

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Ecole Nationale Polytechnique  
Agence Spatiale ALgérienne



المدرسة الوطنية متعددة التقنيات  
Ecole Nationale Polytechnique



Mémoire de Projet de Fin d'Etude pour l'obtention du diplôme d'Ingénieur  
d'Etat en Electronique

---

# Détection de Changement par Images d'Observation de la Terre et Techniques Avancées d'Apprentissage Profond

---

Réalisé par :

**OUADAH Cylia**  
**HAMADACHE Rachika ElHassna**

Encadreur et co-encadreur :

**Dr Meziane IFTENE**      **Dr Nesrine BOUADJENEK**

Présenté le 21/09/2020

## Composition du Jury :

Président	Cherif LARBES	Professeur à l'ENP
Examinatrice	Nour El Houda BENALIA	Maître de Conférence (B) à l'ENP
Encadreur	Meziane IFTENE	Maître de Recherche à l'ASAL
Co-Encadreur	Nesrine BOUADJENEK	Maître de Conférence (B) à l'ENP

ENP 2020



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Ecole Nationale Polytechnique  
Agence Spatiale ALgérienne



المدرسة الوطنية متعددة التقنيات  
Ecole Nationale Polytechnique



Mémoire de Projet de Fin d'Etude pour l'obtention du diplôme d'Ingénieur  
d'Etat en Electronique

---

# Détection de Changement par Images d'Observation de la Terre et Techniques Avancées d'Apprentissage Profond

---

Réalisé par :

**OUADAH Cylia**  
**HAMADACHE Rachika ElHassna**

Encadreur et co-encadreur :

**Dr Meziane IFTENE**      **Dr Nesrine BOUADJENEK**

Présenté le 21/09/2020

## Composition du Jury :

Président	Cherif LARBES	Professeur à l'ENP
Examinatrice	Nour El Houda BENALIA	Maître de Conférence (B) à l'ENP
Encadreur	Meziane IFTENE	Maître de Recherche à l'ASAL
Co-Encadreur	Nesrine BOUADJENEK	Maître de Conférence (B) à l'ENP

ENP 2020

## ملخص

أحدث التقدم التكنولوجي في مجال الرؤية الحاسوبية مع تطوير أجيال جديدة من الأقمار الصناعية ثورة في علم مراقبة الأرض، خاصة في مجال كشف التغيير. لهذا الغرض، يسלט عملنا الضوء على التكنولوجيات الحديثة باقتراح نظامين للكشف عن التغيير يعتمدان على الشبكات العصبية الترشيحية كليا. المقاربة الأولى « FCN-VGG-16 adapté comme extracteur de caractéristiques » تقترح طريقة غير خاضعة للإشراف مستغلة استخراج الميزات من طبقات مختلفة لشبكة (VGG-16) مكيفة دون طبقات تجميع. تسלט المقاربة الثانية « FCN-Siamois ED&DI sans maxpooling » الضوء على أداء شبكة سيامية ترشيحية كليا ومعدلة، مدربة تدريب شامل و مستخدمة تقنية « long skip-connections » لاستغلال مفاهيم المسافة الإقليدية وفرق الصورة بهدف التصنيف على مستوى البيكسل. يعتمد تدريب هذه الشبكة على مجموعة بيانات أنشأناها من صور الأقمار الصناعية المتوفرة مع خرائط التغيير المرجعية لها. تم تقييم النظامين على ثلاث مجموعات بيانات لأنواع مختلفة من صور الأقمار الصناعية. وقد أعطت التجارب المختلفة التي أجريت على هذه المجموعة من الصور نتائج كمية ومرئية جد واعدة، وأظهرت كفاءة النماذج المقترحة في كشف التغييرات في البيئات الحضرية والزراعية.

