophes naturelles [174], et le suivi environnemental plus généralement [44]. Ce suivi, qui ne s'inscrit pas à un instant précis, mais qui suit l'évolution d'un objet ou d'une surface, exprime de nouvelles informations et permet de mieux cerner la dynamique de l'objet d'étude. Cela permet de prédire et de contrôler plus efficacement les changements futurs et ainsi de donner aux décideurs des indices plus fiables pour leur prise de décision.

Aujourd'hui, avec les techniques de machine learning, et plus précisément grâce aux techniques de Deep Learning [175], une détection automatique de changement de l'occupation des sols devient possible. L'enjeu est important, créer de tels indices manuellement demande un travail d'expertise qui se compte en centaines d'heures de travail (dépendant de la métrique voulue) pour un territoire comme la Nouvelle-Calédonie. Pour une observation à grande échelle ou sur une durée très courte, des méthodes d'automatisation sont nécessaires pour réduire les coûts, néanmoins celles-ci doivent être suffisamment performantes pour proposer des résultats cohérents [176, 177].

4.1.2 Objectif général de l'étude

Comme pour l'étude précédemment menée sur les données du satellite SPOT 6, le Deep Learning est appliqué au problème de la couverture des sols de la province Sud de la Nouvelle-Calédonie, mais à une résolution spatiale plus faible et sur l'ensemble du territoire pour différents intervalles de temps. Dans cette étude, les données d'apprentissage seront les données du MOS de 2014 créées par l'Observatoire de l'Environnement de Nouvelle-Calédonie (OEIL). L'objectif étant de créer une occupation du sol sur 5 années consécutives, et d'en faire la comparaison dans le temps pour en extraire des statistiques simples. Cette étude a donc pour but d'évaluer la possibilité d'une telle production et de déterminer qu'elles seront les besoins supplémentaires pour donner de meilleurs indices pour un suivi environnemental à grande échelle.

4.2 Adaptation et création de données

Cette partie décrit les méthodes utilisées pour obtenir, à partir de données satellites brutes et une technique de machine learning, une cartographie de la couverture des sols. Pour cela, sont détaillés : les traitements appliqués aux données satellites pour arriver à des mosaïques annuelles ; la revue du MOS et de ces limites dans son adaptation aux problématiques de Deep Learning, qui a mené à créer une nouvelle nomenclature de la topologie ; et la mise en forme des données pour un apprentissage de l'architecture choisie.

4.2.1 Les données du satellite Sentinel-2

Le satellite Sentinel-2 fait partie du programme Copernicus. Conduit par l'Union européenne, ce projet a pour but d'établir une observation et une surveillance de la surface de la Terre. Les données brutes ainsi que des produits corrigés des biais atmosphériques sont mis en libre accès par le CNES.

Sentinel-2A et Sentinel-2B ont été mis en fonctionnement le 23 juin 2015 et le 6 mars 2017 respectivement. Les deux satellites forment la constellation Sentinel-2 qui documente la surface de la Terre pour une revisite d'un même lieu géographique de 5 jours. Ils disposent d'une fauchée au sol de $290\,km$ et des capteurs allant jusqu'à 13 bandes de résolution spectrale allant du visible au moyen infrarouge.

Des 13 canaux, 4 d'entre eux ont une résolution de $10\,m$. Il s'agit des bandes spectrales bleu ($0.490\,\mu m$), verte ($0.560\,\mu m$), rouge ($0.665\,\mu m$) et proche infrarouge ($0.842\,\mu m$). Les bandes spectrales destinées aux corrections atmosphériques (0.443, 0.945 et $1.375\,\mu m$) sont à une résolution de $60\,m$ et les 6 bandes restantes (0.705, 0.740, 0.783, 0.865, 1.610 et $2.190\,\mu m$) ont une résolution de $20\,m$.

Les images du niveau L2A de Sentinel-2 sont pré-sélectionnées manuellement selon une grille de la Nouvelle-Calédonie dans le but de créer une mosaïque annuelle. Le tableau B.2 présente le détail de chaque image utilisée. Malgré le nombre d'images disponibles, il n'est pas possible de présenter un produit mensuel à cause de la forte couverture nuageuse recouvrant l'intégralité du paysage, Figure 4.1. Même des images possédant une visibilité correcte ne sont pas exploitables à cause de la présence de fines particules de nuages brouillant le signal du capteur satellite (image de droite Figure 4.1). Pour s'assurer d'une bonne qualité de l'ensemble des données, ces images seront écartées.

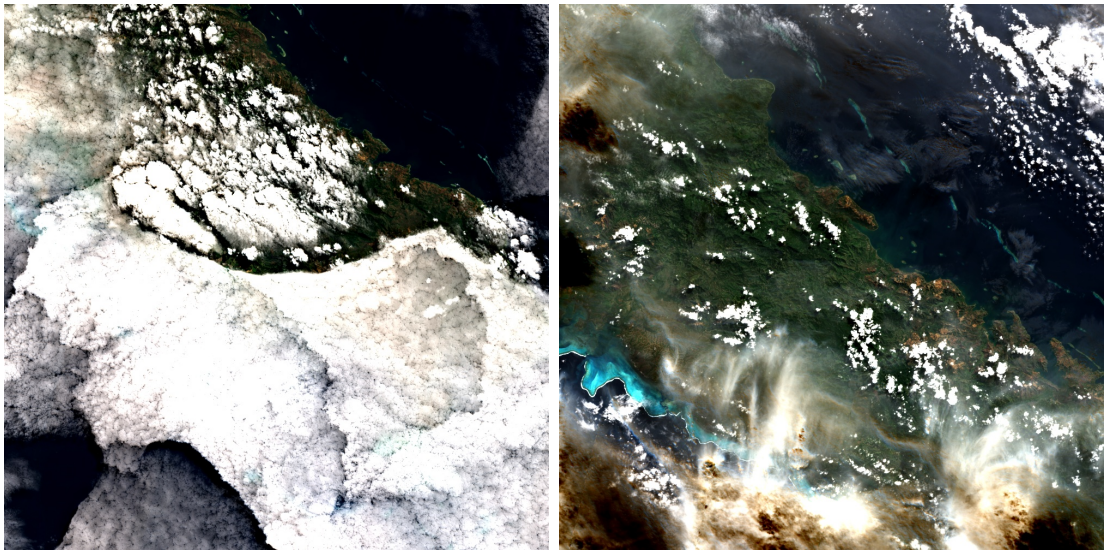


FIGURE 4.1 – Exemple de deux images satellites Sentinel-2 inutilisables

4.2.2 Adaptation du MOS

Avant toute utilisation des cartes d'occupation du sol créées en Nouvelle-Calédonie, il convient de comprendre leur objectif premier pour déterminer si elles sont exploitables dans l'optique d'un traitement automatique. Dans cette section, en plus de voir en détail la carte d'occupation du sol, une adaptation de celle-ci à travers une nouvelle nomenclature sera présentée. Enfin l'adaptation de ces données et leur mise en forme dans des jeux de données adaptés à l'entrée de modèles Deep Learning seront détaillées.

Les MOS de Nouvelle-Calédonie

Le Mode d'Occupation des Sols (MOS), utilisé comme base d'apprentissage des réseaux de neurones denses, représente une cartographie actualisée de l'occupation des sols pour un lieu donné. Cependant, ce nom peut porter à confusion avec les cartes d'occupation du sol de Nouvelle-Calédonie. Créées sous le même nom, « MOS », ces cartes désignent une carte d'usage des sols. Plusieurs de ces cartes ont été produites, mais avec des nomenclatures et des objectifs différents. Elles ne sont donc pas comparables dans le temps et ne peuvent donc pas être utilisées pour effectuer un suivi des « modes d'occupation du sol ».

Nous garderons cependant le nom de MOS pour désigner ces cartes d'usages du sol par convenance. Plusieurs de ces cartes ont été créées et mises à disposition sur le site de l'OEIL et son géoportail. Ces différents MOS sont les suivants :

- Une couverture sur l'ensemble de la Nouvelle-Calédonie avec des données SPOT 1, 2 et 3 de 1996 a été réalisée. Une dizaine de classes ont été identifiées pour la végétation ;
- Une couverture, sur l'ensemble de la Nouvelle-Calédonie avec des données SPOT 5 de 2008, a été réalisée. Les classes ont été ajustées par rapport au CORINE Land Cover pour une restitution au 1/25 000. L'objectif principal était de faire un état des lieux de la dégradation des sols induite par l'activité minière, cependant la carte créée n'est pas comparable au MOS de 1996 ;
- Un MOS sur l'ensemble de la province Sud est réalisé à partir de données satellites de 2010. Les classes s'inspirent du Corine Land Cover, mais réajustées au contexte de la Nouvelle-Calédonie. L'objectif principal est un suivi environnemental de la province Sud ;
- Un MOS sur l'ensemble de la province Nord, à partir de données RapidEye de 2013, a été réalisé. Le MOS a été réalisé avec les mêmes objectifs que le MOS de 2010, les mêmes classes ont donc utilisés entre ces deux projets. Le capteur de RapidEye propose 5 bandes spectrales à une résolution de $5\,m$: RVB, red-edge (bande à la limite du proche infrarouge) et proche infrarouge ;
- Un MOS sur l'ensemble de la Nouvelle-Calédonie avec des données SPOT 6 de 2014 a été réalisé (Figure 4.2). Les classes sont comparables à celles des précédents MOS avec un objectif identique, mais une nouvelle méthodologie définie par l'OEIL a été appliquée.

Création du MOS de 2014

Le MOS utilisé dans cette étude comme base d'apprentissage est le MOS de 2014 (Figure 4.2). Les classifications sont fournies directement par l'OEIL et les images SPOT 6 utilisées proviennent d'Airbus. Le MOS de 2014 a été produit avec pour objectif une précision géographique de 1ha, pour une échelle d'utilisation de 1/15 000 à 1/18 000 sur l'ensemble de la province Sud ($\sim 7000\,km^2$). Les données ont été orthorectifiées par rapport au Modèle Numérique de Terrain (MNT) du Gouvernement de la Nouvelle-Calédonie à 10 mètres de résolution.

En plus des données satellites, des données exogènes dans la création du MOS ont été employées. Des données sur les parcelles agricoles et la sylviculture de la province Sud ont été ajoutées au modèle. Des données sur des milieux naturels, comme la forêt sèche et la mangrove, ont été aussi ajoutées au modèle. Enfin la couche du réseau routier a été aussi ajoutée.

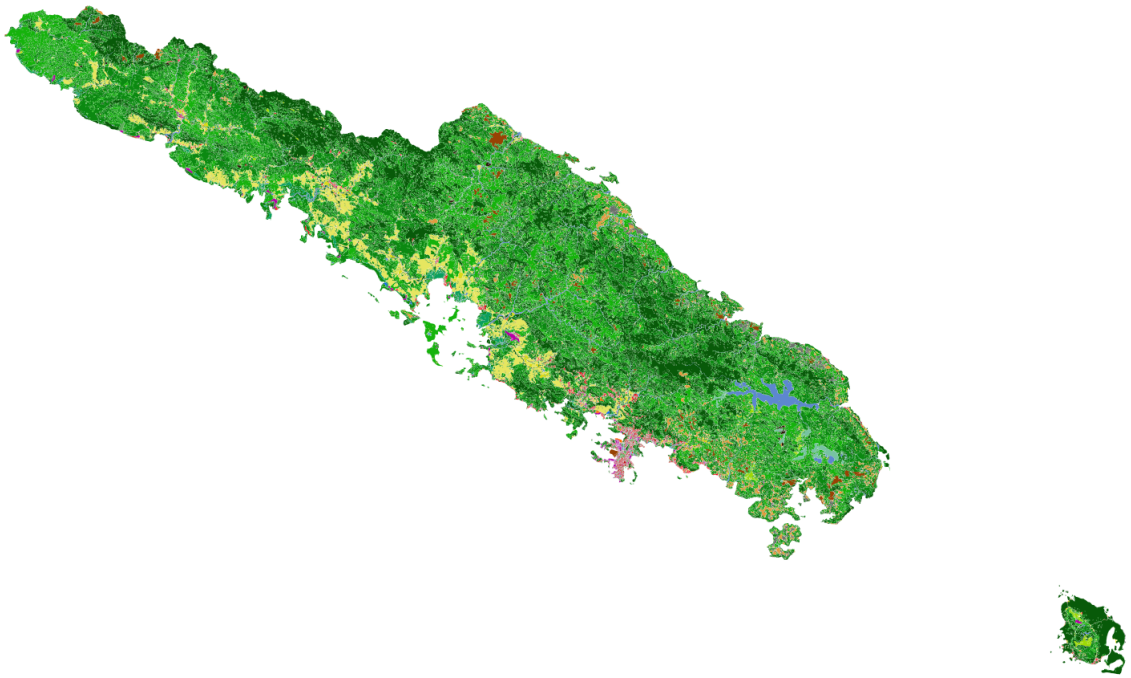


FIGURE 4.2 – MOS de 2014 produit par l'OEIL à partir d'image SPOT6 sur l'ensemble de la province Sud de Nouvelle-Calédonie.

Le résultat du MOS a été obtenu par une méthodologie mixte. Une photo-interprétation et exploitation des données exogènes citées avant a été tout d'abord menée. Puis d'après une nomenclature hiérarchisée (Table 4.1), une classification semi-automatique des grands espaces naturels a été réalisée, le tout sous le contrôle d'un expert qui s'assurait par photo-interprétation que la classe attribuée était correcte. Puis dans une dernière étape, les données ont été adaptées à l'ossature de la base de données topologique des voiries de la province Sud. L'ensemble du MOS a été alors testé sur différentes zones de validation qui totalisent une surface de $160\,km^2$. Ces zones ont été sélectionnées pour la variété des objets, leur accessibilité et la quantité de données exogènes disponibles.

Pour chaque type de sol présenté dans cette nomenclature, un échantillon d'une série d'images satellites est proposé pour illustrer chaque classe. En observant la Figure 4.3, les limites d'une méthode de classification semi-supervisée commencent à apparaître.

L1	L2	L3	Description
1			Territoires artificialisés
	11		Zones urbanisées
		111	Tissu urbain continu
		112	Tissu urbain discontinu
		113	Habitat isolé
	12		Zones industrielles ou commerciales et équipements
	13		Décharges, chantiers, extraction de matériaux : hors mines
	14		Infrastructures et installations minières
	15		Espaces verts artificialisés et équipements
	16		Réseaux de communication
2			Territoires agricoles
	21		Terres arables et cultures permanentes
	23		Terres pastorales aménagées structurées
	24		Plantation d'arbres – sylviculture
3			Formation végétale
	31		Strate arborée
		311	Strate arborée
	32		Strate arbustive et herbacée
		321	Strate arbustive
		322	Strate herbacée
	33		Espaces ouverts, sans ou avec peu de végétation
		331	Plages, dunes et sable
		332	Roches et sols nus
		333	Végétation clairsemée
		334	Zones incendiées
4			Zones humides
	41		Zones humides maritimes
		411	Marais maritimes
		412	Marais salants
		413	Zones intertidales
	42		Zones humides intérieures
5			Surfaces en eau
	51		Eaux continentales
	52		Eaux maritimes

Tableau 4.1 – Nomenclature hiérarchisée du Mode d'Occupation du Sol de Nouvelle-Calédonie sur 3 niveaux.

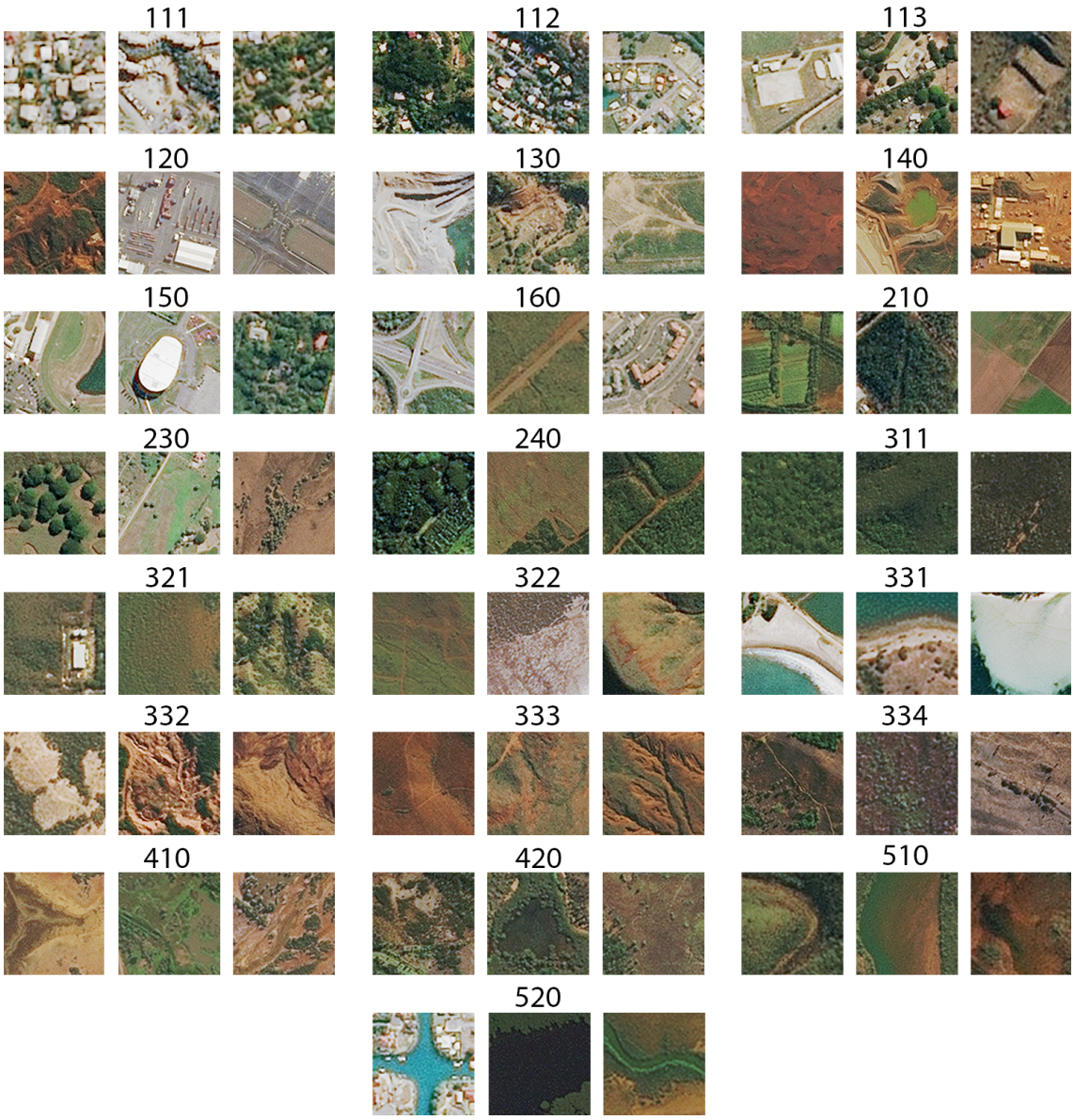


FIGURE 4.3 – Illustration des différentes classes de la nomenclature hiérarchisée du MOS avec des images satellites de SPOT 6

Les enseignements du MOS

Le MOS tel qu'il est présenté aujourd'hui ne convient pas pour une approche en Deep Learning. En effet, le résultat de la classification semi-supervisée est de basse résolution ($> 1\,ha$) par rapport à des données exogènes très précises. De plus, les données exogènes apportent une connaissance supplémentaire que l'architecture en apprentissage ne dispose pas. Par exemple, un chantier de construction, qu'il appartienne à une société minière ou à une société de construction, ne fait pas de différence pour un observateur extérieur, d'autant pour une architecture de Deep Learning. Mais avec la nomenclature telle que présentée Table 4.1, il existe une différence. Cette information doit être donc ignorée sinon le modèle en apprentissage commencera à faire une distinction qui lui est à la base impossible ,hormis en apprenant par coeur la donnée. D'autres problèmes subsistent du fait que des données exogènes ont été exploitées pour la création du MOS de l'OEIL.

Certaines classes sont très peu présentes, par exemple les réseaux routiers ou les plantations d'arbres ne représentent qu'une part trop faible de l'ensemble des données et seront donc difficiles à inclure dans l'apprentissage pour un modèle de Deep Learning. En effet, l'équilibre des classes est un point central de l'apprentissage d'une méthode de machine learning. C'est-à-dire que le nombre d'exemples d'une certaine classe doit être du même ordre de grandeur que les autres, sinon l'architecture risque de négliger entièrement cette classe, car elle affectera très peu la précision globale du modèle.

Comme dit précédemment, certaines classes possèdent une spécification trop précise et ne peuvent pas être distinguées d'une autre classe simplement à partir d'une observation par photo-interprétation par un opérateur humain. Mais une faible distinction entre deux classes peut être également le résultat d'une définition trop générale ou trop imprécise de certaines classes. Les zones industrielles, par exemple, peuvent faire référence à toute surface utilisée dans un contexte commercial. Dans le MOS, des zones telles que des parcs éoliens ou la zone entourant les aéroports sont considérées comme des zones commerciales. Ces deux exemples présentent de vastes régions de sols nus ou couverts d'herbes et sans une connaissance fine du terrain, il est difficile de faire un rapprochement entre ces zones, que ce soit pour un opérateur humain ou une technique d'intelligence artificielle.

Prenons l'exemple que certaines classes du MOS sont classifiées incorrectement. Si on considère que la classe incorrecte n'est pas sous-représentée et donc en manque d'exemples représentatifs, ce n'est alors pas une difficulté pour un apprentissage en

machine learning. En effet, si 95% des exemples sont correctement étiquetés alors les méthodes de machine learning les moins sensibles arriveront à ignorer l'erreur sans entrer en sur-apprentissage. Cette affirmation est à relativiser néanmoins, car si la classe est très complexe et demande une diversité d'exemples très différents les uns des autres, alors une redéfinition de la classe elle-même ou sa division en plusieurs sous-classes devraient être envisagées.

La précision du réseau routier est un autre problème, car c'est une couche vectorielle exogène aux données de télédétection et exploitée en aval de la classification semi-supervisée du MOS. Si cette couche précise au pixel près est erronée ou mal géoréférencée, le décalage de l'image entraînerait une mauvaise labellisation d'un pan entier du réseau routier. Bien que les données SPOT 6 soient à $1.5m$ de résolution, leur orthorectification s'est fait à partir d'un MNT de $10m$ de résolution. Le fait d'avoir des routes en décalage par rapport à leur véritable valeur de pixel et d'autres qui sont bien positionnées peut perturber fortement l'apprentissage, qui se base sur des données contradictoires. Comme le nombre d'exemples erronés contrebalance le nombre d'exemples correctement labellisés, la méthode de machine learning ne parviendra pas à extraire les caractéristiques principales de la classe. De plus, les données de Sentinel-2 sont quant à elle à $5\,m$ de résolution, une résolution ne permettant de distinguer que les axes routiers les plus importants. Une réadaptation partielle ou totale doit donc être étudiée avant tout changement de résolution spatiale de la donnée.

Pour adapter la nomenclature du Deep Learning, il ne suffit pas d'utiliser un niveau inférieur de la nomenclature pour contourner le problème. Le détail des classes serait perdu et la définition des classes serait trop large pour pouvoir faire un apprentissage efficace. L'adaptation de la nomenclature doit permettre l'utilisation des données de la classification du MOS à des fins d'apprentissage, mais la présence d'objet de basse résolution avec des objets de haute résolution ne serait pas réglée. Il faut donc créer de nouvelle nomenclature adaptée à cette problématique et créer les données associées.

C'est ce qui a été effectué dans le chapitre précédent, où des données ont été labellisées à la main pour correspondre aux exigences d'une résolution précise de $1.5m$ du satellite SPOT 6 et des classes qui soient distinguables uniquement par photo-interprétation. Cependant pour le capteur du satellite Sentinel-2, on souhaite utiliser les données du MOS de 2014 pour un apprentissage à grande échelle, une nouvelle définition de la nomenclature et l'adaptation des données du MOS doivent également être menées.

Proposition de nouvelles nomenclatures

Une nouvelle nomenclature de la couverture des sols est proposée dans la table 4.2, ci-dessous :

Tableau 4.2 – Nomenclature de la couverture des sols

N1	Description
1	Territoires artificialisés
2	Sols nus
3	Végétation de forte densité, forêts
4	Végétation de moyenne densité, arbustes
5	Végétation de faible densité, herbes
6	Surfaces d'eau

Cette nomenclature, volontairement simple, permet d'adapter les classifications réalisées dans le MOS de 2014 avec le travail effectué sur la labellisation manuelle présenté Figure 3.6. C'est la classification de l'usage des sols proposée dans le chapitre précédent qui a été adaptée, car elle répond mieux à la notion de tâche urbaine telle que pratiquée dans le MOS.

Partant de la nomenclature présentée Table 3.2 pour aboutir à la nomenclature Table 4.2, des ré-attributions de classes ont été effectuées. Les classes 21, 22, 23 et 31 sont devenues respectivement les classes 3, 4, 5 et 6. Les classes appartenant au tissu urbain (classe 11 et 12) ont été rassemblées dans la classe 1. Et toutes les autres classes (13, 14, 24, 25, 26 et 32) ont été rassemblées dans la classe 2 de la nouvelle nomenclature. Les classes des nuages et des ombres n'existent plus, car l'ensemble des nuages a été masqué au préalable dans la donnée Sentinel-2. Les ombres quant à elles, ne sont présentent que dans le relief montagneux de la Nouvelle-Calédonie, endroit de la région où la forêt est très présente. Dans le reste de cette étude, et comme elles ont été classées dans le MOS de 2014, les régions d'ombres seront considérées ici comme de la forêt.

Partant de la nomenclature Tableau 4.1 pour aboutir à la nomenclature Tableau 4.2, les classes sont ré-attribuées de telle manière que :

- Les territoires artificialisés seront constitués des classes 111, 112, 113, 120 et 130 ;
- Les sols nus seront constitués des classes 130, 140, 160, 331, 332, 333 et 334 ;
- La végétation de forte densité sera constituée des classes 240 et 311 ;
- La végétation de moyenne densité sera la classe 321 ;

- La végétation de faible densité sera constituée des classes 150, 210, 230, 322, 411 et 420 ;
- Les surfaces en eau seront constituées des classes 510 et 520.

4.2.3 Création de données d'apprentissage pour le modèle de Deep Learning

Les techniques d'apprentissage profond utilisées dans cette étude sont principalement basées sur l'utilisation du CNN [69]. Ce type d'architecture a fait ses preuves en télédétection par la succession de couches convolutionnelles et de mise en commun qui permet d'extraire efficacement les informations essentielles d'un pixel tout en conservant le contexte dans lequel il est présent. Ces méthodes sont très performantes dans le cadre de la détection d'objets, en considérant un ensemble de pixels reliés entre eux par une certaine logique. Dans le chapitre précédent, l'architecture DeepLabV3+ s'est distinguée par de bonnes performances dans le cadre de la détection de couverture et d'usage des sols en Nouvelle-Calédonie à partir de données très haute résolution. Dans ce chapitre, les recherches menées sur la détection des changements d'occupation des sols se feront sur la base de cette unique architecture.

Les données Sentinel-2 couvrant partiellement la province Sud de la Nouvelle-Calédonie de 2016 à 2020 sont récupérées. Comme Sentinel-2B n'était pas en fonctionnement pour l'année 2016, et du fait de la forte couverture nuageuse, seule une couverture partielle de la Nouvelle-Calédonie a pu être réalisée pour l'année 2016. Sur les années 2017, 2018 et 2019, seules quelques zones ne disposeront pas d'une couverture satellite, notamment la région de la commune de Thio qui est absente. L'année 2020 dispose d'une couverture complète.

Les différents canaux des images satellites sont présentés dans des fichiers individuels. Sur toutes les bandes disponibles, seules les données à $10m$ seront retenues, c'est-à-dire 4 canaux : RVB et PIR, et réassemblés en un seul fichier avant toute exploitation. De l'expérience du traitement des données SPOT 6, ces 4 canaux sont suffisants pour réaliser une carte de la couverture terrestre.

Sur l'ensemble des données récupérées, un masque nuage créer manuellement est appliqué pour chaque image. Un masque du territoire de la Nouvelle-Calédonie est aussi appliqué, retirant toutes les zones de lagon et de mer autour des terres. Une mosaïque de l'ensemble des données est réalisée en prenant la moyenne des pixels lorsqu'ils il y a un recouvrement de plusieurs images, puis reprojeté dans le système de Lambert (RGCN91-93 / Lambert New Caledonia).

Une remise à l'échelle des données des entiers de 0 à 255 (8-bits) est effectuée d'après la formule ci-dessous, où $x.min$ et $x.max$ sont calculés à partir des quantiles fixés respectivement à $0,02$ et $0,98$ pour limités les valeurs extrêmes dans la distribution.

$$x' = 255 \times \frac{(x - x.min)}{(x.max - x.min)} \tag{4.1}$$

Par la suite les images sont découpées selon la grille présentée Figure 4.4, cette même grille qui découpe les données SPOT 6 utilisé au chapitre précédent. Par la suite les données sont rassemblées pour constituer le jeu de données. Ce jeu de données utilisera toutes les dalles en vert comme jeu d'apprentissage. Les données en rouge sont quant à elle exclues de ce jeu d'apprentissage, car elles serviront de zones de test pour notre modèle en sortie.

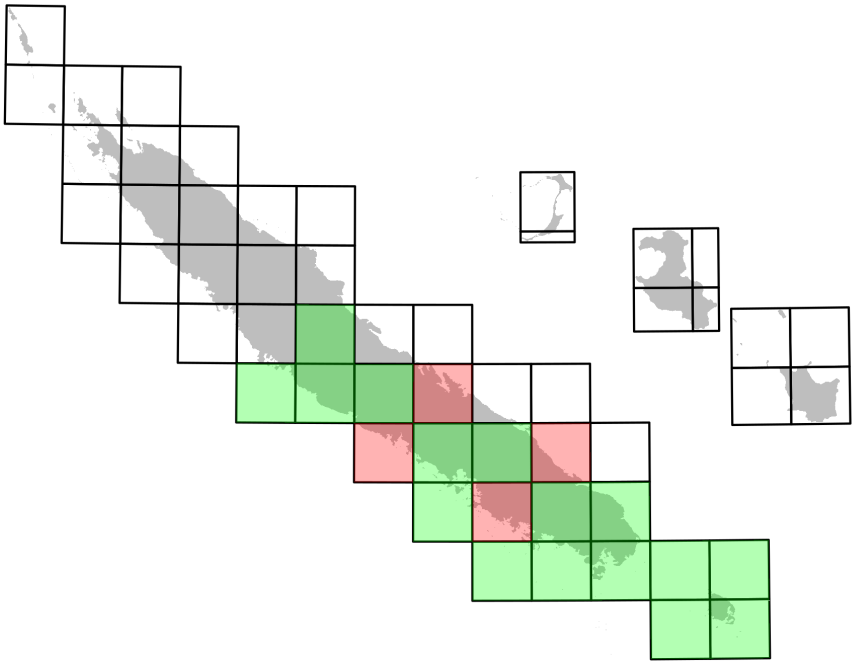


FIGURE 4.4 – Grille de découpage des données satellites. En vert les images utilisées pour l'apprentissage. En rouge les images utilisées pour le test.

Les données vecteurs du MOS de 2014 sont donc « rasterisées » à la résolution des données satellite Sentinel-2, à $10\,m$ de résolution. Un index est ensuite créé sur l'ensemble des données afin d'équilibrer les classes. En effet cet index permettra de référencer le nombre de pixels de chaque classe présente dans les images, une pondération pourra alors être appliquée en plus des données d'entrées pour favoriser les classes les moins représentées.

4.3 Analyse et discussion autour des résultats

Dans cette section, le résultat du modèle de Deep Learning est directement comparé aux 5 zones étiquetées à la main (Chapitre 3), puis comparé aux données du MOS de 2014 produit pour le compte de la Nouvelle-Calédonie. Dans le but d'automatiser le processus de création d'une cartographie complète de la couverture du sol de tout un territoire sur une année, une discussion sera menée sur les moyens à mettre en oeuvre pour y parvenir, à la lumière des résultats présentés.

4.3.1 Le MOS comparé à la réalité

Tout d'abord, une comparaison entre les données du MOS de 2014 et les données des 5 zones étiquetées manuellement est réalisée. Pour cela, ces deux jeux de données qui sont sous un format vectoriel (sous forme de polygone géométrique) sont rééchantillonnés à l'échelle des données satellites Sentinel-2. C'est-à-dire qu'ils sont mis sous le format d'une grille de pixels de $10\,m$ de résolution. Pour effectuer cette comparaison, les données du MOS et les données de l'usage des sols sont rapportées à la classification simplifiée présentée Tableau 4.2 et détaillée précédemment.

Deux matrices de confusion de cette comparaison sont montrées : la première (Tableau 4.3) présente les scores associés à la précision du point de vue de l'utilisateur (PU) et la deuxième (Tableau 4.4) du point de vue du producteur (PP). Une précision globale de $61,79\,\%$ et un F1-score de $62,17\,\%$ sont obtenus. Une PU de $67,53\,\%$ et une PP de $57,59\,\%$ sont aussi obtenues. Cependant, une grande disparité entre les deux matrices, avec un écart pouvant aller jusqu'à $50\,\%$ entre les deux précisions, est aussi constatée.

Tableau 4.3 – Matrice de confusion présentant la précision utilisateur entre la couverture du sol produite manuellement et le MOS de 2014.

	Prédictions - PU					
	1	2	3	4	5	6
1	**86,26**	6,03	1,50	2,12	4,07	0,02
2	2,94	**29,03**	1,36	8,95	34,44	23,28
3	4,38	1,06	53,84	35,31	4,72	0,69
4	2,79	1,54	14,31	60,09	20,64	0,64
5	0,57	2,07	0,81	16,69	79,63	0,24
6	0,00	1,07	0,24	1,63	0,71	**96,35**

TABLEAU 4.4 – Matrice de confusion présentant la précision producteur entre la couverture du sol produite manuellement et le MOS de 2014.

	Prédictions - PP					
	1	2	3	4	5	6
1	**36,54**	2,23	0,13	0,11	0,20	0,01
2	7,86	**67,87**	0,72	2,92	10,76	47,48
3	32,29	6,83	78,54	31,70	4,05	3,86
4	19,11	9,19	19,41	50,16	16,49	3,33
5	4,19	13,32	1,18	14,99	68,45	1,32
6	0,00	0,56	0,03	0,12	0,05	**44,00**

La disparité qu'il existe entre la PP et la PU résulte directement des deux méthodologies appliquées pour obtenir les deux classifications. Cette différence est visible Figure 4.5, où un exemple de l'adaptation des deux classifications à la nouvelle nomenclature (Tableau 4.2) y est illustré sur la commune de Thio.

Ainsi, en considérant la classification manuelle comme étant la vérité terrain, le MOS de 2014 surestime l'étendue de la tâche urbaine. Cette surdétection est visible Figure 4.5 et se manifeste dans les matrices de confusion avec le faible score de la PP (36, 54 % comparé à 86, 26 % pour la PU). De même pour les rivières où la vérité terrain, c'est-à-dire la classification manuelle, classe les rivières en tant que sol nu (zones asséchées) ou en tant que végétation (zones sous couvert végétal). Une forte PU (96, 35 %) est constaté au détriment de la PP (44, 00 %). Inversement, des zones entières considérées comme sols nus par la vérité terrain sont absentes du MOS de 2014. C'est le phénomène inverse qui s'opère, la plupart des zones étant des sols nus sont correctement étiquetées par le MOS de 2014 (PP de 67, 87 %) mais leur étendue totale est sous-estimée (PU de 29, 03 %). Enfin pour les strates herbacées, arbustives et arborées, les classes ont en moyenne un taux de reconnaissances proche de 60 %, résultant quasi intégralement du floue de la frontière qu'il existe entre ces différentes classes.

En fonction des méthodes employées, de la subjectivité de l'opérateur humain et la typologie choisit, de très grandes différences dans l'obtention d'un mode des occupations du sol d'un territoire sont constatées. Cette différence, que l'on peut qualifier de subjectivité dans la qualification de la donnée, est un frein dans la mise en place de méthodes de machine learning comme nous le verrons dans la suite de ce chapitre.

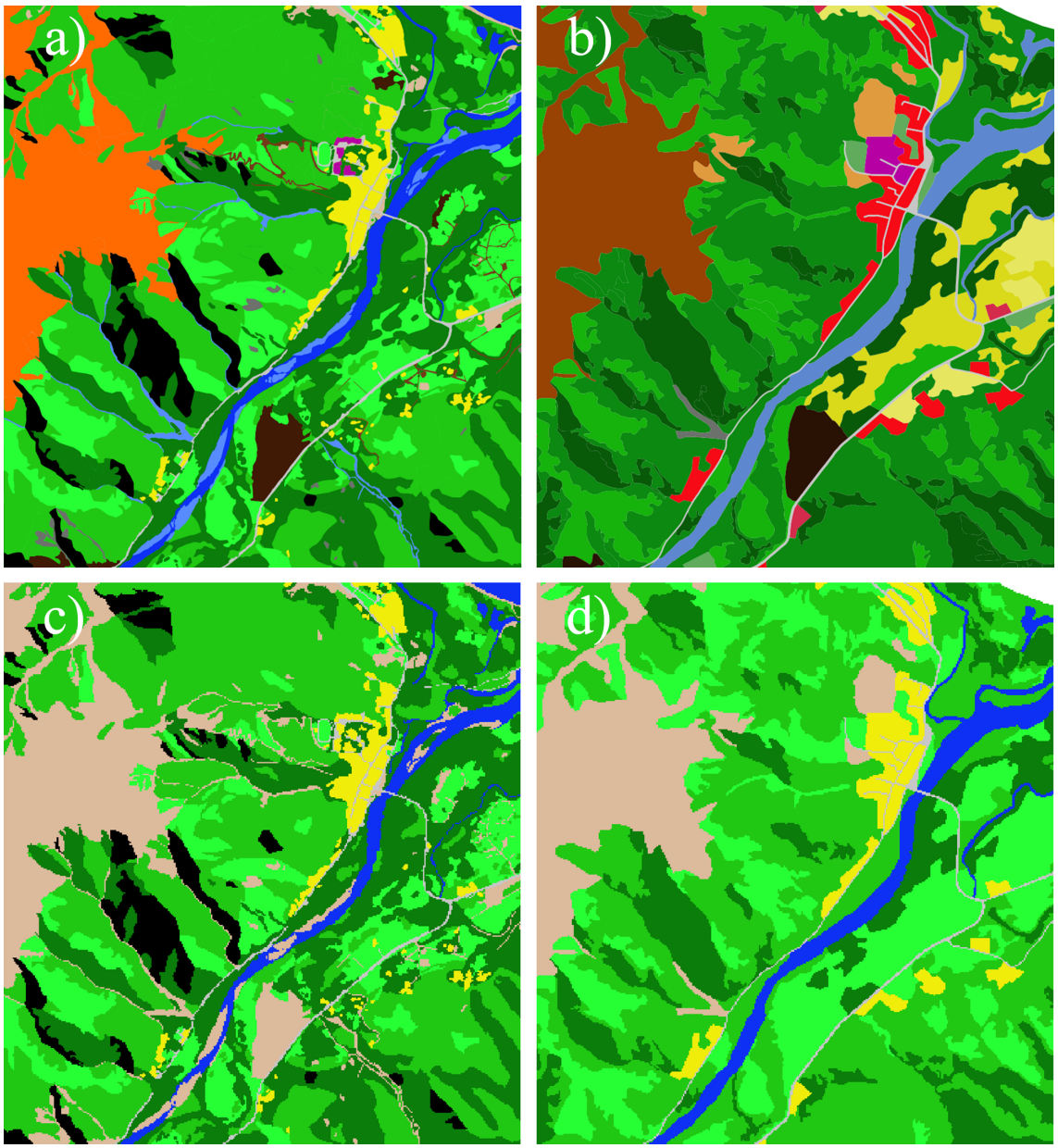


FIGURE 4.5 – La classification manuelle (a) et le MOS de 2014 sur la commune de Thio, adaptés respectivement (c et d) à la nouvelle nomenclature, Tableau 4.2.

4.3.2 Le Deep Learning appliqué au MOS

L'ensemble des données Sentinel-2 ont été collectées pour créer des mosaïques annuelles de l'année 2016 à l'année 2020. Le modèle de Deep Learning DeepLabV3+ est entraîné à partir de ces données satellites et du MOS de 2014. Les données Sentinel-2 de 2016 sont utilisées pour entraîner le modèle. L'entraînement et le test de l'architecture sont effectués en accord avec le découpage des données indiqué Figure 4.4. Les résultats sur les zones de tests (zones rouges de la Figure 4.4) sont présentés sous forme de matrices de confusion, une présentant la PU (Tableau 4.5) et l'autre la PP (Tableau 4.6).

Une précision globale de 69, 95 % et un F1-score de 66, 74 % sont obtenus. Pour information, sur l'ensemble de la province Sud (zone d'entraînement inclus) c'est une précision globale de 79, 20 % et un F1-score de 76, 36 % qui sont obtenus. Sur le jeu de test de nouveau, c'est une PU globale de 71, 20 % et une PP globale de 62, 80 % qui sont obtenues.

TABLEAU 4.5 – Matrice de confusion présentant la précision utilisateur entre la sortie de l'architecture de Deep Learning DeepLab et le MOS de 2014.

	Prédictions - PU : 71,20%					
	1	2	3	4	5	6
1	77,26	1,91	1,28	3,12	14,40	2,02
2	10,83	61,74	1,51	5,51	17,57	2,85
3	1,07	0,16	79,07	15,75	3,40	0,55
4	1,69	0,79	18,65	61,25	16,96	0,67
5	2,49	4,57	1,55	19,81	71,02	0,56
6	2,37	2,52	4,94	6,97	6,14	77,05

TABLEAU 4.6 – Matrice de confusion présentant la précision producteur entre la sortie de l'architecture de Deep Learning DeepLab et le MOS de 2014.

	Prédictions - PP : 62,80%					
	1	2	3	4	5	6
1	52,73	1,45	0,12	0,28	1,50	3,73
2	8,83	56,00	0,17	0,60	2,18	6,29
3	7,71	1,26	77,57	15,16	3,74	10,75
4	13,73	7,10	20,57	66,30	20,98	14,63
5	16,38	33,45	1,39	17,41	71,34	9,93
6	0,62	0,74	0,18	0,24	0,25	54,67

Dans cette table, la différence entre la précision utilisateur et producteur est bien moindre comparée à celle observée entre les Tableaux 4.3 et 4.4. Cependant les mêmes types de biais sont constatés pour la classe urbaine et la classe des zones en eau. Le modèle de Deep Learning a tendance à surestimer la présence de ces classes. Idem pour la classe de sol nu, ce qui diffère par rapport aux observations faites dans les matrices de confusion précédentes. Les classes de végétation ont des résultats quasi similaires pour la PU et la PP, la majeure partie des erreurs du modèle se résumant, comme précédemment, à l'ambiguïté de la frontière existant entre les trois classes de végétation.

Le modèle, entraîné à partir de données Sentinel-2 sur l'ensemble de la province Sud, est appliqué aux mosaïques des années suivantes, Figure 4.6. D'une année sur l'autre, il est à noter que les surfaces remarquables (le lac de Yaté au Sud, la zone urbaine de Nouméa, les grands ensembles forestiers) sont correctement détectées et n'évoluent pas sensiblement. A contrario, les résultats des classifications mettent en évidence une grande variabilité des classes de végétation sur l'ensemble du territoire. Ces changements, qu'ils soient induits par un changement d'occupation du sol, une prise d'images satellitaires opérée dans des contextes climatiques différents (pluie, sécheresse...) ou d'une erreur du modèle, doivent faire l'objet d'une vérification par photo-interprétation. Malheureusement, ce sont des données qui ne sont pas disponibles et qui demanderaient la mobilisation d'un opérateur humain. Il n'est donc pas possible à l'heure de cette étude de quantifier l'erreur du modèle d'une année sur l'autre. Cependant plusieurs suggestions d'améliorations du processus sont possibles.