**الكلمات المفتاحية :** صور الأقمار الصناعية، الكشف عن التغييرات، التعلم العميق، CNN، نقل المعارف.

## Abstract

The technological progress in the field of computer vision along with the new generations of satellites has revolutionized the science of earth observation, especially in change detection. For this purpose, our work highlights these new technologies and proposes two fully convolutional neural network-based change detection systems. The first approach « Adapted FCN-VGG-16 Network as feature extractor » presents an unsupervised method exploiting features extraction from different layers of a VGG-16 network adapted without pooling layers. The second approach « FCN-Siamoise ED&DI without maxpooling » highlights the performance of a modified Siamese fully convolutional network, using end-to-end training and employing « long skip-connections » to exploit the results of the Euclidean distance and the image difference for the pixel classification. The training of this method refers to a database created by ourselves from the available bitemporal satellite images and their respective ground truths.

The system is evaluated on three datasets of different types of satellite images. The different experiments carried out on this set of images have given very promising quantitative and visual results, and show the efficiency of the proposed models for change detection in urban and agricultural environments.

**Keywords :** Satellite Imagery, Change Detection, Deep Learning, Convolutional Neural Networks, Transfer Learning.

## Résumé

Le progrès technologique qu'a connu le domaine de la vision par ordinateur et les nouvelles générations de satellites, a fait révolutionner la science de l'observation de la terre, en particulier la problématique de la détection de changement.

A cet effet, notre travail met en avant ces technologies modernes et propose deux systèmes de détection de changement basés sur les réseaux de neurones entièrement convolutifs. La première approche « FCN-VGG-16 adapté comme extracteur de caractéristiques » présente une méthode non-supervisée exploitant l'extraction de caractéristiques de différentes couches d'un réseau VGG-16 adapté sans couches de Pooling. La seconde approche « FCN-Siamois ED&DI sans maxpooling » met en avant les performances d'un réseau siamois modifié entièrement convolutif, entraîné de façon end-to-end et utilisant les « long skip-connections » afin d'exploiter la distance euclidienne et la différence d'image pour la classification des pixels. L'entraînement de cette deuxième approche supervisée se réfère à une base de données créée par nous-mêmes, à partir des images satellites bitemporelles disponibles avec leurs cartes de changements respectives.

L'évaluation des deux systèmes est faite sur trois datasets de différents types. Les différentes expériences menées sur cet ensemble d'images ont donné des résultats quantitatifs et visuels très prometteurs, et montrent l'efficacité des modèles proposés pour la détection de changement en milieu urbain et en milieu agricole.

**Mots-clés :** Imagerie satellitaire, Détection de changement, apprentissage profond, réseaux de neurones convolutifs, apprentissage par transfert.

# Remerciements

Au terme de ce travail, nous adressons nos sincères remerciements à notre encadreur Dr Meziane IFTENE, Maître de recherche à la cellule de veille technologique au niveau de l'Agence Spatiale Algérienne pour son temps, son accueil et la confiance qu'il nous a accordée en nous proposant le thème de notre projet.

Nous exprimons aussi notre sincère reconnaissance envers notre co-encadreur, Dr Nesrine BOUADJENEK, Maître de conférences (B) à l'Ecole Nationale Polytechnique pour son soutien, sa patience et sa disponibilité tout au long de la période de notre travail.

Nous tenons à témoigner notre profonde gratitude envers tous les membres du jury, Mr Cherif LARBES, Professeur à l'Ecole Nationale Polytechnique et Mme Nour El Houda BENALIA, Maître de conférences (B) à l'Ecole Nationale Polytechnique pour l'honneur qu'ils nous ont accordé en acceptant de lire et d'examiner notre travail.

Notre immense gratitude se destine également à la famille polytechnicienne, tout particulièrement, aux enseignants du département d'Électronique auxquels nous devons notre formation d'ingénieurs.