La bonne qualité d'une méthode de machine learning est liée à la qualité des données utilisées pour l'apprentissage du modèle. Ici, les données Sentinel-2 utilisées dans la mise au point des mosaïques annuelles devraient être bisannuelles pour prendre en compte la saisonnalité de la Nouvelle-Calédonie. De la saison sèche à la saison fraîche, de grandes différences de précipitations sont mesurées avec donc des conséquences directes sur la végétation et donc sur le signal radiométrique renvoyé. Il a été considéré au début de cette étude que les architectures de Deep Learning, du fait de leur grande capacité d'abstraction de la donnée, étaient capables de mesurer ce biais et de classifier correctement un type de sol malgré la saisonnalité. Cependant, s'affranchir de cette variable en proposant un modèle pour chaque saison, permettrait d'améliorer la fiabilité des modèles.

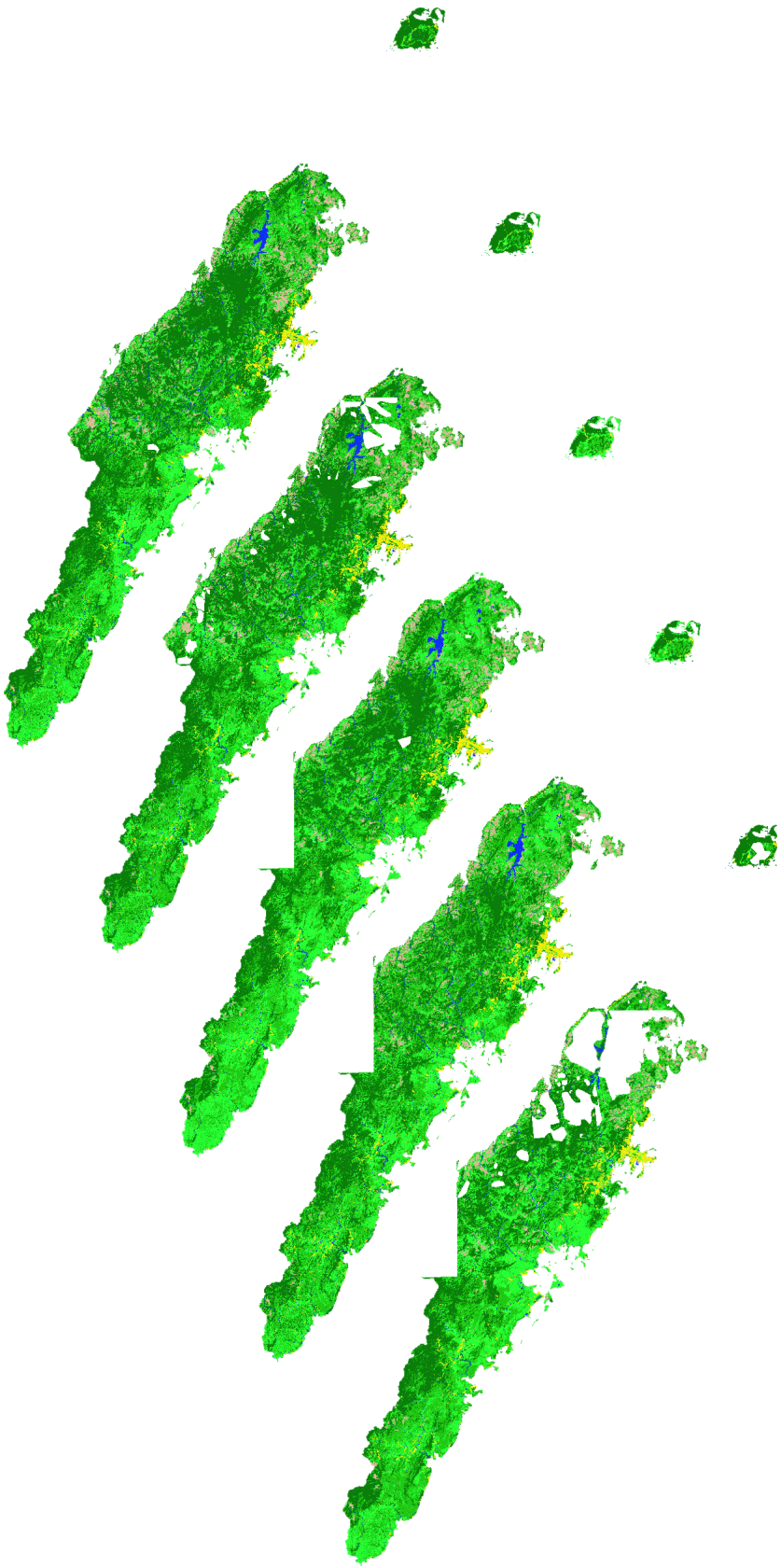


FIGURE 4.6 – Cartographie de la couverture du sol de la province Sud des années 2016 à 2020 grâce à l'architecture DeepLabV3+ à partir de données Sentinel-2.

Du fait du peu de données disponibles à cause de la forte couverture nuageuse dans la zone du Pacifique Sud, toutes les données disponibles sont exploitées sans prendre en compte les paramètres de leur acquisition. En fonction de l'ensoleillement, de l'angle de prise du satellite, des variations dans le signal radiométrique sont à prendre en compte. Dans la création des mosaïques, un travail plus approfondi du traitement du signal doit être conduit pour s'assurer de la qualité des données d'apprentissage.

En outre la qualité des données satellites, la vérité terrain, c'est à dire ici les données du MOS de 2014, doivent répondre à des critères de qualité et de consistance pour assurer un bon apprentissage. Les données du MOS avec une finesse d'objet n'allant pas en dessous de $1km^2$, (ex. : une surface en eau inférieure à $1km^2$ n'est pas référencée) utilise des données exogènes de précision submétrique. Cette fusion de données à des résolutions différentes résulte en des inconsistances des résultats finaux, perturbant les architectures de Deep Learning.

4.4 Conclusions

Dans cette étude, les résultats du MOS ont été comparés avec la classification manuelle créée dans le chapitre précédent dans une matrice de confusion. Un score de $62,17\,\%$ pour le F1-score a été obtenu pour une classification de 6 classes selon la nouvelle nomenclature de la couverture terrestre Tableau 4.2. Ce faible score s'explique par le manque de précision induit par la méthode utilisée pour créer le MOS de 2014 et par la subjectivité de la limite de certaines classes.

Sur ce même MOS, une architecture de Deep Learning, DeepLabV3+, a été entraînée pour une application à grande échelle sur l'ensemble du territoire. Sur les résultats du modèle, un F1-score de $66,74\,\%$ a été obtenu sur l'ensemble de tests et un F1-score de $76,36\,\%$ a été obtenu sur l'ensemble de la province Sud. Bien que des performances similaires aient été obtenues, il est difficile d'évaluer la précision du modèle. En effet du fait du faible score obtenu précédemment entre la vérité terrain et les données utilisées en apprentissage, il est difficile de déterminer si les erreurs de classification proviennent des données d'entraînement ou des résultats du modèle.

Les différentes nomenclatures utilisées dans cette étude doivent être mieux définies pour d'une part répondre au mieux aux différentes problématiques du territoire, et d'autre part respecter une cohérence dans la définition de ces classes pour permettre l'automatisation de la détection. Par exemple dans la nomenclature présentée Tableau 4.1, une confusion entre les notions de couverture et d'usage du sol est observée. Une zone incendiée, en fonction de son niveau de recolonisation de la végétation, peut être classifiée comme un sol nu, comme une strate herbacée ou comme une strate arbustive.

Le produit tel que présenté actuellement peut faire encore l'objet de nombreuses améliorations bien qu'il soit capable de produire une carte de la couverture terrestre à l'échelle du territoire de la Nouvelle-Calédonie, Figure 4.7. Cependant, nous pouvons constater sur cette figure que de nombreuses zones de sols nus sont classifiées comme étant de l'urbain en province Nord. En effet, un modèle appris uniquement sur une zone géographique spécifique est difficilement généralisable à d'autres zones, surtout si elles répondent à des dynamiques environnementales différentes.

Pour une application plus générale sur l'ensemble du territoire, il faut donc échantillonner des données qualifiées supplémentaires, reflétant au mieux la grande diversité des territoires de la Nouvelle-Calédonie. Pour cela, nous proposons d'utiliser des architectures de Deep Learning classant le pixel central d'une image plutôt que celles effectuant une segmentation sémantique. Ces deux types d'architectures ont des per-

formances égales, mais dans l'acquisition rapide par photo-interprétation et à large échelle de données de terrain, il est plus aisé d'utiliser un moyen d'échantillonnage par point.

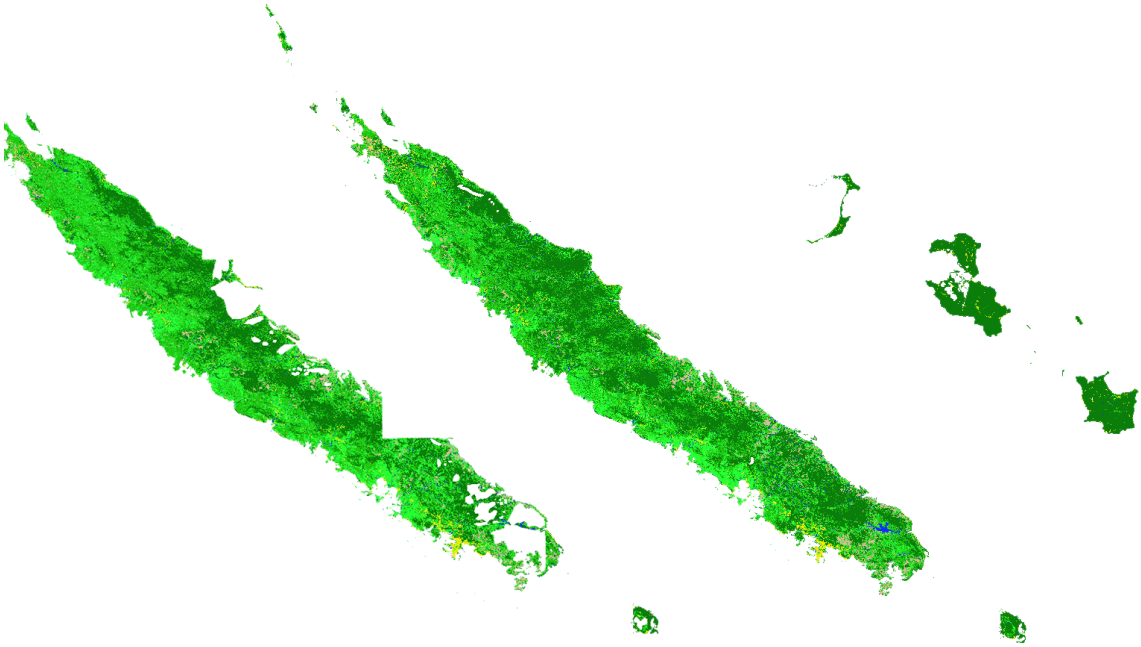
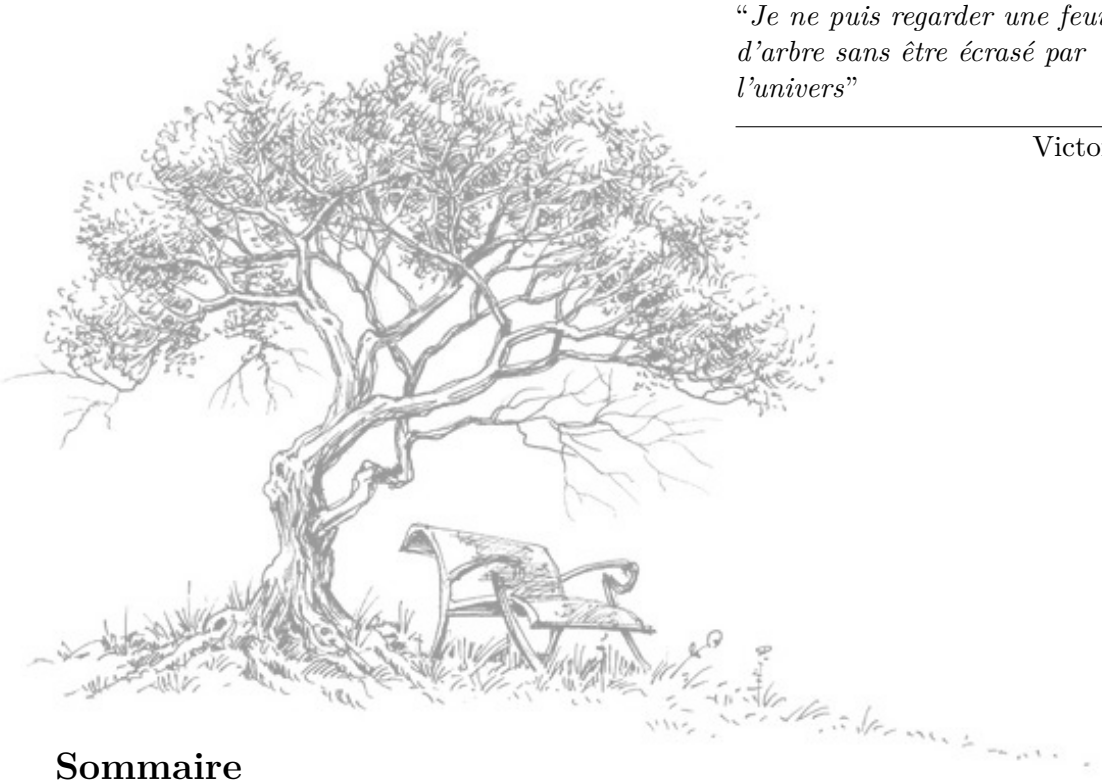


FIGURE 4.7 – Cartographie de la couverture du sol de Nouvelle-Calédonie des années 2016 et 2020 grâce à l'architecture DeepLabV3+ à partir de données Sentinel-2.

CHAPITRE 5

DÉTECTION DU BABAÇU AU BRÉSIL



"Je ne puis regarder une feuille d'arbre sans être écrasé par l'univers"

Victor Hugo

Sommaire

5.1 Introduction 133
5.1.1 Caractérisation de la zone d'étude 133
5.1.2 La télédétection de palmiers 135
5.1.3 Objectif de l'étude 137
5.2 Données et méthodes 138
5.2.1 Les données satellites Pléiades 138
5.2.2 Les données DART 138
5.2.3 Gestion des données 140
5.2.4 Évaluation des modèles de Deep Learning 143
5.3 Analyse et performances des modèles 146
5.3.1 Comparaison des architectures de Deep Learning 146
5.3.2 Apport des données simulées à l'apprentissage machine . 148
5.3.3 Détection du babaçu à partir d'une image satellite 151
5.4 Conclusions 153

5.1 Introduction

Dans ce chapitre, une autre grande problématique du domaine de la télédétection est abordée : la détection d'objet. Des architectures de Deep Learning seront appliquées à des données satellites Pléiades pour la détection du babaçu au Brésil. En plus de la détection d'objet, des données DART entièrement simulées informatiquement seront utilisées dans l'apprentissage des réseaux de neurones. La caractérisation de cette nouvelle zone d'étude et les notions entourant la détection d'objets ainsi que les données simulées sont détaillées dans cette section.

5.1.1 Caractérisation de la zone d'étude

Le Pará, état du nord du Brésil, s'étend sur une superficie d'environ 1,25 million de kilomètres carrés pour une densité humaine de seulement 4 hab/km^2. Sous l'influence d'un climat tropical humide, cette région, centrée autour de l'estuaire de l'Amazone, est recouverte en grande partie par de forêts, qui abrite un dixième des espèces vivantes mondiales [178]. La zone d'étude est située à l'est de l'état de Pará (Figure 5.1) et abrite les terres agricoles de Benfica (S 05°16'20", W 49°50'25").

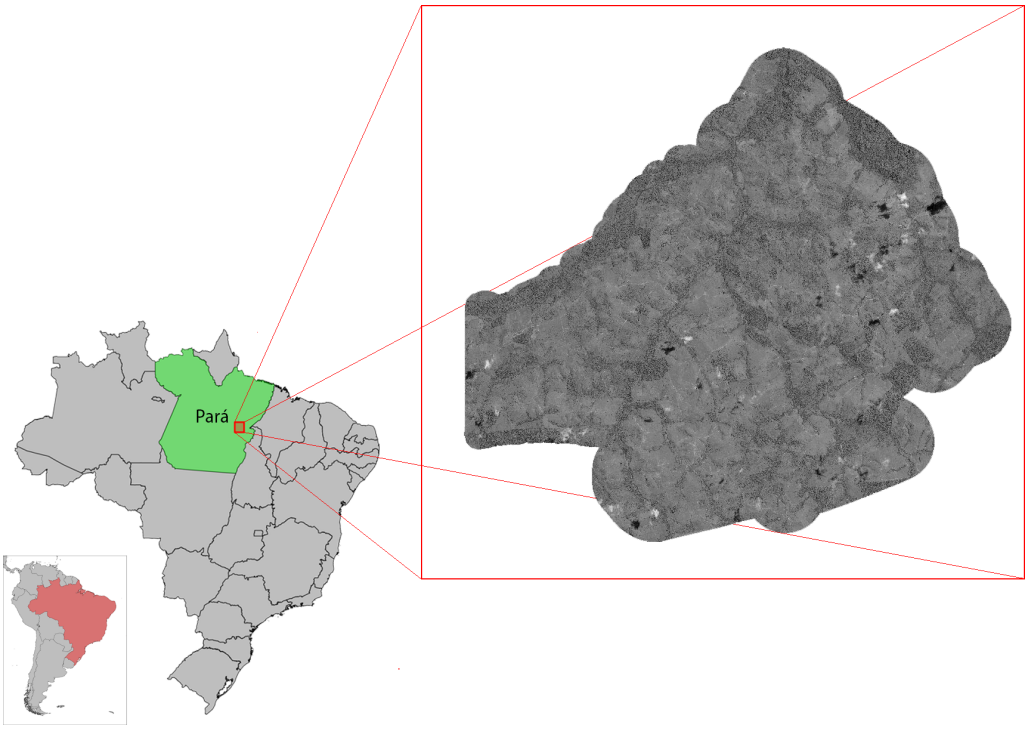


FIGURE 5.1 – Image satellite Pléiades montrant la zone d'étude dans les terres agricoles de Benfica au Brésil.

Sur les 9 501 ha de sa superficie, forêts et pâturages se disputent les terres agricoles de Benfica. Le territoire est découpé en une succession de segments, un tiers de sa surface alterne forêts primaires et secondaires, un autre tiers des zones de pâturages et de forêts secondaires en devenir, le dernier tiers des zones de pâturages abandonnées et des sols nus. De nombreux palmiers prospèrent dans cette région, aussi bien dans les parcelles de forêt que dans les zones dégagées [54]. C'est l'étude de ces palmiers qui a fait de cette région le lieu d'un projet de recherche bénéficiant du soutien du CNES via le programme national TOSCA qui a produit déjà plusieurs articles scientifiques [57, 179].

Le palmier babaçu

Babaçu est le mot portugais qui désigne une espèce de palmier présente au Brésil, l'*Attalea speciosa* Mart. ex Spreng. de la famille des Arecaceae (Figure 5.2). Il couvre plus de 200 000 km^2 des terres du Brésil où il est signalé [180]. Parfois de nature envahissante [181], certains cultivateurs les défrichent des terres agricoles, mais de par sa résilience et de par sa capacité de reproduction [182], le babaçu arrive à prospérer dans la région de Benfica.

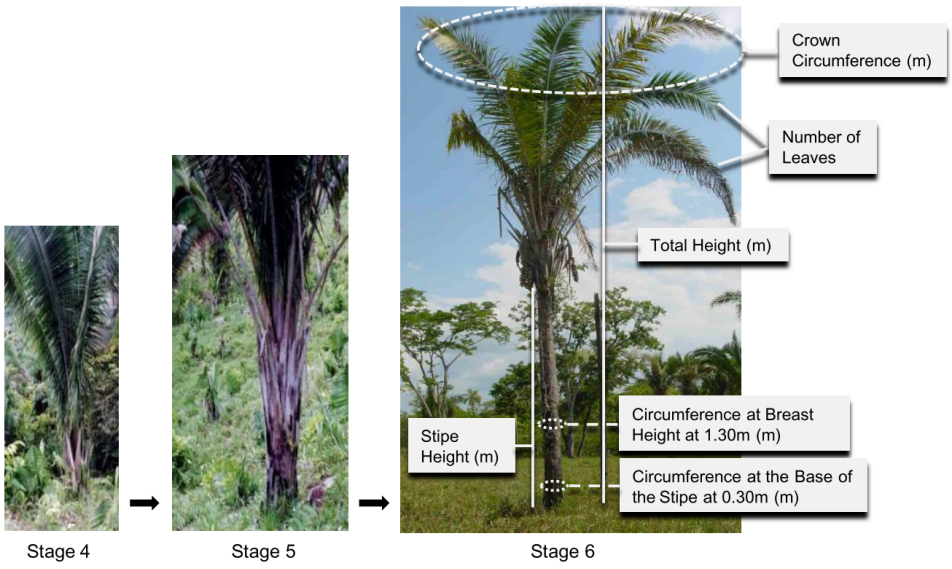


FIGURE 5.2 – Les 3 derniers stades du cycle de vie du babaçu et les informations structurelles du palmier à Benfica. Image extraite de [57].

Il pousse à la fois dans les pâturages et au cœur de la forêt tropicale, où il y est difficile d'obtenir un décompte précis. Des informations sur la quantité de semences commercialisées et de revenus correspondants sont disponibles grâce à l'Institut

brésilien de géographie et de statistique (IBGE), mais sont insuffisantes pour dresser l'inventaire et le calcul de la production annuelle de babaçu. Information essentielle, car les palmiers sont utilisés de diverses manières par les populations rurales les plus pauvres, sa principale exploitation étant la production d'huile à partir de ses graines [55].

5.1.2 La télédétection de palmiers

La détection d'une espèce d'arbre en forêt tropicale dans une image de télédétection est une tâche réputée difficile et ceci pour plusieurs raisons. La première difficulté provient d'une caractéristique connue en écologie d'un écosystème complexe comme celui des forêts tropicales : le nombre d'espèces est une variable ayant une distribution possédant d'une queue épaisse et longue [183, 184]. De tels écosystèmes sont composés de quelques espèces possédant un nombre important d'individus et un grand nombre d'espèces constituées de peu d'individus (ou peu d'individus détectés). La seconde est que ce milieu végétal possède une structuration aléatoire avec des couronnes d'arbres de différentes espèces qui se côtoient et se chevauchent ce qui rend complexe la détection.

La détection automatique du babaçu représente un enjeu majeur pour le décompte et la surveillance de ce palmier présent sur le sol brésilien. Cela est rendu possible grâce à l'exploitation de l'imagerie satellitaire haute résolution, car le processus serait trop long et trop coûteux s'il était effectué manuellement. Cette reconnaissance automatique permettrait de surveiller l'état de santé des milieux abritant les palmiers [185] sur de très grandes surfaces, d'assurer une meilleure gestion de cette ressource [186] et de permettre le suivi des dates de récolte [187].

En ce qui concerne le babaçu, des travaux de télédétection ont déjà été réalisés en milieu ouvert. Un algorithme prenant en compte l'ombre projetée par les palmiers sur le pâturage a été réalisé [57]. Par ailleurs, des algorithmes axés sur la couronne des feuilles de palmier pour les détecter ont été mis en œuvre [56] et pourraient être applicables pour le babaçu. Cependant, ces méthodes ne sont plus exploitables en milieu forestier, où la canopée peut atteindre 25 à 30 mètres d'altitude, certaines espèces d'arbres forestiers atteignant parfois 50 mètres [188]. En effet, les couronnes des arbres se retrouvent le plus souvent accolées les unes aux autres et les arbres plus grands occultent en partie les couronnes des arbres plus petits. Les ombres également ne sont plus visibles dans ces milieux. Des techniques de reconnaissance d'objets plus performantes doivent être mises en œuvre.

La détection d'objet

La détection d'objets dans une image consiste à identifier les emplacements correspondant aux caractéristiques de l'objet. Pour déterminer les emplacements potentiels dans une image, il existe plusieurs méthodes. Par exemple, en utilisant une fenêtre glissante sur l'ensemble de l'image ou encore en détectant des points d'intérêts. Une fois les candidats identifiés, l'extraction des caractéristiques principales peut s'opérer à l'aide du Deep Learning. Grâce à ses capacités d'abstraction de données, le Deep Learning représente en effet une technique parmi les plus performantes dans le domaine de la détection d'objets [15] et dans le domaine de la télédétection en général [189, 13]. C'est l'application de filtres successifs dans la phase d'encodage de l'architecture qui permet d'extraire l'information essentielle. L'objet est alors classé selon une base de données de référence pour la classification finale.

S'il est possible de faire de la détection d'objet à partir de positions relatives de l'image, c'est-à-dire avec des boîtes entourant l'objet identifié, il est aussi possible d'identifier les contours de l'objet. Il s'agit pour cela de faire de la segmentation d'image, chaque pixel identifié sera alors labellisé comme appartenant à l'objet. C'est le cas pour des études sur la conduite autonome, où un modèle parvient à identifier en temps réel, à partir d'un flux vidéo, la route, les piétons, les voitures et tous les éléments nécessaires à la conduite [190]. Des études appliquées plus spécifiquement à la reconnaissance de palmiers à l'aide du Deep Learning ont aussi été menées avec succès [52, 53].

Il est donc possible d'appliquer les notions vues précédemment, la segmentation sémantique utilisée dans la création d'une cartographie de l'occupation du sol, à la détection d'objet dans une image. Une méthode de Deep Learning appliquée à des images de télédétection haute résolution permettrait une telle mise en oeuvre sur l'intégralité des terres agricoles de Benfica.

Le Deep Learning et les données simulées

Un frein majeur à l'exploitation des techniques d'apprentissage profond pour la détection d'objets est la nécessité de disposer de nombreuses annotations. Il existe donc une communauté très active parmi les chercheurs en Intelligence Artificielle pour développer des méthodes requérant le moins d'annotations coûteuses possible, par exemple en considérant les approches d'apprentissage actif [191], de transfert de modèle appris (transfer learning [192]) et toutes les variantes du type « few-shot learning » [193], « zero-shot learning » [194] ou « one-shot learning » [195]. Une autre

technique consiste à enrichir la base d'apprentissage par des données simulées. C'est ce que nous proposons de faire dans ce chapitre.

Comme nous avons pu le voir dans le chapitre précédent, à l'échelle d'un territoire, les données référencées pour la détection des types de sols sont nombreuses et suffisantes pour une détection par apprentissage profond. Cependant, à une échelle spatiale plus fine telle que celle d'un objet (arbres, voitures, etc.), les informations peuvent s'avérer insuffisantes et ne permettent pas d'opérer un apprentissage satisfaisant lorsque le réseau de neurones dense est trop complexe. Dans le cas de la détection de palmier, il est très difficile et coûteux de collecter suffisamment de données de terrain annotées. Pour pallier à ce manque de données de terrain, une solution consiste à simuler des données afin de compléter les bases de données d'apprentissage. Pour ce faire, nous nous sommes approchés d'une équipe développant des travaux de recherche visant à simuler par ordinateur des données de télédétection, et en particulier à créer artificiellement des palmiers dans ces images.

DART est un acronyme qui signifie Discrete Anisotropic Radiative Transfer. Créé par le CESBIO en 1992, il s'agit d'un modèle de transfert radiatif 3D. Ce modèle permet la création de données de télédétection artificielle simulant le comportement des capteurs d'un satellite, à la fois avec des capteurs optiques et des capteurs de type LIDAR [196]. Il peut ainsi simuler tout type d'environnement sur terre comme s'il était observé depuis un satellite, c'est-à-dire en simulant tout type d'atmosphère et d'inclinaison du capteur du satellite. Depuis l'environnement naturel comme les surfaces d'eau, les zones de végétation jusqu'aux surfaces urbaines, le modèle peut créer n'importe quelle scène pour n'importe quelle inclinaison et illumination du soleil [58]. Ces données seront ensuite utilisées pour créer des ensembles de données qui seront ensuite fournies comme entrée de modèle aux architectures de Deep Learning.

5.1.3 Objectif de l'étude

L'objectif ici est d'étudier l'adéquation qui peut exister entre les techniques de machine learning et les méthodes de simulation de la donnée pour l'observation terrestre de milieux naturels peu référencés. Cette étude passe par l'exploitation des données de babaçu, simulées via le modèle DART, dans le but d'améliorer les performances des techniques de détection automatique de ces palmiers dans un environnement forestier complexe.

5.2 Données et méthodes

Le détail ainsi que la gestion des données satellites Pléiades et des données simulées sous la forme de jeux de données sont explicités dans cette section.

5.2.1 Les données satellites Pléiades

Initié par le CNES, Pléiades est un couple de deux satellites optiques d'observation de la Terre opérant sur la même orbite à 180° l'un de l'autre. Les satellites Pléiades 1A et Pléiades 1B fournissent des images à très hautes résolutions spatiales quotidiennement (sous certaines conditions) sur les zones géographiques. L'objectif de cette constellation de satellites est de fournir des données capables de produire une cartographie fine pour de nombreuses thématiques, comme l'aménagement du territoire (urbanisme, agriculture, génie civil...), la sécurité civile ou encore la défense. Mais aussi pour des problématiques environnementales telles que des surveillances de forêt, de la mer et du littoral.

Pléiades-1A a été lancé le 17 décembre 2011 et Pléiades-1B a été lancé 2 décembre 2012. Pléiades disposent de 5 canaux : un canal panchromatique (0.480–0.830 μm) à 70 cm de résolution qui peut être ré-échantillonné à 50 cm de résolution, des canaux couleurs bleu (0.430–0.550 μm), vert (0.490–0.610 μm) et rouge (0.600–0.720 μm), et d'un canal proche infrarouge (0.750–0.950 μm). Pléiades a une fauchée au sol de 20 km de largeur et peut faire une mosaïque de 100 $km \times 100\,km$ en un seul passage.

Une image Pléiades acquise le 12 juillet 2019 est utilisée. L'image est orthorectifiée et couvre toute la zone d'étude pour une étendue totale d'environ 18 $km \times 15,6\,km$. Seul le mode panchromatique avec une résolution de 70 cm rééchantillonné à 50 cm est utilisé. Une partie de l'image sera utilisée pour la constitution de l'ensemble de données d'apprentissage et une autre partie indépendante géographiquement sera utilisée pour l'ensemble de données de test.

5.2.2 Les données DART

Les paramètres DART sont choisis de telle sorte que les données créées par le modèle correspondent à la simulation d'une observation spatiale par le couple de satellites Pléiades. Les objets observés par le modèle sont de deux types, l'observation d'un environnement forestier (Figure 5.3) et le même environnement avec un palmier isolé centré sur l'image (Figure 5.4).

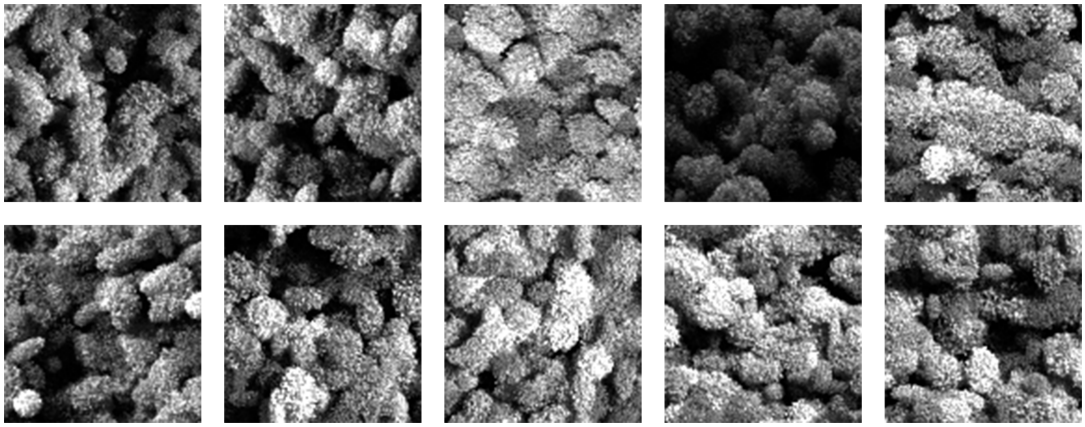


FIGURE 5.3 – Images DART simulées représentant une section de forêt capturée par satellite.

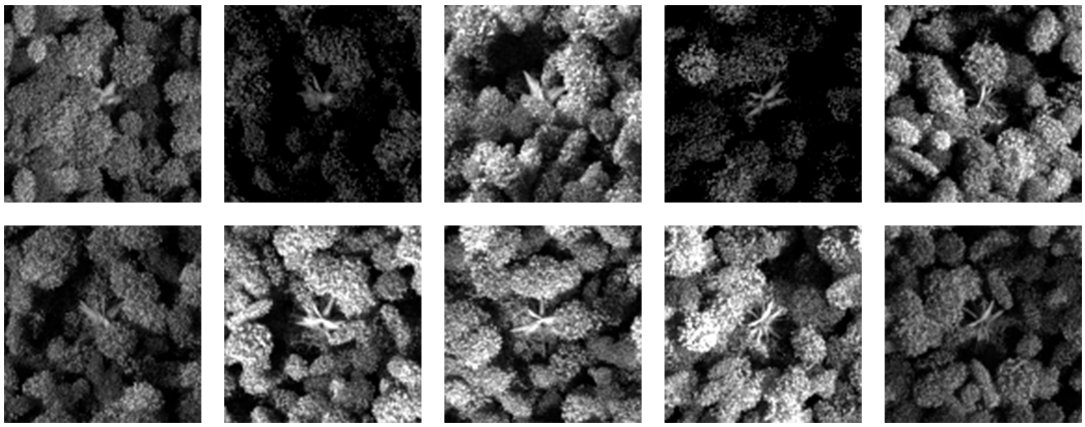


FIGURE 5.4 – Images DART simulées représentant un palmier au milieu d'une section de forêt capturée par satellite.

Les arbres forestiers et les palmiers sont représentés artificiellement dans le modèle DART selon plusieurs paramètres qui simulent différents types de croissance pour refléter au mieux la réalité. Comme le modèle prend en compte la canopée dans son ensemble, certains babaçus seront entièrement ou partiellement cachés par des arbres. Les images sont simulées sous différents angles d'observation et pour différentes incidences de la lumière du soleil (différentes heures de la journée). Les heures UTC choisies sont : 12h, 13h et 14h.

En adaptant la paramétrisation des modèles à l'image panchromatique des images Pléiades, il a été possible de créer plus de de 140 000 images de babaçu dans la forêt et 140 000 images de forêt sans babaçu qui sont disponibles à une résolution de $[106 \times 106]$ pixels. Pour notre étude c'est un total de 50 000 images de chaque qui sera utilisé.

Les images générées par DART initialement étaient plus nettes que les images satellites Pléiades. L'hypothèse est que le flou des images satellites serait dû au bruit du capteur (lentilles, photodiodes, etc.), qui normalement devrait être corrigé. Pour les images DART, le signal généré est simulé avant l'entrée dans le capteur lui-même et n'a pas cette caractéristique. Afin de mieux coller visuellement et spectralement (histogramme des canaux) aux données satellites, un filtre gaussien est appliqué pour améliorer la texture de l'image simulée.

Enfin, les images DART possèdent une radiométrie plus sombre que les images satellites. La plus grande différence vient des ombres, plus noir pour les images simulées que pour les images satellites. Cette différence radiométrique n'est pas un problème pour l'architecture de Deep Learning DenseNet utilisé. L'architecture utilise des couches de « Batch-normalization » qui permet de normaliser la donnée au fur et à mesure des couches de convolution.

5.2.3 Gestion des données

À partir des deux images Pléiades, un étiquetage par photo-interprétation des palmiers de la région est réalisé. La haute résolution du satellite et la forme particulière des couronnes du palmier, facilement reconnaissables dans la canopée, rendent cette identification possible. On peut distinguer trois cas :

- le premier cas concerne les zones de faible densité (palmiers isolés ou en petit nombre), où le comptage et l'étiquetage du centre de chaque couronne de palmier sont possibles (Figure 5.5a) ;
- le second cas concerne les zones délimitées sans babaçus, où une simple délimitation est nécessaire (Figure 5.5b) ;
- le dernier cas est les zones de forte densité (palmiers se recouvrant les uns les autres), où le comptage n'est pas possible par photo-interprétation, une délimitation par zone est effectuée à la place (Figure 5.5c).

Entre les images utilisées pour constituer le jeu d'apprentissage et le jeu de test, les zones d'échantillonnage sélectionnées sont spatialement indépendantes.

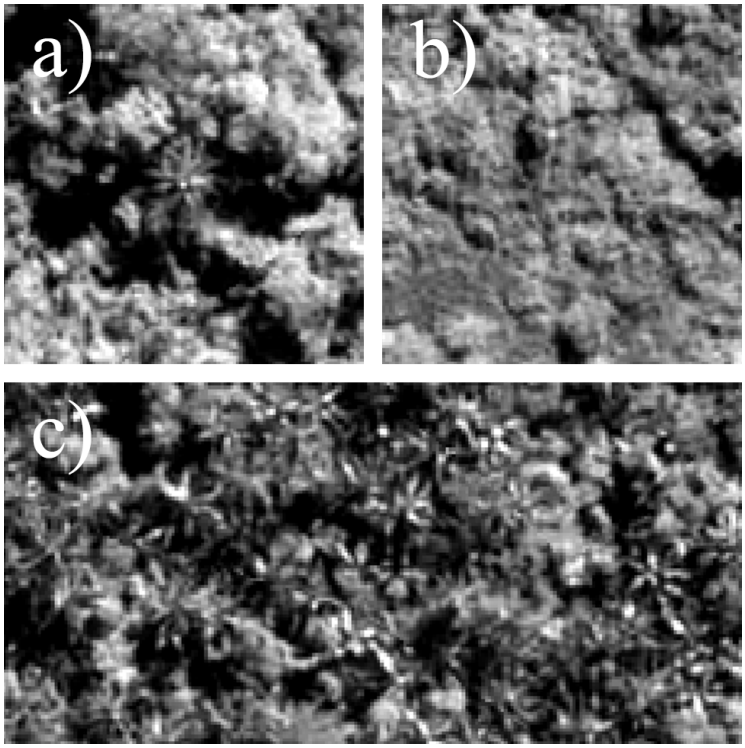


FIGURE 5.5 – Illustration sur trois extraits d'images Pléiades : d'un pointage possible d'un babaçu au milieu d'une forêt secondaire jeune (a), d'une forêt secondaire jeune sans babaçu (b) et d'une forêt secondaire à forte densité de babaçu où la distinction entre les différents individus est difficile par télédétection (c).

Création d'un jeu de données de contrôle

Dans le cas des zones de faibles densités, le pointage a été fait manuellement comme illustré Figure 5.6. C'est un total de 500 points identifiés comme babaçu et 500 points sans babaçu qui sont ainsi référencés pour la constitution d'un jeu d'apprentissage. Pour chaque point, c'est une image de $[48 \times 48]$ autour du pixel central qui sont extraites de l'image satellite Pléiades.

De la même façon, sur une zone spatialement indépendante de ces premiers points, c'est de nouveau 500 points de babaçu et 500 points sans babaçu qui sont référencés pour créer un jeu de données de test. C'est donc un total de 2000 points, dont la moitié indique le centre de la couronne d'un palmier, qui ont été ainsi référencés.

À ces images s'ajouteront les données simulées issues du modèle DART. Les données DART, de mêmes résolutions que l'imagerie satellite, seront elles aussi découpées autour de son pixel central pour correspondre aux dimensions de la base de données. Nous passerons donc d'une image de $[106 \times 106]$ à une image de $[48 \times 48]$.

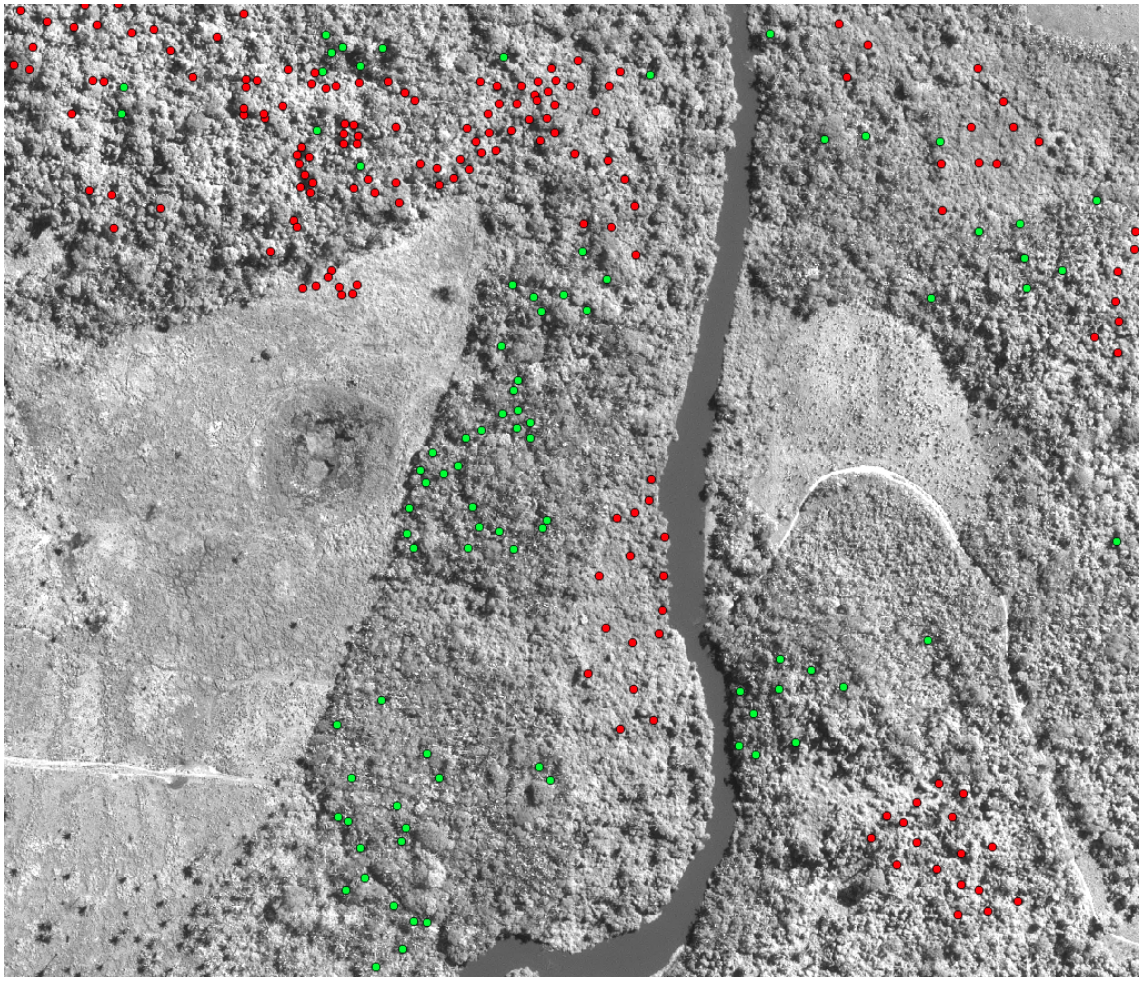


FIGURE 5.6 – Image Pléiades où le centre de babaçus (en vert) et des zones sans babaçus (en rouge) sont pointés.

C'est à partir de ce jeu de données de référence, mise en forme sous un format hdf5, que la comparaison entre les architectures de Deep Learning sera réalisée et que l'apport des données simulées sera démontré.

Création d'un jeu de données étendu

La littérature le démontre, pour faire fonctionner correctement une architecture de Deep Learning, il est nécessaire de disposer de données (d'entraînement et de vérification) en nombre suffisant [197]. Dans le cadre de cette étude, il s'est révélé nécessaire de compléter le jeu de données. En effet, pour appliquer les méthodes de Deep Learning pour la détection des babaçus dans l'image satellite une fenêtre glissante de taille $[48 \times 48]$ pixels avec un pas de 24 pixels (décalage entre 2 appositions de fenêtre) est utilisée. Celle-ci est appliquée sur les zones de l'image présentant

des babaçus. De la même manière, la même fenêtre glissante est appliquée sur des sous-zones, exempts de tous babaçus, pour compléter l'ensemble des données. Ainsi, si le modèle détecte un babaçu au centre de la fenêtre, alors le résultat sera positif, autrement il sera négatif.