# Dédicaces

C'est avec grande gratitude que je dédie ce travail,

À mes chers parents qui, grâce à leurs sacrifices, leur soutien et leurs encouragements tout au long de mes années d'études, j'ai pu réussir à surpasser chaque obstacle. À ma mère, tout particulièrement, qui a toujours été présente pour moi, et qui a été compréhensive durant toutes les épreuves difficiles que j'ai pu endurées.

À ma sœur, pour m'avoir encouragée et conseillée, pour tous les moments et les discussions partagés ensemble, qui ont rendu les périodes difficiles plus paisibles.

À ma grand-mère, ma cousine Meriem et ma tante Karima, pour leur soutien inconditionnel et leurs encouragements.

À l'ensemble de mes amis, Meriem en particulier, pour m'avoir aidée dans les moments d'incertitude et pour m'avoir soutenue dans tout ce que j'ai pu entreprendre.

À mes camarades du département d'Electronique de l'ENP, pour avoir rendu mon expérience universitaire plus joyeuse.

À Hasna, ma partenaire de travail, qui a été très productive, patiente et compréhensive. Nous avons passé beaucoup de périodes de stress et d'incertitudes ensemble, pour enfin réussir à finir convenablement notre projet.

**Cyria**

# Dédicaces

Je dédie ce modeste travail,

A mes grands-mères; pour vos prières, votre bienveillance et précieux conseils. A mes grands-pères qui sont à l'au-delà, que Dieu, Miséricordieux, vous accueille en son vaste paradis.

A mes chers parents et mes chères soeurs; pour votre soutien, votre sagesse et vos encouragements qui m'ont accompagné tout au long de ma vie et qui m'ont permis de toujours persévérer et donner le meilleur de moi-même. Merci encore pour votre patience et de m'avoir *supporté* durant cette longue et dure période.. En particulier ma mère, pour avoir toujours veillé sur moi.

A ma chère amie Maissa, avec qui on a surmonté en binôme les innombrables défis et périlleuses aventures de la prépa. Merci pour ton soutien, ta bonne humeur et ta positive (*green ♣*) attitude..

A mes valeureuses amies Hana, Asma et Bouchra, pour leur écoute, leurs encouragements et les moments partagés au cours de ces années.

Aux merveilleuses personnes que j'ai rencontré à l'ENP et à travers le CAP : Sara, Ahlem, Camélia, Yasmine, Imene, Nassila, Lotfi et Ghani ainsi qu'à toute la team WS5.

A mes camarades d'Électronique, pour l'entraide et la complicité qui nous unissent.

A ma binôme Cylia, pour l'incroyable travail que tu as accompli, pour tes efforts et ton perfectionnisme, pour le stress partagé et ta patience durant ces mois.

*~ Une pensée amère à tous les détenus d'opinion.. Liberté à tous ces prisonniers de l'injustice!*

***Hasna***

# Table des matières

Liste des figures

Liste des tableaux

Liste des abréviations

<b>Introduction générale</b>	<b>12</b>
<b>1 Détection de Changement en Imagerie Satellite</b>	<b>14</b>
1.1 Introduction . . . . .	15
1.2 Imagerie satellite . . . . .	15
1.2.1 Types de résolutions en télédétection . . . . .	15
1.3 Détection de changement . . . . .	16
1.3.1 Approches de la détection de changements . . . . .	18
1.3.2 Modes de comparaison . . . . .	19
1.4 Revue de la littérature . . . . .	19
1.4.1 Méthodes basées sur le Machine Learning . . . . .	20
1.4.2 Méthodes basées sur les réseaux de neurones profonds . . . . .	20
1.4.3 Applications des réseaux de neurones convolutifs « CNN » à la détection de changement en Imagerie satellitaire . . . . .	21
1.5 Challenges et problématiques liés à la détection de changement en télédé- tection . . . . .	24
1.6 Conclusion . . . . .	24
<b>2 Réseaux de Neurones Convolutifs</b>	<b>26</b>
2.1 Introduction . . . . .	27
2.2 Réseaux de Neurones Artificiels . . . . .	27
2.2.1 Le perceptron . . . . .	28
2.2.2 Perceptron multicouches MLP . . . . .	28
2.2.3 Apprentissage d'un MLP . . . . .	29
2.3 Réseaux de neurones convolutifs . . . . .	32
2.3.1 Couches d'un CNN . . . . .	33
2.3.2 Sous-apprentissage et Sur-apprentissage . . . . .	36
2.3.3 Optimisation de l'apprentissage par des algorithmes d'optimisation avancés . . . . .	38
2.3.4 Apprentissage par Transfert . . . . .	39
2.3.5 Architectures des CNNs les plus utilisées . . . . .	41
2.4 Conclusion . . . . .	43