Or, si dans le jeu de données initial, le modèle permet d'identifier les palmiers isolés, il exclut les zones à forte densité de babaçus. Ainsi, le jeu de données d'entraînement relativement faible (500 palmiers repérés par expertise), avec les caractéristiques de cette fenêtre, engendre une sous-représentation des palmiers individuels. En conséquence, un décalage de 5 et 10 pixels par rapport au pixel central des images présentant un babaçu isolé est effectué, permettant un ajout de 24 images supplémentaires de $[48 \times 48]$ pixels à l'ensemble des données. Si le babaçu est excentré par rapport à la fenêtre de détection alors il ne sera pas comptabilisé par le modèle. Ce jeu de données étendu permettra alors d'améliorer sensiblement la détection des babaçus sur une scène.

5.2.4 Évaluation des modèles de Deep Learning

Deux architectures de Deep Learning sont utilisées/testées dans cette étude, AlexNet et DenseNet. L'architecture AlexNet doit être adaptée ici pour accepter les images en entrée. Car pour rappel, au fur et à mesure de l'encodage de l'information, la résolution de l'image est divisée par deux à chaque couche de convolution. Comme l'image ne peut être plus petite qu'un pixel (limitation physique), il est nécessaire de, soit augmenter la taille de l'image en entrée, soit changer les couches de l'architecture pour correspondre aux données. Nous nous retrouvons donc avec les paramètres de la Table 5.1 pour l'architecture AlexNet. La sortie de l'architecture est une sortie binaire, reflétant l'absence ou la présence d'un babaçu au centre de l'image.

Tableau 5.1 – Configuration des couches de l'architecture AlexNet pour la détection de babaçu

Couche	Taille du noyau	Nombre de filtres	stride	padding
Convolution 1	5×5	64	1	2
Convolution 2	5×5	192	1	2
Convolution 3	3×3	384	1	1
Convolution 4	3×3	256	1	1
Convolution 5	3×3	256	1	1
Max Pooling	3×3	-	2	-

L'architecture DenseNet est une architecture qui a montré d'excellentes performances dans l'étude précédente (Chapitre 3) et n'a donc pas nécessité de modifications. Une fois les modèles entraînés et la sortie des modèles obtenus, leurs performances seront évaluées par matrice de confusion. En effet, à partir de cette matrice quatre indicateurs de performances seront calculés, la précision globale, la précision, le rappel et le F1-score. La matrice de confusion est un croisement entre les valeurs observées et les valeurs prédites permettant d'évaluer le taux de « vrais positifs » et de « vrais négatifs » (ou l'opposé). Elle est constituée de la façon suivante :

Tableau 5.2 – Exemple d'une matrice de confusion standard

		Valeurs Observées	
		Positive	Négative
Valeurs Prédites	Positive	VP (Vrai Positif)	FP (Faux Positif)
	Négative	FN (Faux Négatif)	VN (Vrai Négatif)

À partir de cette matrice, il existe plusieurs équations applicables permettant d'évaluer les résultats du modèle. Nous définissons les équations suivantes :

$$\text{Précision globale} = \frac{VP + VN}{VP + VN + FP + FN} \tag{5.1}$$

Ces différents indicateurs permettent de mesurer plusieurs aspects d'un modèle. La précision globale ne suffit pas à elle seule pour déterminer le comportement d'un modèle. Comme on peut le constater dans les équations ci-après, la précision et le rappel incluent deux paramètres très différents dans leur calcul. La précision prend en compte les faux positifs, c'est-à-dire les éléments détectés comme vrais par le modèle, mais qui ne sont pas corrects en réalité.

$$\text{Précision} = \frac{VP}{VP + FP} \tag{5.2}$$

$$\text{Rappel} = \frac{VP}{VP + FN} \tag{5.3}$$

Le rappel quant à lui considère les faux négatifs, c'est-à-dire les éléments classés comme faux par le modèle, mais qui s'avère être juste en réalité. Comme pour la précision utilisateur et la précision du producteur vu dans les chapitres précédents (Chapitres 3 et 4), ces deux paramètres sont importants à prendre en compte pour avoir une bonne vision des performances du modèle.

$$\text{F1-score} = \frac{2 \times \text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} \tag{5.4}$$

Dans d'autres problématiques, comme en médecine, il est parfois préférable de favoriser un paramètre par rapport à l'autre. En effet, lors d'un test médical par exemple, il est moins dangereux de détecter une personne comme « malade », alors qu'elle ne l'est pas (faux positif), auquel cas d'autres tests plus poussés pourront révéler l'erreur. A contrario, il est plus dangereux de laisser partir un patient, grâce au résultat négatif au test, alors qu'il était réellement malade (faux négatif). Dans ce cas le rappel sera privilégié sur la précision. Dans notre étude, où la précision et le rappel sont importants, le F1-score présentant une métrique prenant en compte ces deux paramètres, il sera privilégié pour mesurer les modèles.

5.3 Analyse et performances des modèles

Dans un premier temps, les architectures de Deep Learning seront comparées avec le jeu de données restreint ne contenant que les données satellites. Dans un second temps les données simulées seront progressivement ajoutées au jeu de données pour déterminer leur influence sur l'apprentissage. Enfin, avec le jeu de données étendu, l'application de l'architecture DenseNet sera faite pour une détection sur une image Pléiades des babaçus.

5.3.1 Comparaison des architectures de Deep Learning

À partir d'un jeu de données ne contenant que les données de photo-interprétation des babaçus isolés identifiés par pointage, les deux architectures de Deep Learning, AlexNet et DenseNet, sont entraînées. Les modèles seront entraînés successivement avec un jeu d'apprentissage contenant 100, 200, 300, 400 et enfin 500 individus de babaçu. Le résultat de ces modèles sera alors comparé dans une table de confusion des 500 individus de babaçu constituant le jeu de test. Pour rappel, à chaque fois, le jeu de données est constitué de moitié d'images contenant un babaçu, et de l'autre moitié sans. Pour conserver une balance des classes, c'est donc autant d'images sans babaçu qui sont introduites. Par exemple pour 500 individus de babaçu, ce sont 1000 images au total qui sont fournies en apprentissage au modèle. De même pour le jeu de test.

Les résultats sont présentés table 5.3, où les 4 indices ont été calculés : la Précision Globale (PG), la Précision (Pre), le Rappel (Rap) et le F1-score (F1s).

Tableau 5.3 – Résultats des architectures DenseNet et AlexNet pour la détection du babaçu sur le jeu de données constitué uniquement d'images photo-interprétées.

	AlexNet				**DenseNet**			
nb. d'images	**PG**	**Pre**	**Rap**	**F1s**	**PG**	**Pre**	**Rap**	**F1s**
200	66,8	63,76	77,83	70,1	68,53	71,42	61,8	66,26
400	71,1	70,35	72,93	71,62	75,19	64,67	69,53	69,53
600	64	59,55	87,33	70,81	72,78	75,61	67,27	71,19
800	**78,8**	70,04	**79,7**	**74,56**	73,45	77,49	66,1	71,34
1000	72,68	74,14	69,67	71,83	**75,48**	76,89	**72,87**	**74,82**

Dans la figure 5.7, le F1-score des résultats de la table 5.3 des deux architectures de Deep Learning a été rapporté. Il est à remarquer que l'architecture d'AlexNet, moins complexe (possédant moins de paramètres dans son architecture interne) que DenseNet, parvient à maintenir un F1-score proche de 72%, même avec très peu de données en entrée. À l'inverse, l'architecture DenseNet gagne en performance plus le nombre d'entrées est importante. Fonctionnement normal et attendu des architectures de Deep Learning qui ont besoin normalement d'énormément de données pour fonctionner.

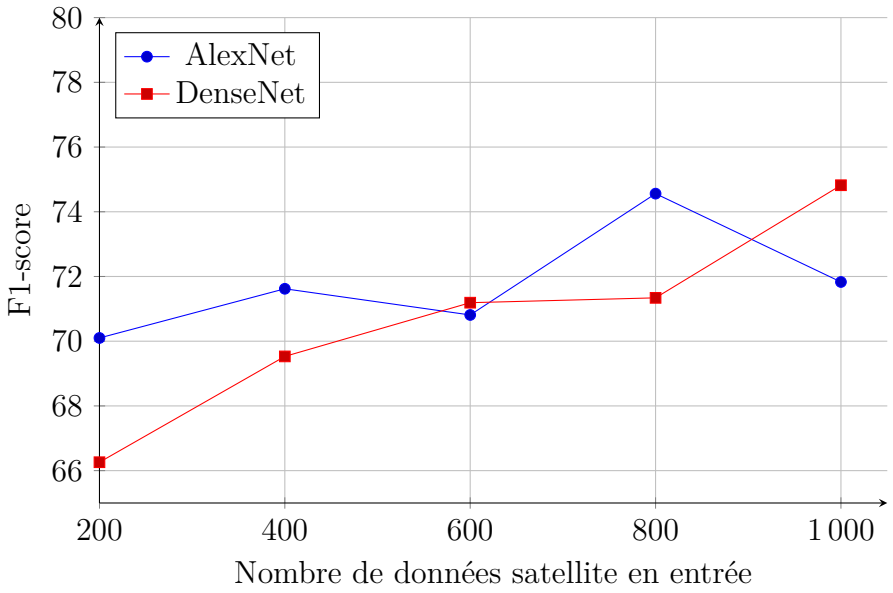


FIGURE 5.7 – Comparaison du F1-score en fonction du nombre d'images en entrée des architectures AlexNet et DenseNet pour la tâche de détection du babaçu.

Il est à noter que de nombreuses séries d'apprentissages ont dû être effectuées par chaque architecture, car à cause du trop faible nombre de données, très souvent les modèles ne parvenaient pas à converger.

Au final, même avec très peu de données, la meilleure architecture de Deep Learning est l'architecture DenseNet, qui parvient avec un jeu de données de 1000 individus à 75, 48 % de précision globale, avec une précision, un rappel et un F1-score de même performance. Même si AlexNet affiche des performances égales à DenseNet, de grandes variations de son résultat ont été remarquées entre plusieurs apprentissages. L'architecture DenseNet présentant des résultats plus stables comparé à AlexNet, c'est elle qui a été choisie pour le reste de l'étude.

5.3.2 Apport des données simulées à l'apprentissage machine

Par la suite, 3 jeux de données sont utilisés ; un jeu de données ne contenant que des images simulées DART ; un deuxième jeu de données contenant 200 images satellites Pléiades en plus des données DART ; et un dernier jeu de données contenant 1000 images Pléiades en plus des données DART. Les données DART seront ajoutées progressivement à ces jeux de données pour étudier leur impact sur l'apprentissage. Comme pour l'étude précédente, les modèles seront comparés sur le jeu de données de test constitué de 1000 images satellites Pléiades (500 individus de babaçu).

Pour un esprit de représentation graphique, on considère pour la table 5.4 et la figure 5.8 que le nombre d'individus indiqués représente en réalité le nombre d'individus par classe (la classe babaçu et la classe sans babaçu). La table 5.4 représente donc le résultat des 4 indices sur les tables de confusion obtenu pour les 3 types de jeux de données. Et la figure 5.8, le F1-score de cette table en fonction du nombre de données simulées pour les 3 jeux de données.

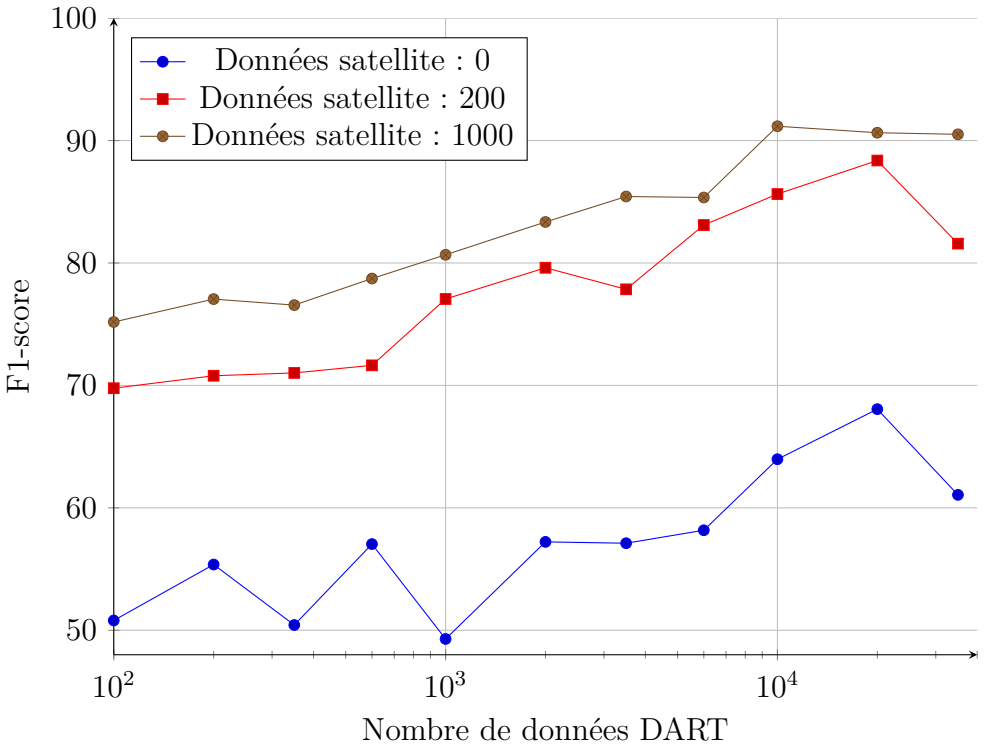


FIGURE 5.8 – Graphique de comparaison sur une échelle logarithmique du F1-score de l'architecture DenseNet en fonction du nombre d'entrées DART pour 3 jeux de données différents.

TABLEAU 5.4 – Résultats de l'architecture DenseNet pour la tâche de détection du babaçu pour 3 jeux de données constitués de différentes quantités d'images DART.

	0 images Pléiades				**200 images Pléiades**				**1.000 images Pléiades**			
Nb. d'images DART	**PG**	**Pre**	**Rap**	**F1s**	**PG**	**Pre**	**Rap**	**F1s**	**PG**	**Pre**	**Rap**	**F1s**
100	49,52	49,54	52,13	50,8	70,87	72,5	67,23	69,77	75,88	77,45	73,03	75,18
200	54,52	54,35	56,43	55,37	71,2	71,81	69,8	70,79	77,57	78,86	75,33	77,05
350	51,42	51,48	49,43	50,43	72,37	74,65	67,73	71,02	76,83	77,47	75,67	76,56
600	53,78	53,29	61,37	57,04	73,3	76,4	67,43	71,64	78,83	79,12	78,33	78,73
1.000	50,03	50,03	48,57	49,29	76,85	76,37	77,77	77,06	80,86	80,73	80,6	80,67
2.000	53,4	52,88	62,33	57,22	78,47	75,6	84,07	79,61	83,08	82,07	84,67	83,35
3.500	59,7	61,03	53,67	57,11	76,88	74,73	81,23	77,85	85,62	86,53	84,37	85,43
6.000	55,78	55,19	61,5	58,17	80,83	74,3	94,27	83,1	84,92	82,95	87,9	85,35
10.000	60,17	58,39	70,73	63,97	85,45	74,59	86,7	85,63	90,58	88,09	94,47	91,17
20.000	64,29	61,56	76,08	68,06	88,88	92,68	84,43	88,37	90,35	88,01	93,43	90,64
35.000	61,78	62,38	58,9	61,06	78,13	70,47	96,87	81,58	90,24	88,03	93,14	90,51

La première chose à noter est que les performances initiales de chaque modèle sont liées directement au nombre d'images satellites dont elle dispose. L'hypothèse est que le jeu de données DART n'arrive pas complètement à recréer la réalité et toutes les spécificités d'une image prise par le capteur Pléiades. En introduisant des données de terrain, les modèles apprennent les variations inhérentes au capteur, difficilement productible avec des données simulées, permettant ainsi une meilleure classification. Cette observation est en adéquation avec les conclusions tirées de la figure 5.7.

Les performances (F1-score) des modèles qui exploitent uniquement des données simulées sont comprises entre 50 % (apprentissage avec 100 images DART, 1 chance sur deux : prédiction aléatoire), et 68, 06 % (apprentissage avec 2000 images DART). C'est donc un gain de 18 % qui est observé.

Une tendance similaire des trois courbes se dégage. Plus le nombre de données simulées augmente, et plus les performances des modèles augmentent. Les meilleures performances sont obtenues pour 20 000 données DART ajoutées. Au-delà de cette valeur, nous observons une chute de performance. L'hypothèse derrière cette chute de performance provient du fait que le nombre de données simulées devient trop important par rapport aux données satellites, les « noyant » dans l'apprentissage.

Si on observe ce « décrochage » pour le jeu de données ne contenant aucune donnée DART et celui en contenant une centaine, ce n'est pas le cas pour le dernier jeu de données qui est plus sur une asymptote. Cependant le graphique s'arrêtant à 35 000 données DART, on ne peut pas affirmer que ce modèle ne « décrochera » pas par la suite. Il existe cependant une valeur optimale du ratio entre les données satellites et les données simulées.

Grâce à l'ajout de données DART, on observe un net gain des performances globales sur tous les indices de 15 %. Avec le peu de données disponibles, nous passons d'une précision globale de 78, 8 % à 90, 58 % pour le meilleur modèle, celui incluant les 1000 images Pléiades supplémentaires. En résumé, l'ajout de données simulées aux modèles de Deep Learning améliore significativement les résultats et cela de manière linéaire sur une échelle logarithmique jusqu'à un optimum situé aux alentours de 20 000 images DART. L'ajout de données simulées après cet optimum entraîne une perte de performances.

5.3.3 Détection du babaçu à partir d'une image satellite

Une des informations importantes de cette étude consiste à montrer que le jeu de données de pointage précédentes n'est pas suffisant pour l'obtention d'un modèle permettant de détecter des babaçus dans une image satellite comme Pléiades à résolution spatiale submétrique. Par ailleurs, comme vu dans la Figure 5.5, le cas où plusieurs palmiers se chevauchent n'a pas été résolu. Pour le jeu de données contrôlé, cette occurrence n'apparaît pas et est donc complètement ignorée par le modèle de Deep Learning.

Le jeu de données doit donc être enrichi de données supplémentaires. L'architecture DenseNet est donc entraînée sur le jeu de données étendu puis appliqué à l'image. Des données DART ont été utilisées pour la création de ce modèle. Sans elles, le modèle dysfonctionnait et produisait de nombreux faux-positifs. Avec l'ajout des données Pléiades, nous obtenons l'extrait de l'image Pléiades ainsi que le résultat de l'application du modèle DenseNet présenté dans la Figure 5.9.

Les tâches en vert sur l'image 5.9 correspondent aux zones que l'algorithme juge comme étant les plus proches d'une détection de babaçu. En effet, seuls les points dont le modèle est sûr à 98 % sont montrés. Cette détection peut être modifiée en changeant le seuil de détection. Même pour un opérateur humain il est difficile de distinguer les différents babaçus, en particulier pour les palmiers dont la couronne est occultée partiellement par l'ombre des arbres qui les dominent. De plus, des résultats quantitatifs ne sont pas possibles sans l'aide d'un expert effectuant une photo-interprétation ou sans l'aide d'image de plus haute résolution.

Sur les résultats cartographiques, on constate visuellement de nombreux faux positifs, surtout à l'ombre des arbres. Nous constatons aussi que pour les fortes densités de palmiers (zone en bas à gauche de l'image), le modèle marque de grandes zones. Ce marquage permet bel et bien de distinguer plusieurs palmiers, mais ne permet pas d'effectuer un comptage des palmiers. Un jeu de données enrichi semble nécessaire pour améliorer les performances de détection des modèles neuronaux denses.

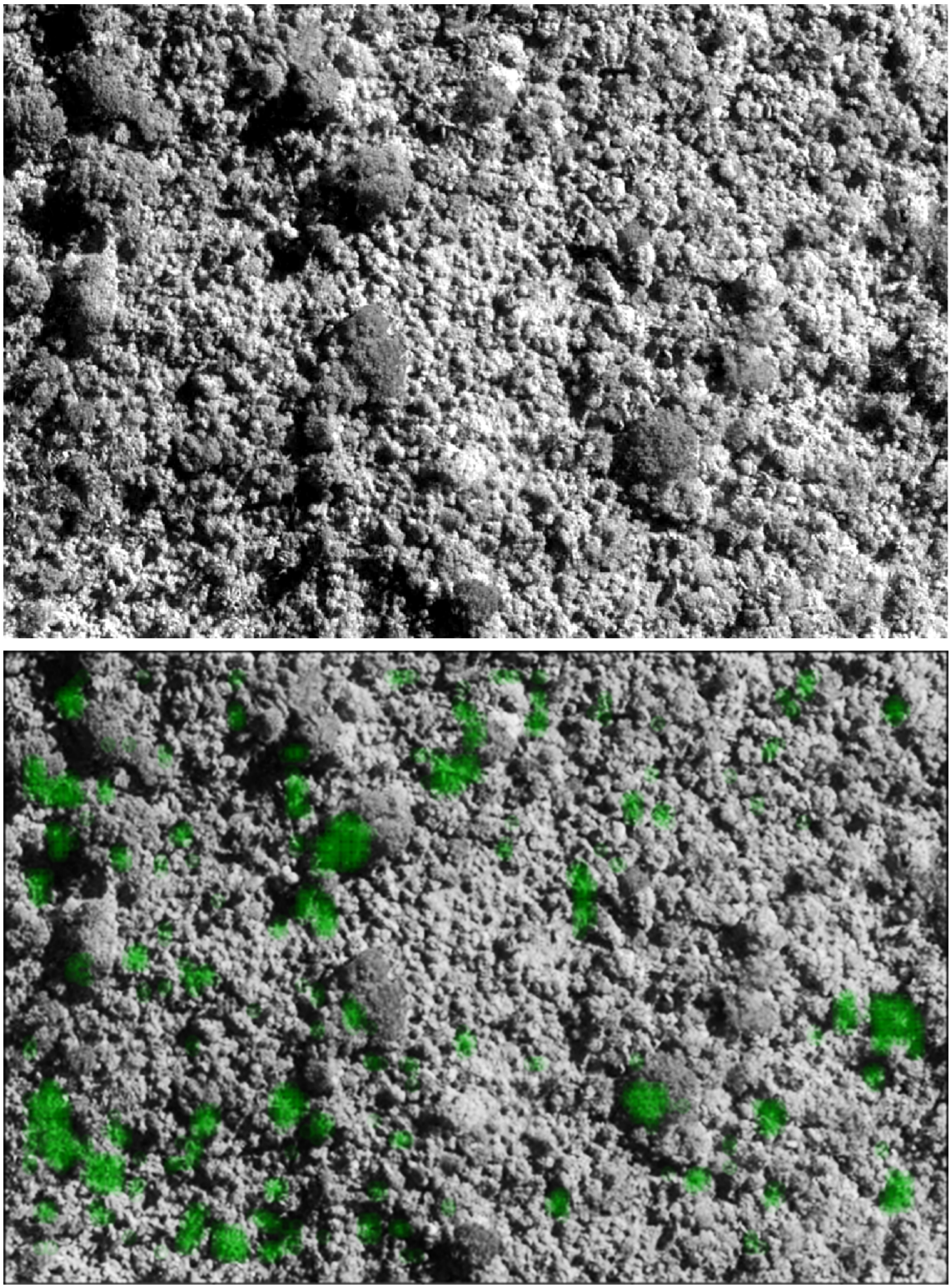


FIGURE 5.9 – Extrait d'une image Pléiades (haut) sur laquelle un modèle de Deep Learning, DenseNet, a été appliqué pour détecter le palmier babaçu (bas).

5.4 Conclusions

Une étude antérieure à nos travaux permettait d'effectuer un comptage du babaçu dans les pâturages, mais n'était pas adaptée pour les détecter dans des milieux forestiers complexes. Dans ce chapitre, des techniques de Deep Learning ont été développées pour atteindre cet objectif. En exploitant des données entièrement simulées DART, nous avons constaté dans cette étude une amélioration très significative des performances de détection des babaçus (90, 58 % en PG) sur des bases de tests indépendantes de la base d'apprentissage. L'apport des données DART s'est avéré être déterminante pour aboutir à de tels résultats : un gain général de 15 % a été en effet observé en exploitant ces données.

Les données simulées offrent donc une solution intéressante au manque de données collectées sur le terrain pour certains types de problèmes de télédétection. Cette méthode permet en effet de créer des ensembles de données contrôlées et fiables pour alimenter de manière automatique et systématique une architecture complexe de Deep Learning. Cependant, comme nous avons pu le constater dans notre étude, l'utilisation de données simulées ne permet pas de s'affranchir des données de terrain. Un net gain des performances (environ 25 % en PG) est constaté lorsqu'un petit nombre d'entre elles sont ajoutées en apprentissage.

Le modèle de Deep Learning, tel qu'il est présenté dans ces travaux, nécessite des améliorations afin d'être pleinement opérationnel et automatique. Comme discuté précédemment, l'application du modèle de Deep Learning à des images Pléiades ne permet pas d'obtenir un comptage précis, car la technique fournit uniquement une carte de probabilités de présence. Le modèle n'est pas entraîné pour faire la distinction entre plusieurs palmiers lorsque la densité de babaçus est trop importante. Des données complémentaires, simulées ou de terrain, intégrant plusieurs babaçus sur une même image avec les coordonnées de chacun d'eux doivent être récoltées afin de pouvoir atteindre cet objectif.

CHAPITRE 6

CONCLUSIONS ET PERSPECTIVES

"Il est une chose dont nous avons maintenant la certitude : les robots changent la face du monde et nous mènent vers un avenir que nous ne pouvons encore clairement définir."

Isaac Asimov

Sommaire

6.1 Une cartographie de l'occupation du sol automatisée . . 155

6.2 Une détection automatique d'espèces végétales avec des données simulées 159

6.1 Une cartographie de l'occupation du sol automatisée

La première partie de ce mémoire de thèse porte sur la classification des types de sols (sa couverture biogéophysique et son usage) par le biais de plusieurs architectures de Deep Learning. Un jeu d'entraînement et de contrôle a été construit spécifiquement pour ces travaux de recherche située en province Sud, en Nouvelle-Calédonie. Ces données sont mises à disposition de la communauté scientifique et pourront servir de benchmark aux chercheurs désirant construire de nouvelles méthodes de détection de couverture et d'usage des sols en milieu tropical insulaire. Si les techniques de machine learning parviennent à obtenir des résultats comparables sur des problématiques simples, les résultats de nos travaux de recherches indiquent que les techniques de Deep Learning affichent de meilleures performances (jusqu'à 10% de précision en plus) sur des problématiques plus complexes comme la classification de l'usage des sols. Une des forces de ces techniques d'apprentissage profond est leur capacité à extraire l'information essentielle dans les données brutes disponibles. Dans le cadre des architectures du type CNN, cette extraction des caractéristiques utiles à la détection dans l'image, s'opère dans la partie « encodeur » de l'architecture. Ceci représente une propriété très utile et innovante dans une communauté qui à l'habitude d'élaborer des cartes à partir de combinaisons des canaux bruts (néocanaux, textures...). Les techniques de Deep Learning possèdent en effet une capacité d'abstraction et d'extraction d'information de la donnée intégrée, là où les techniques de machine learning plus classique comme XGBoost, sont beaucoup plus dépendantes de la donnée d'entrée.

Cependant, les techniques d'apprentissage profond restent fortement dépendant des données d'apprentissage, et s'il est possible d'obtenir d'excellents résultats en ce qui concerne la classification de la couverture et de l'usage du sol, il est nécessaire pour cela de disposer d'un grand nombre de données de terrain fiables. Le manque de donnée constitue donc un frein majeur à une utilisation systématique de ces techniques dans la communauté des géomaticiens.

L'étude s'est appuyée dans un premier temps sur des données très haute résolution SPOT 6 puis a été réalisée à partir de données satellites libres d'accès, Sentinel-2, disponibles sur plusieurs années (depuis 2016). Si la chaîne de traitement automatique a été implémentée, il a été difficile dans le cadre de ses travaux, de quantifier la précision des modèles et il semble encore prématuré d'exploiter cette détection pour une surveillance automatique des lisières de forêt par exemple.

Hormis la mise au point d'un jeu de données sur plusieurs années destiné à évaluer précisément la qualité des modèles, il existe de nombreuses autres améliorations qui peuvent être apportées afin de proposer un produit fonctionnel et prêt à l'usage pour des décideurs sur des problématiques d'occupation du sol. En effet, le but des études des Chapitres 3 et 4 étaient de déterminer le potentiel et les limites des réseaux de neurones denses à détecter les couches de couverture et d'usage des sols avec uniquement des données brutes des satellites en entrée. Aucun prétraitement, à part celui permettant de mettre en forme les jeux de données, et aucun post-traitement n'ont été effectués dans ces études. Il est à noter que l'application simple d'un « tamis » par exemple, permettant de retirer les pixels isolés et unifier les ensembles de pixels entre eux, permet d'améliorer sensiblement les résultats (1-2%). Il est à noter également que seuls les canaux les plus résolus des données Sentinel-2 ont été utilisés, sans exploiter tout le potentiel que propose ce capteur au travers de sa gamme de 13 bandes spectrales.

Une approche hiérarchique de la typographie (nomenclature de la couverture et de l'usage du sol) permettrait de mieux contrôler chaque étape de l'algorithme. C'est-à-dire qu'au lieu de classer l'ensemble de l'image en une seule itération, l'architecture pourrait produire une classification binaire de l'ensemble, par exemple : végétation ; non-végétation, en affinant au fur et à mesure les classes. Cela permettrait de diminuer le degré de complexité induit par le nombre de classes qu'une architecture doit prendre en considération dans sa classification, et l'enchaînement de classification binaire permettrait de diminuer progressivement l'espace des possibilités de chaque classe. De plus certaines de ces classifications binaires ne demanderaient pas systématiquement l'intervention d'une architecture de Deep Learning.

Comme vu dans les résultats précédents, les méthodes de machine learning classique parviennent à des performances équivalentes à l'apprentissage profond lorsque la tâche de classification est simple. Le potentiel des architectures de Deep Learning pourrait alors être utilisé pour les parties les plus critiques de la détection demandant une compréhension du contexte et du voisinage spatial et temporel du pixel à classifier.

L'application de modèles successifs permettrait aussi d'associer des données exogènes et opportunes pour chaque classification. En effet, des données exogènes simples comme le MNT, le trait de côte ou encore des données faisant état de l'altimétrie du terrain en général représentent une donnée essentielle très souvent exploitée en télédétection. Ces données exogènes sont essentielles pour séparer l'espace de détection en sous-espaces plus facile à isoler. Par exemple, dans le cas de la

Nouvelle-Calédonie, il n'existe pas de surface en eau au-delà d'une certaine altimétrie.

L'élaboration d'une architecture de réseaux de neurones dense du type CNN résulte souvent d'un compromis entre la taille des images fournies en entrée, la taille du lot d'images (« batch ») fournies lors du processus d'apprentissage et la capacité de mémoire et de calcul de la carte graphique. Pourtant ces tailles d'images et de lots influent fortement sur les performances et la capacité de discrimination d'un réseau de neurones. Afin de limiter les problèmes de détection dus à un contexte spatial réduit (il est difficile par exemple de distinguer un sol nu d'un chantier des mines sans considérer un contexte spatial large) et de rendre l'architecture plus robuste, l'idéal serait de pouvoir alimenter les modèles avec de très grandes zones et cela sans limiter la taille des lots. Malheureusement augmenter la taille de la fenêtre coulissante limiterait aussi la taille des lots d'images fournies en entrée de l'architecture, et donc imposerait de disposer de plus grandes zones étiquetées.

Les séries temporelles représentent une réelle plus-value pour la détection des types de sols, et n'ont pas été exploitées dans le cadre de ces travaux. En effet pour la détection des ombres et des nuages c'est une technique qui est appliquée pour détecter les changements entre deux images similaires. Les nuages et les ombres générées sont des éléments qui disparaissent d'une prise de vue à une autre, et si des techniques basiques permettent d'obtenir d'excellents résultats, alors l'acquisition d'un masque généré automatiquement est possible [198].

La mise en place d'architectures de Deep Learning, permettant d'intégrer des informations provenant de différents capteurs représente également une perspective intéressante pour la création de cartes de couverture et d'utilisation des sols à des précisions plus grandes et sur des périodes très courtes [199]. Il existe plusieurs architectures permettant cette fusion de données à des niveaux différents des architectures d'apprentissage profond (Figure 6.1 et 6.2). L'application de ces architectures permettrait alors d'utiliser toute la gamme de données disponibles (satellite, drone, terrain, etc.) pour la mise au point de modèles très performants [135].

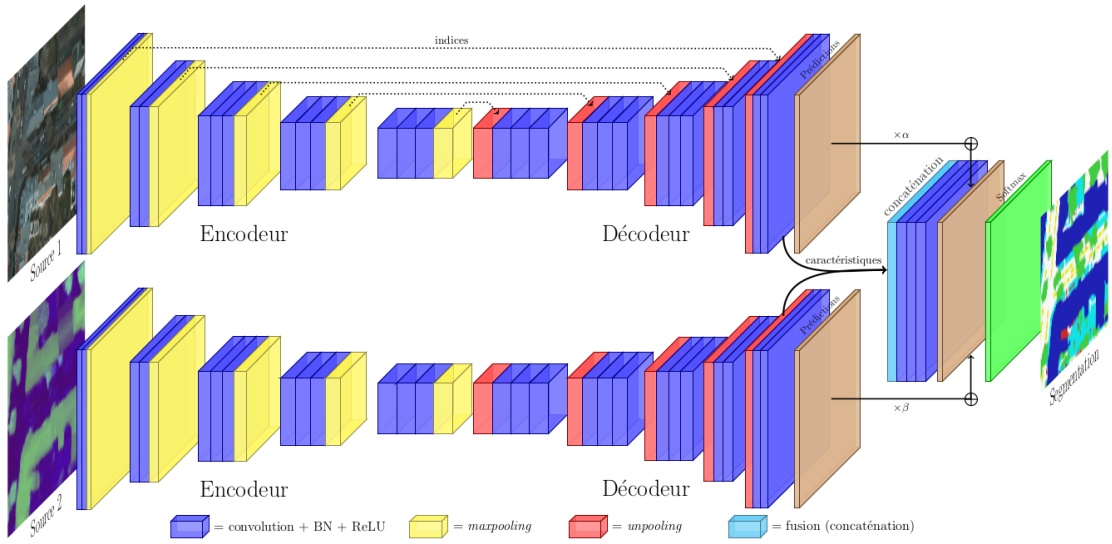


FIGURE 6.1 – Illustration de la fusion des sorties de deux architectures de Deep Learning différentes pour une détection d'images satellites de résolutions spectrales et spatiales différentes. Figure issue d'une présentation de Nicolas Audebert.

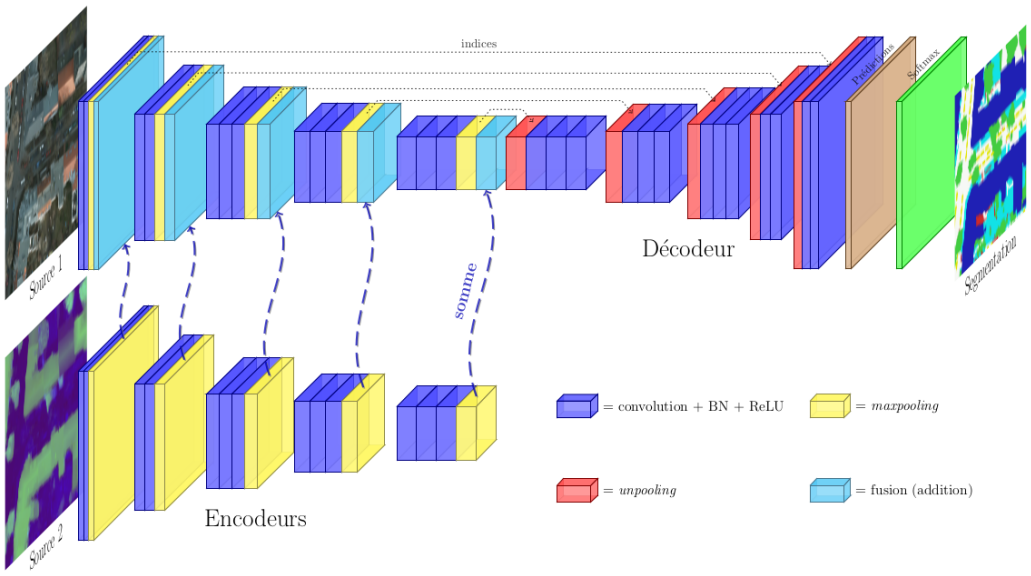


FIGURE 6.2 – [Illustration de la fusion des encodeurs de deux architectures de Deep Learning avant le décodage global de l'information. Figure issue d'une présentation de Nicolas Audebert.

6.2 Une détection automatique d'espèces végétales avec des données simulées

La deuxième partie de ce mémoire de thèse porte sur la détection d'objets et plus particulièrement sur la détection du palmier babaçu au Brésil. L'originalité de la démarche réside dans l'utilisation de données de télédétection simulées numériquement afin d'améliorer l'apprentissage des réseaux de neurones profonds. Ces travaux ont démontré que l'ajout de données simulées améliorait significativement les performances globales du modèle neuronal.

Une étude sur le terrain de relevé de babaçus, par l'analyse d'images drone par exemple, est nécessaire pour lever le doute lié à la quantification des résultats. Notre étude se limite à localiser les babaçus au sein de la forêt à l'aide d'une fenêtre coulissante aboutissant sur la génération d'une carte de probabilité de présence. Les prochaines étapes seraient de définir des cadres autour des palmiers, comme présenté Figure 6.3.

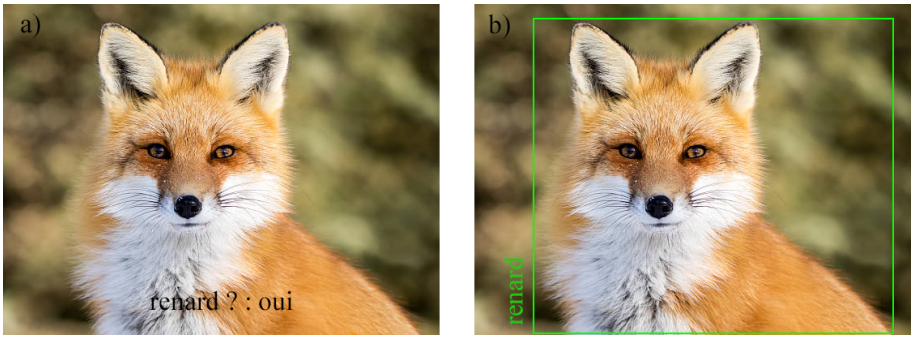


FIGURE 6.3 – Illustration de détection d'un objet dans une image, une méthode identifiant la présence d'un renard ou non dans l'image (a) et une autre dessinant les limites de la zone contenant l'individu (b).

Il serait très intéressant de mener la même étude sur les palmiers du pâturage. Les données simulées existent et il ne manquerait qu'un effort de photo-interprétation pour constituer un jeu de données complet à partir des images satellites existantes. Cette étude permettrait de faire fonctionner l'algorithme sur l'ensemble de la scène. De plus, les résultats de ce modèle sur les pâturages pourraient être comparés avec les algorithmes déjà existants. L'application de données simulées dans ce contexte serait aussi à considérer afin d'atteindre des performances de détection équivalentes ou supérieures à un opérateur humain.

Dans le contexte de la Nouvelle-Calédonie, où certaines zones restent difficiles d'accès et où la récolte de données de terrain peut s'avérer complexe, l'utilisation de

données simulées pourrait s'avérer très utile. De plus, si en effort d'enrichissement des modèles numériques était mené, il serait possible de détecter des espèces menacées avec peu d'individus sur le territoire, par exemple les kaoris.

ANNEXE A

RÉFÉRENCES DES ILLUSTRATIONS

— Chapitre 1 : Illustration extraite de *The architectural tourist 2* de Ian Stuart Campbell, publié par « The Royal Incorporation of Architects in Scotland » en 2019.

— Chapitre 2 : Illustration extraite de *la patente de la machine à écrire* de W. J. Barron en 1897.

— Chapitre 3 : Illustration de Stonehenge libre de droit.

— Chapitre 4 : Dessin de Brandon Starr de 2011. https://brandonstarr.com/visdev--illustration.htmlLien vers le site internet.

— Chapitre 5 : Illustration d'un banc sous un arbre de Torky. Image libre de droit.

— Chapitre 6 : Dessin de Martina Arend de 2020. @martinaarend sur Instagram.

ANNEXE B

RÉFÉRENCES DES IMAGES SATELLITES

Tableau B.1 – Détails des images SPOT 6 utilisées dans l'étude

Nom de l'image satellite	Date d'acquisition
SPOT6_PMS_201306012252059_ORT_1119496701	01/06/2013
SPOT6_PMS_201306292238055_ORT_1119497101	29/06/2013
SPOT6_PMS_201406282239012_ORT_1119498101	28/06/2014
SPOT6_PMS_201406192257028_ORT_1119502101	19/06/2014
SPOT6_PMS_201406192257053_ORT_1119503101	19/06/2014
SPOT6_PMS_201406192258009_ORT_1119504101	19/06/2014
SPOT6_PMS_201406212242046_ORT_1119505101	21/06/2014
SPOT6_PMS_201406262253057_ORT_1119506101	26/06/2014
SPOT6_PMS_201406262254015_ORT_1119507101	26/06/2014
SPOT6_PMS_201406262254031_ORT_1119508101	26/06/2014
SPOT6_PMS_201406282239047_ORT_1119509101	28/06/2014
SPOT6_PMS_201407122232016_ORT_1119525101	12/07/2014
SPOT6_PMS_201407172242046_ORT_1119526101	17/07/2014
SPOT6_PMS_201407172243021_ORT_1119527101	17/07/2014
SPOT6_PMS_201407242239020_ORT_1119528101	24/07/2014
SPOT6_PMS_201407292250014_ORT_1119529101	29/07/2014
SPOT6_PMS_201407292250032_ORT_1119530101	29/07/2014
SPOT6_PMS_201407292250059_ORT_1119531101	29/07/2014
SPOT6_PMS_201407312236004_ORT_1119532101	31/07/2014
SPOT6_PMS_201408122243031_ORT_1119533101	12/08/2014
SPOT6_PMS_201410032243032_ORT_1119534101	03/10/2014
SPOT6_PMS_201406212242031_ORT_1303779101	21/06/2014
SPOT6_PMS_201501222238037_ORT_1303780101	22/01/2015
SPOT6_PMS_201410242231044_ORT_1303781101	24/10/2014
SPOT6_PMS_201410172236006_ORT_1303782101	17/10/2014
SPOT6_PMS_201410172235049_ORT_1303783101	17/10/2014
SPOT6_PMS_201407312235012_ORT_1303784101	31/07/2014
SPOT6_PMS_201411122235040_ORT_1303785101	12/11/2014
SPOT6_PMS_201501222238052_ORT_1303786101	22/01/2015

TABLEAU B.2 – Nom des images Sentinel-2 utilisées

Nom de l'image		
S2A_20160616-232124-461_L2A_T58KDC_D_V1-4_FRE	S2A_20181127-230237-058_L2A_T58KGV_C_V2-0_FRE	S2B_20190802-231229-983_L2A_T58KEB_C_V2-2_FRE
S2A_20160713-230909-458_L2A_T58KEB_D_V1-4_FRE	S2A_20181220-231221-797_L2A_T58KFA_D_V1-9_FRE	S2B_20190802-231231-645_L2A_T58KDB_C_V2-2_FRE
S2A_20160809-230151-045_L2A_T58KGA_D_V1-4_FRE	S2B_20180218-230858-455_L2A_T58KFA_D_V1-5_FRE	S2B_20190822-231206-474_L2A_T58KFC_C_V2-2_FRE
S2A_20160901-231142-457_L2A_T58KGA_D_V1-4_FRE	S2B_20180502-231851-457_L2A_T58KCC_C_V2-0_FRE	S2B_20190822-231217-002_L2A_T58KGB_C_V2-2_FRE
S2A_20160921-231137-462_L2A_T58KDB_D_V1-4_FRE	S2B_20180509-230904-462_L2A_T58KGC_C_V2-0_FRE	S2B_20190822-231235-419_L2A_T58KFA_C_V2-2_FRE
S2A_20160921-231137-462_L2A_T58KEA_D_V1-4_FRE	S2B_20180522-231850-456_L2A_T58KDB_D_V1-8_FRE	S2B_20190908-230232-803_L2A_T58KGA_C_V2-2_FRE
S2A_20160921-231137-462_L2A_T58KEB_D_V1-4_FRE	S2B_20180529-230902-464_L2A_T58KDB_D_V1-8_FRE	S2B_20191120-231213-172_L2A_T58KFB_C_V2-2_FRE
S2A_20160924-232117-456_L2A_T58KDB_D_V1-4_FRE	S2B_20180708-230904-458_L2A_T58KEA_D_V1-8_FRE	S2B_20191210-231209-964_L2A_T58KFB_C_V2-2_FRE
S2A_20160924-232117-456_L2A_T58KDC_D_V1-4_FRE	S2B_20180708-230904-458_L2A_T58KFA_D_V1-8_FRE	S2A_20200324-231211-927_L2A_T58KFB_C_V2-2_FRE
S2A_20160924-232117-456_L2A_T58KEB_D_V1-4_FRE	S2B_20180718-231153-021_L2A_T58KFC_C_V2-0_FRE	S2A_20200324-231215-296_L2A_T58KEB_C_V2-2_FRE
S2A_20161001-231201-853_L2A_T58KFA_D_V1-4_FRE	S2B_20180718-231153-021_L2A_T58KGC_C_V2-0_FRE	S2A_20200324-231229-995_L2A_T58KEA_C_V2-2_FRE
S2A_20161004-231931-751_L2A_T58KDB_D_V1-4_FRE	S2B_20180817-231130-990_L2A_T58KEB_D_V1-8_FRE	S2A_20200410-230229-552_L2A_T58KGA_C_V2-2_FRE
S2A_20161004-231931-751_L2A_T58KDC_D_V1-4_FRE	S2B_20180817-231130-990_L2A_T58KFA_D_V1-8_FRE	S2A_20200510-230217-654_L2A_T58KHB_C_V2-2_FRE
S2A_20161021-231056-905_L2A_T58KFA_D_V1-4_FRE	S2B_20180817-231130-990_L2A_T58KGB_C_V2-0_FRE	S2A_20200510-230221-274_L2A_T58KGB_C_V2-2_FRE
S2A_20161031-230903-459_L2A_T58KFA_D_V1-4_FRE	S2B_20180817-231130-990_L2A_T58KGC_C_V2-0_FRE	S2A_20200520-230222-333_L2A_T58KGB_C_V2-2_FRE
S2A_20161210-230901-462_L2A_T58KEB_D_V1-4_FRE	S2B_20180830-231843-463_L2A_T58KCC_C_V2-0_FRE	S2A_20200523-231232-361_L2A_T58KGA_C_V2-2_FRE
S2A_20170608-231144-464_L2A_T58KFC_D_V1-9_FRE	S2B_20180830-231843-463_L2A_T58KDC_D_V1-8_FRE	S2A_20200625-232212-135_L2A_T58KCC_C_V2-2_FRE
S2A_20170608-231144-464_L2A_T58KGB_D_V1-9_FRE	S2B_20180830-231843-463_L2A_T58KEC_D_V1-8_FRE	S2A_20200629-230250-668_L2A_T58KGV_C_V2-2_FRE
S2A_20170608-231144-464_L2A_T58KGC_D_V1-9_FRE	S2B_20181125-231215-386_L2A_T58KEB_D_V1-9_FRE	S2A_20200705-232156-649_L2A_T58KCD_C_V2-2_FRE
S2A_20170708-231205-853_L2A_T58KDB_D_V1-5_FRE	S2B_20181202-230226-452_L2A_T58KGA_D_V1-9_FRE	S2A_20200705-232207-791_L2A_T58KDC_C_V2-2_FRE
S2A_20170708-231205-853_L2A_T58KEB_D_V1-5_FRE	S2A_20190106-230226-104_L2A_T58KGA_D_V1-9_FRE	S2A_20200705-232211-129_L2A_T58KCC_C_V2-2_FRE
S2A_20170708-231205-853_L2A_T58KFA_D_V1-5_FRE	S2A_20190106-230240-436_L2A_T58KGV_C_V2-0_FRE	S2A_20200719-230236-316_L2A_T58KGA_C_V2-2_FRE
S2A_20170708-231205-853_L2A_T58KGA_D_V1-4_FRE	S2A_20190330-231212-098_L2A_T58KGB_C_V2-0_FRE	S2A_20200811-231207-275_L2A_T58KFC_C_V2-2_FRE
S2A_20170721-232123-458_L2A_T58KCC_D_V1-9_FRE	S2A_20190519-231237-298_L2A_T58KEA_C_V2-1_FRE	S2A_20200811-231210-533_L2A_T58KEC_C_V2-2_FRE
S2A_20170721-232123-458_L2A_T58KDC_D_V1-5_FRE	S2A_20190618-231234-015_L2A_T58KFA_C_V2-2_FRE	S2A_20200811-231221-636_L2A_T58KFB_C_V2-2_FRE
S2A_20170817-231207-853_L2A_T58KFA_D_V1-7_FRE	S2A_20190718-231235-451_L2A_T58KFA_C_V2-2_FRE	S2A_20200811-231225-005_L2A_T58KEB_C_V2-2_FRE
S2B_20170703-230905-463_L2A_T58KGB_D_V1-9_FRE	S2A_20190721-232211-730_L2A_T58KCC_C_V2-2_FRE	S2A_20200811-231232-495_L2A_T58KGA_C_V2-2_FRE
S2B_20170703-231205-852_L2A_T58KEA_D_V1-5_FRE	S2A_20190725-230236-146_L2A_T58KGA_C_V2-2_FRE	S2A_20200811-231236-211_L2A_T58KFA_C_V2-2_FRE
S2B_20170703-231205-852_L2A_T58KEB_D_V1-5_FRE	S2A_20190731-232207-910_L2A_T58KDC_C_V2-2_FRE	S2A_20200818-230236-964_L2A_T58KGA_C_V2-2_FRE
S2B_20170706-232122-463_L2A_T58KDB_D_V1-5_FRE	S2A_20190731-232211-250_L2A_T58KCC_C_V2-2_FRE	S2B_20200319-231217-009_L2A_T58KEB_C_V2-2_FRE
S2B_20170706-232122-463_L2A_T58KDC_D_V1-5_FRE	S2A_20190906-231214-952_L2A_T58KFB_C_V2-2_FRE	S2B_20200528-231231-551_L2A_T58KFA_C_V2-2_FRE
S2B_20170720-230203-463_L2A_T58KGB_D_V1-9_FRE	S2A_20190906-231229-529_L2A_T58KFA_C_V2-2_FRE	S2B_20200627-231159-795_L2A_T58KGC_C_V2-2_FRE
S2B_20170928-230005-457_L2A_T58KFA_D_V1-7_FRE	S2A_20191016-231218-822_L2A_T58KFB_C_V2-2_FRE	S2B_20200627-231206-995_L2A_T58KEC_C_V2-2_FRE
S2B_20170928-230005-457_L2A_T58KGA_D_V1-7_FRE	S2B_20190521-230225-373_L2A_T58KGB_C_V2-1_FRE	S2B_20200627-231211-052_L2A_T58KDC_C_V2-2_FRE
S2B_20170928-230005-457_L2A_T58KGV_D_V1-9_FRE	S2B_20190521-230239-842_L2A_T58KGA_C_V2-1_FRE	S2B_20200627-231214-451_L2A_T58KGB_C_V2-2_FRE
S2A_20180315-230901-463_L2A_T58KFA_D_V1-7_FRE	S2B_20190703-231206-100_L2A_T58KGC_C_V2-2_FRE	S2B_20200627-231221-458_L2A_T58KEB_C_V2-2_FRE
S2A_20180404-230905-456_L2A_T58KEA_D_V1-7_FRE	S2B_20190703-231220-397_L2A_T58KGB_C_V2-2_FRE	S2B_20200627-231224-515_L2A_T58KDB_C_V2-2_FRE
S2A_20180623-230906-463_L2A_T58KEC_D_V1-8_FRE	S2B_20190713-231206-175_L2A_T58KGC_C_V2-2_FRE	S2B_20200717-231158-732_L2A_T58KGC_C_V2-2_FRE
S2A_20180805-231852-462_L2A_T58KCC_C_V2-0_FRE	S2B_20190720-230225-355_L2A_T58KGB_C_V2-2_FRE	S2B_20200717-231231-617_L2A_T58KFA_C_V2-2_FRE
S2A_20180822-230904-461_L2A_T58KFA_D_V1-8_FRE	S2B_20190723-231213-070_L2A_T58KEC_C_V2-2_FRE	S2B_20200806-231218-953_L2A_T58KFB_C_V2-2_FRE
S2A_20181127-230208-250_L2A_T58KGB_C_V2-0_FRE	S2B_20190726-232225-913_L2A_T58KDB_C_V2-2_FRE	S2B_20200806-231237-020_L2A_T58KEA_C_V2-2_FRE
S2A_20181127-230222-736_L2A_T58KGA_D_V1-9_FRE	S2B_20190802-231223-252_L2A_T58KEB_C_V2-2_FRE	S2B_20200819-232206-870_L2A_T58KDC_C_V2-2_FRE
S2A_20181127-230228-669_L2A_T58KFA_D_V1-9_FRE	S2B_20190802-231224-767_L2A_T58KDB_C_V2-2_FRE	S2B_20200819-232221-200_L2A_T58KDB_C_V2-2_FRE

BIBLIOGRAPHIE

[1] J. R. Jensen *et al.*, *Introductory digital image processing : a remote sensing perspective.* Ed. 2, Prentice-Hall Inc., 1996. 15

[2] T. Blaschke, "Object based image analysis for remote sensing," *ISPRS journal of photogrammetry and remote sensing*, vol. 65, no. 1, pp. 2–16, 2010. 15

[3] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geoscience and remote sensing magazine*, vol. 1, no. 2, pp. 6–36, 2013. 15

[4] X. Jia, B.-C. Kuo, and M. M. Crawford, "Feature mining for hyperspectral image classification," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 676–697, 2013. 15

[5] U. C. Benz, P. Hofmann, G. Willhauck, I. Lingenfelder, and M. Heynen, "Multi-resolution, object-oriented fuzzy analysis of remote sensing data for gis-ready information," *ISPRS Journal of photogrammetry and remote sensing*, vol. 58, no. 3-4, pp. 239–258, 2004. 15

[6] T. Blaschke and J. Strobl, "What's wrong with pixels ? some recent developments interfacing remote sensing and gis," *Zeitschrift für Geoinformationssysteme*, vol. 14, pp. 12–17, 2001. 15

[7] V. Walter, "Object-based classification of remote sensing data for change detection," *ISPRS Journal of photogrammetry and remote sensing*, vol. 58, no. 3-4, pp. 225–238, 2004. 15

[8] W. Zhou, A. Troy, and M. Grove, "Object-based land cover classification and change analysis in the baltimore metropolitan area using multitemporal high resolution remote sensing data," *Sensors*, vol. 8, no. 3, pp. 1613–1636, 2008. 15

[9] Q. Yu, P. Gong, N. Clinton, G. Biging, M. Kelly, and D. Schirokauer, "Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery," *Photogrammetric Engineering & Remote Sensing*, vol. 72, no. 7, pp. 799–811, 2006. 15

[10] A. Tzotsos and D. Argialas, "Support vector machine classification for object-based image analysis," in *Object-Based Image Analysis*, pp. 663–677, Springer, 2008. 15

[11] F. J. Gallego, "Remote sensing and land cover area estimation," *International Journal of Remote Sensing*, vol. 25, no. 15, pp. 3019–3047, 2004. 15

[12] J. Liu, J. Li, W. Li, and J. Wu, "Rethinking big data : A review on the data quality and usage issues," *ISPRS journal of photogrammetry and remote sensing*, vol. 115, pp. 134–142, 2016. 16

[13] L. Zhang, L. Zhang, and V. Kumar, "Deep learning for remote sensing data," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22–40, 2016. 16, 17, 136

[14] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification : Benchmark and state of the art," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017. 16, 18

[15] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning : A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019. 16, 136

[16] J. Schmidhuber, "Deep learning in neural networks : An overview," *Neural networks*, vol. 61, pp. 85–117, 2014. 16

[17] N. Hordri, A. Samar, S. Yuhaniz, and S. M. Shamsuddin, "A systematic literature review on features of deep learning in big data analytics," *International Journal of Soft Computing and Its Applications(IJASCA)*, vol. 9, no. 1, pp. 32–49, 2017. 16

[18] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning-a new frontier in artificial intelligence research [research frontier]," *IEEE computational intelligence magazine*, vol. 5, no. 4, pp. 13–18, 2010. 16

[19] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural computation*, vol. 22, no. 12, pp. 3207–3220, 2010. 16

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances In Neural Information Processing Systems*, vol. 25, pp. 1–9, 2012. 16, 44, 45

[21] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, "Gpu computing," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 879–899, 2008. 16

[22] S. Shi, Q. Wang, and X. Chu, "Performance modeling and evaluation of distributed deep learning frameworks on gpus," in *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pp. 949–957, IEEE, 2018. 16

[23] L. Wang, K. A. Scott, L. Xu, and D. A. Clausi, "Sea ice concentration estimation during melt from dual-pol sar scenes using deep convolutional neural

networks : A case study," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4524–4533, 2016. 16

[24] D. T. Bui, N.-D. Hoang, F. Martínez-Álvarez, P.-T. T. Ngo, P. V. Hoa, T. D. Pham, P. Samui, and R. Costache, "A novel deep learning neural network approach for predicting flash flood susceptibility : A case study at a high frequency tropical storm area," *Science of The Total Environment*, vol. 701, p. 134413, 2020. 16

[25] Y. Li, S. Ye, and I. Bartoli, "Semisupervised classification of hurricane damage from postevent aerial imagery using deep learning," *Journal of Applied Remote Sensing*, vol. 12, no. 4, p. 045008, 2018. 16

[26] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, 2015. 16

[27] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification : A dimension reduction and deep learning approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4544–4554, 2016. 16

[28] H. Lee and H. Kwon, "Going deeper with contextual cnn for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4843–4855, 2017. 16

[29] G. J. Scott, M. R. England, W. A. Starms, R. A. Marcum, and C. H. Davis, "Training deep convolutional neural networks for land–cover classification of high-resolution imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 4, pp. 549–553, 2017. 16

[30] M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh, and Y. Zhang, "Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery," *Remote Sensing*, vol. 10, no. 7, p. 1119, 2018. 16

[31] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *arXiv preprint arXiv :1508.00092*, 2015. 16

[32] M. Vakalopoulou, K. Karantzalos, N. Komodakis, and N. Paragios, "Building detection in very high resolution multispectral data with deep learning features," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 1873–1876, IEEE, 2015. 16, 18

[33] T. Tian, C. Li, J. Xu, and J. Ma, "Urban area detection in very high resolution remote sensing images using deep convolutional neural networks," *Sensors*, vol. 18, no. 3, p. 904, 2018. 16

[34] D. G. Hadjimitsis, C. Clayton, and V. Hope, "An assessment of the effectiveness of atmospheric correction algorithms through the remote sensing of some reservoirs," *International Journal of Remote Sensing*, vol. 25, no. 18, pp. 3651–3674, 2004. 17

[35] N. Cohen, O. Sharir, and A. Shashua, "On the expressive power of deep learning : A tensor analysis," *Journal of Machine Learning Research*, vol. 49, pp. 698–728, 2016. 17

[36] J. Brownlee, *Better Deep Learning : Train Faster, Reduce Overfitting, and Make Better Predictions*. Machine Learning Mastery, 2018. 17

[37] L. Rice, E. Wong, and Z. Kolter, "Overfitting in adversarially robust deep learning," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 8093–8104, Virtual : PMLR, 2020. 17

[38] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications : A meta-analysis and review," *ISPRS journal of photogrammetry and remote sensing*, vol. 152, pp. 166–177, 2019. 17

[39] R. Pellens and P. Grandcolas, "Conservation and management of the biodiversity in a hotspot characterized by short range endemism and rarity : The challenge of new caledonia," *New Developments in Biodiversity Conservation*, vol. 5, no. 4, pp. 139–151, 2010. 17

[40] J. Chazeau, "Research on new caledonian terrestrial fauna : Achievements and prospects," *Biodiversity Letters*, vol. 1, no. 3/4, pp. 123–129, 1993. 17

[41] M. Pascal, B. R. De Forges, H. Le Guyader, and D. Simberloff, "Mining and other threats to the new caledonia biodiversity hotspot," *Conservation Biology*, vol. 22, no. 2, pp. 498–499, 2008. 17

[42] M. Mangeas, J. André, C. Gomez, M. Despinoy, G. Wattelez, and T. Touraivane, "A spatially explicit integrative model for estimating the risk of wildfire impacts in new-caledonia," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 34, no. 1, pp. 37–52, 2019. 17

[43] A. K. Rolland, A. Crépin, C. Kenner, and P. Afro, "Production de données d'occupation du sol 2010-2014 sur la province sud." Available online : http://oeil.nc/cdrn/index.php/resource/bibliographie/view/27689 (accessed on 2 February 2020), 2017. 17, 68

[44] T. Blaschke, S. Lang, E. Lorup, J. Strobl, and P. Zeil, "Object-oriented image processing in an integrated gis/remote sensing environment and perspectives for environmental applications," *Environmental information for planning, politics and the public*, vol. 2, pp. 555–570, 2000. 18, 110

[45] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 117, pp. 11–28, 2016. 18

[46] M. Napiorkowska, D. Petit, and P. Marti, "Three applications of deep learning algorithms for object detection in satellite imagery," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 4839–4842, IEEE, 2018. 18

[47] Y. Xu, Z. Xie, Y. Feng, and Z. Chen, "Road extraction from high-resolution remote sensing imagery using deep learning," *Remote Sensing*, vol. 10, no. 9, p. 1461, 2018. 18

[48] Q. Shi, X. Liu, and X. Li, "Road detection from remote sensing images by generative adversarial networks," *IEEE Access*, vol. 6, pp. 25486–25494, 2018. 18

[49] X. Lu, Y. Zhong, Z. Zheng, Y. Liu, J. Zhao, A. Ma, and J. Yang, "Multi-scale and multi-task deep learning framework for automatic road extraction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9362–9377, 2019. 18

[50] Y. Liu, Z. Zhang, R. Zhong, D. Chen, Y. Ke, J. Peethambaran, C. Chen, and L. Sun, "Multilevel building detection framework in remote sensing images based on convolutional neural networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 10, pp. 3688–3700, 2018. 18

[51] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair, "Deep learning approach for car detection in uav imagery," *Remote Sensing*, vol. 9, no. 4, p. 312, 2017. 18

[52] W. Li, H. Fu, L. Yu, and A. Cracknell, "Deep learning based oil palm tree detection and counting for high-resolution remote sensing images," *Remote Sensing*, vol. 9, no. 1, p. 22, 2017. 18, 136

[53] B. G. Weinstein, S. Marconi, S. Bohlman, A. Zare, and E. White, "Individual tree-crown detection in rgb imagery using semi-supervised deep learning neural networks," *Remote Sensing*, vol. 11, no. 11, p. 1309, 2019. 18, 136

[54] D. Mitja and I. Miranda, "Weed community dynamics in two pastures grown after clearing brazilian amazonian rainforest," *Weed Research*, vol. 50, no. 2, pp. 163–173, 2010. 18, 134

[55] J. L. A. Campos, T. L. L. da Silva, U. P. Albuquerque, N. Peroni, and E. L. Araujo, "Knowledge, use, and management of the babassu palm (attalea speciosa mart. ex spreng) in the araripe region (northeastern brazil)," *Economic Botany*, vol. 69, no. 3, pp. 240–250, 2015. 18, 135

[56] P. Srestasathiern and P. Rakwatin, "Oil palm tree detection with high resolution multi-spectral satellite imagery," *Remote Sensing*, vol. 6, no. 10, pp. 9749–9774, 2014. 18, 135

[57] A. dos Santos, D. Mitja, E. Delaître, L. Demagistri, I. DE SOUZA MIRANDA, T. Libourel Rouge, and M. Petit, "Estimating babassu palm density using automatic palm tree detection with very high spatial resolution satellite images," *Journal of Environmental Management*, vol. 193, pp. 40–51, 2017. 18, 134, 135

[58] J.-P. Gastellu-Etchegorry, "3d modeling of satellite spectral images, radiation budget and energy budget of urban landscapes," *Meteorology and atmospheric physics*, vol. 102, no. 3-4, p. 187, 2008. 18, 137

[59] J. Cherrier, "Les kaoris de nouvelle-calédonie," *Revue Forestière Française*, 1981. 18

[60] E. Teissier Du Cros, "Forêt en nouvelle-calédonie : menaces sur sa biodiversité.," *Revue Forestière Française*, 2005. 18

[61] G. Rousset, M. Despinoy, K. Schindler, and M. Mangeas, "Assessment of deep learning techniques for land use land cover classification in southern new caledonia," *Remote Sensing*, vol. 13, no. 12, p. 2257, 2021. 20

[62] G. Rousset, "Spot6 satellite imagery, land cover and land use classification of 5 areas in the south province of new caledonia." https://doi.org/10.23708/PHVOJD, VERSION PROVISOIRE, 2021. 20, 66, 73

[63] G. Rousset, F. De Boissieu, C. E. Menkes, J. Lefèvre, R. Frouin, M. Rodier, V. Ridoux, S. Laran, S. Bonnet, and C. Dupouy, "Remote sensing of *Trichodesmium* spp. mats in the western tropical south pacific," *Biogeosciences*, vol. 15, no. 16, pp. 5203–5219, 2018. 20

[64] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016. 23

[65] L. Deng and D. Yu, "Deep learning : Methods and applications," *Found. Trends Signal Process.*, vol. 7, no. 3-4, pp. 197–387, 2014. 23, 65

[66] D. Kriesel, "A brief introduction to neural networks," 2007. 23

[67] S. Haykin, *Neural Networks and Learning Machines*. Pearson : Prentice Hall, 2008. 23

[68] F. Rosenblatt, "The perceptron : a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 656, no. 6, pp. 386–408, 1958. 24

[69] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998. 33, 44, 120

[70] A. F. Agarap, "An architecture combining convolutional neural network (cnn) and support vector machine (svm) for image classification," *arXiv preprint arXiv :1712.03541*, 2017. 38

[71] A. Wang, Y. Wang, and Y. Chen, "Hyperspectral image classification based on convolutional neural network and random forest," *Remote Sensing Letters*, vol. 10, no. 11, pp. 1086–1094, 2019. 38

[72] X. Ren, H. Guo, S. Li, S. Wang, and J. Li, "A novel image classification method with cnn-xgboost model," in *International Workshop on Digital Watermarking*, pp. 378–390, Springer, 2017. 38

[73] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv :1207.0580*, 2012. 39

[74] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout : A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. 40

[75] S. Ioffe and C. Szegedy, "Batch normalization : Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015. 40

[76] S. Santurkar, D. Tsipras, A. Ilyas, and A. Mądry, "How does batch normalization help optimization ?," in *Proceedings of the 32nd international conference on neural information processing systems*, pp. 2488–2498, 2018. 41

[77] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization : an empirical study of their impact to deep learning," *Multimedia Tools and Applications*, pp. 1–39, 2020. 41

[78] X. Li, S. Chen, X. Hu, and J. Yang, "Understanding the disharmony between dropout and batch normalization by variance shift," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 41

[79] G. Chen, P. Chen, Y. Shi, C.-Y. Hsieh, B. Liao, and S. Zhang, "Rethinking the usage of batch normalization and dropout in the training of deep neural networks," *arXiv preprint arXiv :1905.05928*, 2019. 41

[80] M. D. Zeiler, "Adadelta : an adaptive learning rate method," *arXiv preprint arXiv :1212.5701*, 2012. 42

[81] Z. Li and S. Arora, "An exponential learning rate schedule for deep learning," *arXiv preprint arXiv :1910.07454*, 2019. 42

[82] A. Gotmare, N. S. Keskar, C. Xiong, and R. Socher, "A closer look at deep learning heuristics : Learning rate restarts, warmup and distillation," *arXiv preprint arXiv :1810.13243*, 2018. 42

[83] O. Sharma, "Deep challenges associated with deep learning," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 72–75, 2019. 42

[84] J. Pfanzagl, *Parametric statistical theory*. Walter de Gruyter, 2011. 42

[85] R. Moore and J. DeNero, "L1 and l2 regularization for multiclass hinge loss models," in *Symposium on machine learning in speech and language processing*, 2011. 42

[86] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019. 42, 94

[87] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1542–1547, IEEE, 2018. 42

[88] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv :1712.04621*, 2017. 42

[89] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 international interdisciplinary PhD workshop (IIPhDW)*, pp. 117–122, IEEE, 2018. 42

[90] J. Wang, L. Perez, *et al.*, "The effectiveness of data augmentation in image classification using deep learning," *Convolutional Neural Networks Vis. Recognit*, vol. 11, pp. 1–8, 2017. 42

[91] J. Kukačka, V. Golkov, and D. Cremers, "Regularization for deep learning : A taxonomy," *arXiv preprint arXiv :1710.10686*, 2017. 43

[92] R. Moradi, R. Berangi, and B. Minaei, "A survey of regularization strategies for deep models," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 3947–3986, 2020. 43

[93] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 44, 46, 47, 48

[94] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017. 44, 49, 50, 51

[95] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, pp. 630–645, Springer, 2016. 46, 57

[96] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015. 52, 53, 55

[97] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet : A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017. 52, 55, 56

[98] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab : Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018. 52, 57, 58

[99] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, pp. 1520–1528, 2015. 55

[100] O. Ronneberger, P. Fischer, and T. Brox, "U-net : Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015. 55

[101] N. Audebert, B. L. Saux, and S. Lefèvre, "Semantic segmentation of earth observation data using multimodal and multi-scale deep networks," in *Asian conference on computer vision*, pp. 180–196, 2016. 55

[102] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab : Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017. 57

[103] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018. 57

[104] F. Chollet, "Xception : Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017. 57

[105] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017. 58

[106] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv :1409.1556*, 2014. 58

[107] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet : A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv :1606.02147*, 2016. 58

[108] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn : Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015. 58

[109] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014. 58

[110] O. Gargominy, *Biodiversité et conservation dans les collectivités françaises d'outre-mer*, vol. Collection Planète Nature. Comité français pour l'UICN, Paris, France, 2003. 63

[111] GIE OCEANIDE, A. Jérémy, W. Antoine, and OEIL, "L'évolution des paysages en province sud." Available online : https://www.oeil.nc/cdrn/index.php/resource/bibliographie/view/2322 (accessed on 12 March 2020), 2012. 63

[112] A. Veldkamp and E. Lambin, "Predicting land-use change," *Agriculture, Ecosystems & Environment*, vol. 85, no. 1, pp. 1–6, 2001. 63

[113] S. Giertz, B. Junge, and B. Diekkrüger, "Assessing the effects of land use change on soil physical properties and hydrological processes in the sub-humid tropical environment of west africa," *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 30, no. 8-10, pp. 485–496, 2005. 63

[114] S. Hu, G. Zhi-mao, and Y. Jun-ping, "The impacts of urbanization on soil erosion in the loess plateau region," *Journal of Geographical Sciences*, vol. 11, no. 3, pp. 282–290, 2001. 63

[115] L.-Y. Wang, Y. Xiao, E.-M. Rao, L. Jiang, Y. Xiao, and Z.-Y. Ouyang, "An assessment of the impact of urbanization on soil erosion in inner mongolia," *International journal of environmental research and public health*, vol. 15, no. 3, p. 550, 2018. 63

[116] A. Jeong and R. I. Dorn, "Soil erosion from urbanization processes in the sonoran desert, arizona, usa," *Land Degradation & Development*, vol. 30, no. 2, pp. 226–238, 2019. 63

[117] B. J. Hibbs and J. M. Sharp Jr, "Hydrogeological impacts of urbanization," *Environmental & Engineering Geoscience*, vol. 18, no. 1, pp. 3–24, 2012. 63

[118] J. M. Marzluff and M. Restani, "The effects of forest fragmentation," *Forest fragmentation : wildlife and management implications*, p. 155, 1999. 63

[119] J. Masó, I. Serral, C. Domingo-Marimon, and A. Zabala, "Earth observations for sustainable development goals monitoring based on essential variables and driver-pressure-state-impact-response indicators," *International Journal of Digital Earth*, vol. 13, no. 2, pp. 217–235, 2020. 63

[120] M. Mainguet, "La télédétection et le suivi écologique : outils d'une nouvelle géographie," in *20 ans de TIGR, 20 ans de géographie* (M. Bazin, J. Domingo, and A. Marre, eds.), n°79-80, pp. 103–115, Reims : Travaux de l'Institut Géographique de Reims, 1990. 64

[121] R. G. Congalton, "A review of assessing the accuracy of classifications of remotely sensed data," *Remote sensing of environment*, vol. 37, no. 1, pp. 35–46, 1991. 64

[122] P. E. Lehner, L. Adelman, R. J. DiStasio, M. C. Erie, J. S. Mittel, and S. L. Olson, "Confirmation bias in the analysis of remote sensing data," *IEEE Transactions on Systems, Man, and Cybernetics-Part A : Systems and Humans*, vol. 39, no. 1, pp. 218–226, 2008. 64

[123] L. J. Jansen and A. Di Gregorio, "The problems of current land cover classifications : development of a new approach," *Land cover and land use information systems for European Union policy needs*, p. 93, 1998. 64

[124] P. Fisher, A. J. Comber, and R. Wadsworth, *Land use and land cover : contradiction or complement*. Wiley Chichester, 2005. 64

[125] S. Isnard, L. L'huillier, F. Rigault, and T. Jaffré, "How did the ultramafic soils shape the flora of the new caledonian hotspot ?," *Plant and Soil*, vol. 403, no. 1-2, pp. 53–76, 2016. 65

[126] R. A. Mittermeier, T. B. Werner, and A. Lees, "New caledonia – a conservation imperative for an ancient land," *Oryx*, vol. 30, no. 2, p. 104–112, 1996. 65

[127] J. S. Wilson, M. Clay, E. Martin, D. Stuckey, and K. Vedder-Risch, "Evaluating environmental influences of zoning in urban ecosystems with remote sensing," *Remote Sensing of Environment*, vol. 86, no. 3, pp. 303–321, 2003. 65

[128] D. J. Lary, A. H. Alavi, A. H. Gandomi, and A. L. Walker, "Machine learning in geosciences and remote sensing," *Geoscience Frontiers*, vol. 7, no. 1, 2015. 65

[129] P. M. Teillet, "Image correction for radiometric effects in remote sensing," *International Journal of Remote Sensing*, vol. 7, no. 12, pp. 1637–1651, 1986. 65

[130] S. M. Vicente-Serrano, F. Pérez-Cabello, and T. Lasanta, "Assessment of radiometric correction techniques in analyzing vegetation variability and change using time series of landsat images," *Remote Sensing of Environment*, vol. 112, no. 10, pp. 3916–3934, 2008. 65

[131] J. Yang, P. Gong, R. Fu, M. Zhang, J. Chen, S. Liang, B. Xu, J. Shi, and R. Dickinson, "The role of satellite remote sensing in climate change studies," *Nature climate change*, vol. 3, no. 10, pp. 875–883, 2013. 66

[132] K. E. Joyce, S. E. Belliss, S. V. Samsonov, S. J. McNeill, and P. J. Glassey, "A review of the status of satellite remote sensing and image processing techniques for mapping natural hazards and disasters," *Progress in physical geography*, vol. 33, no. 2, pp. 183–207, 2009. 66

[133] M. Enenkel, C. Steiner, T. Mistelbauer, W. Dorigo, W. Wagner, L. See, C. Atzberger, S. Schneider, and E. Rogenhofer, "A combined satellite-derived drought indicator to support humanitarian aid organizations," *Remote Sensing*, vol. 8, no. 4, p. 340, 2016. 66

[134] M. A.-A. Hoque, S. Phinn, C. Roelfsema, and I. Childs, "Tropical cyclone disaster management using remote sensing and spatial analysis : A review," *International journal of disaster risk reduction*, vol. 22, pp. 345–354, 2017. 66

[135] J. Gao, P. Li, Z. Chen, and J. Zhang, "A survey on deep learning for multimodal data fusion," *Neural Computation*, vol. 32, pp. 829–864, 2020. 67, 157

[136] H. Y., S. C., C. G., B. M., L. M., W. B., J.-L. Weber, O. C., C. M-H, and N. Sifakis, *Corine Land Cover Technical Guide*. Office for Official Publications of the European Communities, 1994. 68

[137] S. Georganos, T. Grippa, S. Vanhuysse, M. Lennert, M. Shimoni, and E. Wolff, "Very high resolution object-based land use–land cover urban classification using extreme gradient boosting," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, pp. 607–611, 2018. 77

[138] R. Shwartz-Ziv and A. Armon, "Tabular data : Deep learning is not all you need," *arXiv preprint arXiv :2106.03253*, 2021. 77

[139] T. T. Ngo, C. Collet, and V. Mazet, "Détection simultanée de l'ombre et la végétation sur des images aériennes couleur en haute résolution," *Traitement du signal*, vol. 32, 2014. 78

[140] R. B. Myneni, F. G. Hall, P. J. Sellers, and A. L. Marshak, "The interpretation of spectral vegetation indexes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, pp. 481–486, 1995. 78

[141] J. W. Rouse, R. H. Haas, J. A. Schell, and D. Deering, "Monitoring the vernal advancement and retrogradation (green wave effect) of natural vegetation," *Progress Report RSC 1978-1*, p. 112, 1973. 78

[142] T. Gevers and A. W. M. Smeulders, "Color-based object recognition," *Pattern Recognition*, vol. 32, no. 3, pp. 453–464, 1999. 78, 79

[143] J. Qi, A. Chehbouni, A. R. Huete, Y. H. Kerr, and S. Sorooshian, "A modified soil adjusted vegetation index," *Remote Sensing of Environment*, vol. 48, no. 2, pp. 119–126, 1994. 78

[144] D. M. Woebbecke, G. E. Meyer, K. Von Bargen, and D. A. Mortensen, "Color indices for weed identification under various soil, residue, and lighting conditions," *Transactions of the ASAE*, vol. 38, pp. 259–269, 1994. 78

[145] H. Xu, "Modification of normalised difference water index (ndwi) to enhance open water features in remotely sensed imagery," *International Journal of Remote Sensing*, vol. 27, pp. 3025–3033, 2006. 78

[146] G. E. Meyer and J. C. Neto, "Verification of color vegetation indices for automated crop imaging applications," *Computers and Electronics in Agriculture*, vol. 63, no. 2, pp. 282–293, 2008. 79

[147] R. Kerkache, J.-P. Dedieu, F. Bornicchia, and H. Pella, "Apport des informations de texture en télédétection pour l'étude de l'occupation des sols / the contribution to land-use studies of textural analysis using remote sensing data," in *Revue de géographie alpine, tome 85, n°3.*, 1997. 79

[148] J. Zhou, R. Y. Guo, M. Sun, T. T. Di, S. Wang, J. Zhai, and Z. Zhao, "The effects of glcm parameters on lai estimation using texture values from quickbird satellite imagery," *Scientific Reports*, vol. 7, no. 1, pp. 1–12, 2017. 79

[149] C. Song and M. Dickinson, "Extracting forest canopy structure from spatial information of high resolution optical imagery : tree crown size versus leaf area index," *International Journal of Remote Sensing*, vol. 29, no. 19, pp. 5605–5622, 2008. 79

[150] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973. 79

[151] P. Maillard, "Comparing texture analysis methods through classification," *Photogrammetric Engineering and Remote Sensing*, vol. 69, no. 4, pp. 357–367, 2003. 79

[152] G. Chust, D. Ducrot, and J. L. Pretus, "Land cover mapping with patch-derived landscape indices," *Landscape and Urban Planning*, vol. 69, no. 4, pp. 437–449, 2004. 80

[153] M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, and G. Giacinto, "Novel feature extraction, selection and fusion for effective malware family classification," in *Proceedings of the sixth ACM conference on data and application security and privacy*, pp. 183–194, 2016. 80

[154] T. Chen and C. Guestrin, "Xgboost : A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, New York, NY, United States : Association for Computing Machinery, 2016. 80, 81

[155] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, pp. 1–54, 2019. 82

[156] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 2650–2658, 2015. 83

[157] S. Paisitkriangkrai, J. Sherrah, P. Janney, V.-D. Hengel, *et al.*, "Effective semantic pixel labelling with convolutional networks and conditional random fields," pp. 36–43, 2015. 100

[158] C. Gonzalo-Martín, A. GarciaPedrero, M. Lillo-Saavedra, and E. Menasalvas, "Deep learning for superpixel-based classification of remote sensing images," 2016. 100

[159] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 1–23, 2017. 104

[160] T.-C. Su, "A filter-based post-processing technique for improving homogeneity of pixel-wise classification data," *European Journal of Remote Sensing*, vol. 49, pp. 531–552, 2016. 105

[161] N. Myers, R. A. Mittermeier, C. G. Mittermeier, G. A. Da Fonseca, and J. Kent, "Biodiversity hotspots for conservation priorities," *Nature*, vol. 403, no. 6772, p. 853, 2000. 109

[162] Y. Pillon, J. Munzinger, H. Amir, and M. Lebrun, "Ultramafic soils and species sorting in the flora of new caledonia," *Journal of Ecology*, vol. 98, no. 5, pp. 1108–1116, 2010. 109

[163] A. Lillie and R. Brothers, "The geology of new caledonia," *New Zealand Journal of Geology and Geophysics*, vol. 13, no. 1, pp. 145–183, 1970. 109

[164] D. G. Vincent, "The south pacific convergence zone (spcz) : A review," *Monthly Weather Review*, vol. 122, no. 9, pp. 1949–1970, 1994. 109

[165] J. L. Awange, *Environmental monitoring using GNSS : Global navigation satellite systems*. Springer Science & Business Media, 2012. 109

[166] L. G. Hafemann, L. S. Oliveira, and P. Cavalin, "Forest species recognition using deep convolutional neural networks," in *2014 22nd International Conference on Pattern Recognition*, pp. 1103–1107, IEEE, 2014. 109

[167] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, 2017. 109

[168] C. He, Z. Liu, S. Gou, Q. Zhang, J. Zhang, and L. Xu, "Detecting global urban expansion over the last three decades using a fully convolutional network," *Environmental Research Letters*, vol. 14, no. 3, 2019. 109

[169] J. Deng, Y. Huang, B. Chen, C. Tong, P. Liu, H. Wang, and Y. Hong, "A methodology to monitor urban expansion and green space change using a time series of multi-sensor spot and sentinel-2a images," *Remote Sensing*, vol. 11, no. 10, p. 1230, 2019. 109

[170] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3325–3337, 2014. 109

[171] P. Blanc, *Développement de méthodes pour la détection de changement*. PhD thesis, École Nationale Supérieure des Mines de Paris, 1999. 110

[172] A. Fournier, *Détection et classification de changements sur des scènes urbaines en télédétection*. PhD thesis, Institut Supérieur de l'Aéronautique et de l'Espace, 2008. 110

[173] H. Yates and W. Bandeen, "Meteorological applications of remote sensing from satellites," *Proceedings of the IEEE*, vol. 63, no. 1, pp. 148–163, 1975. 110

[174] C. Van Westen, "Remote sensing for natural disaster management," *International archives of photogrammetry and remote sensing*, vol. 33, no. B7/4 ; PART 7, pp. 1609–1617, 2000. 110

[175] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015. 110

[176] M. Gong, T. Zhan, P. Zhang, and Q. Miao, "Superpixel-based difference representation learning for change detection in multispectral remote sensing images," *IEEE Transactions on Geoscience and Remote sensing*, vol. 55, no. 5, pp. 2658–2673, 2017. 110

[177] A. M. El Amin, Q. Liu, and Y. Wang, "Zoom out cnns features for optical remote sensing change detection," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, pp. 812–817, IEEE, 2017. 110

[178] J. Hemming, *Change in the Amazon Basin : The frontier after a decade of colonisation*, vol. 2. Manchester University Press, 1985. 133

[179] D. Mitja, E. Delaître, A. dos Santos, I. DE SOUZA MIRANDA, R. F. Rodriguez Coelho, D. Jorge Macedo, L. Demagistri, and M. Petit, "Satellite images combined with field data reveal negative changes in the distribution of babassu palms after clearing off amazonian forests," *Environmental Management*, vol. 61, no. 2, pp. 321–336, 2018. 134

[180] R. Coelho, I. Miranda, D. Mitja, *et al.*, "Description of successional process at benfica settlement project, southeastern of state of pará, eastern amazon," *Boletim do Museu Paraense Emílio Goeldi, Ciências Naturais*, vol. 7, no. 3, pp. 251–282, 2012. 134

[181] D. Mitja and I. D. K. FERRAZ, "Establishment of babassu in pasture in para, brazil," *Palms*, vol. 45, no. 3, pp. 138–147, 2001. 134

[182] S. Barot, D. Mitja, I. Miranda, G. Meija, and M. Grimaldi, "Reproductive plasticity in an amazonian palm," *Evolutionary Ecology Research*, vol. 7, no. 7, pp. 1051–1065, 2005. 134

[183] A. Hercos, M. Sobansky, H. L. d. Queiroz, and A. E. Magurran, "Local and regional rarity in a diverse tropical fish assemblage," *Proceedings of the Royal Society B : Biological Sciences*, vol. 280, no. 1751, p. 20122076, 2013. 135

[184] G. P. J. M. J. Caley and P. L. Munday, "Rarity in coral reef fish communities," *Coral reef fishes : dynamics and diversity in a complex ecosystem*, p. 81, 2002. 135

[185] J. Zhou, C. Proisy, X. Descombes, G. Le Maire, Y. Nouvellon, J.-L. Stape, G. Viennois, J. Zerubia, and P. Couteron, "Mapping local density of young eucalyptus plantations by individual tree detection in high spatial resolution satellite images," *Forest Ecology and Management*, vol. 301, pp. 129–141, 2013. 135

[186] R. Engler, L. T. Waser, N. E. Zimmermann, M. Schaub, S. Berdos, C. Ginzler, and A. Psomas, "Combining ensemble modeling and remote sensing for mapping individual tree species at high spatial resolution," *Forest Ecology and Management*, vol. 310, pp. 64–73, 2013. 135

[187] S. Malek, Y. Bazi, N. Alajlan, H. AlHichri, and F. Melgani, "Efficient framework for palm tree detection in uav images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 12, pp. 4692–4703, 2014. 135

[188] G. Bertrand, *Characterization of adaptive responses to drought stress in Southeastern Amazon of three forage species grown in mixed crops.* PhD thesis, Université Paris-Est, 2009. 135

[189] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing : A comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017. 136

[190] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving : Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE international conference on computer vision*, pp. 2722–2730, 2015. 136

[191] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012. 136

[192] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1345–1359, 2010. 136

[193] S. Villon, C. Iovan, M. Mangeas, T. Claverie, D. Mouillot, S. Villéger, and L. Vigliola, "Automatic underwater fish species classification with limited data using few-shot learning," *Ecological Informatics*, vol. 63, p. 101320, 2021. 136

[194] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International conference on machine learning*, pp. 2152–2161, PMLR, 2015. 136

[195] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, pp. 3630–3638, 2016. 136

[196] J.-P. Gastellu-Etchegorry, T. Yin, N. Lauret, T. Cajgfinger, T. Gregoire, E. Grau, J.-B. Feret, M. Lopes, J. Guilleux, G. Dedieu, *et al.*, "Discrete aniso-

tropic radiative transfer (dart 5) for modeling airborne and satellite spectroradiometer and lidar acquisitions of natural and urban landscapes," *Remote Sensing*, vol. 7, no. 2, pp. 1667–1701, 2015. 137

[197] J. Cho, K. Lee, E. Shin, G. Choy, and S. Do, "How much data is needed to train a medical image deep learning system to achieve necessary high accuracy ?," *arXiv preprint arXiv :1511.06348*, 2015. 142

[198] Q. Zhang, Q. Yuan, J. Li, Z. Li, H. Shen, and L. Zhang, "Thick cloud and cloud shadow removal in multitemporal imagery using progressively spatio-temporal patch group deep learning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 162, pp. 148–160, 2020. 157

[199] D. Ienco, R. Interdonato, R. Gaetano, and D. Ho Tong Minh, "Combining sentinel-1 and sentinel-2 satellite image time series for land cover mapping via a multi-source deep learning architecture," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 158, pp. 11–22, 2019. 157