<b>3 Réseaux de Neurones Convolutifs Adaptés pour la Détection de Changement</b>	<b>44</b>
3.1 Introduction . . . . .	45
3.2 Les réseaux de neurones entièrement convolutifs . . . . .	45
3.3 L'extraction de caractéristiques pour la détection de changement . . . . .	46
3.3.1 La compréhension des caractéristiques des CNNs . . . . .	46
3.3.2 Architecture basée sur l'extraction de features . . . . .	47
3.4 Réseaux siamois entièrement convolutifs pour la détection de changement .	49
3.4.1 Les réseaux siamois . . . . .	50
3.4.2 Architecture du modèle adopté . . . . .	50
3.5 Conclusion . . . . .	53
<b>4 Implémentation des Méthodes et Résultats</b>	<b>54</b>
4.1 Introduction . . . . .	55
4.2 Base de données . . . . .	55
4.3 Bibliothèques et Outils . . . . .	57
4.3.1 Python . . . . .	57
4.3.2 TensorFlow . . . . .	57
4.3.3 Keras . . . . .	58
4.3.4 OpenCV . . . . .	58
4.3.5 Google Colaboratory . . . . .	58
4.4 Critères d'évaluation . . . . .	59
4.4.1 Classification Accuracy . . . . .	59
4.4.2 Coefficient Kappa . . . . .	59
4.4.3 Précision . . . . .	59
4.5 Expériences et Résultats . . . . .	60
4.5.1 Implémentation et résultats de la méthode <i>"FCN-VGG-16 adapté comme extracteur de caractéristiques"</i> . . . . .	60
4.5.2 Implémentation et résultats de la méthode <i>FCN-siamois ED&amp;DI sans maxpooling</i> . . . . .	63
4.5.3 Interprétation des résultats . . . . .	66
4.5.4 Bilan et Comparaison . . . . .	67
4.6 Conclusion . . . . .	69
<b>Conclusion Générale</b>	<b>70</b>
<b>Références Bibliographiques</b>	<b>76</b>

# Table des figures

1.1	Exemple d'une image numérique et sa représentation en matrice de pixels .	16
1.2	Exemple de DC simple . . . . .	17
1.3	Exemple de DC sémantique . . . . .	17
1.4	Performance des algorithmes ML et méthodes DL pour la tâche de la DC en télédétection . . . . .	23
2.1	Modèle d'un neurone formel (Razafimandimby, Loscri et al. 2016) . . . . .	28
2.2	MLP à une couche intermédiaire . . . . .	29
2.3	Fonction sigmoïde . . . . .	31
2.4	Fonction $\tanh(x)$ . . . . .	31
2.5	Fonction ReLU . . . . .	31
2.6	Fonction Leaky ReLU . . . . .	32
2.7	Exemple d'une convolution avec un filtre 3 x 3 . . . . .	33
2.8	Exemple d'une opération de Max-Pooling 2 X 2 . . . . .	35
2.9	Exemple d'une opération de Flattening . . . . .	35
2.10	Couche complètement connectée de 4 classes . . . . .	36
2.11	Sous-apprentissage, sur-apprentissage et généralisation. (Sharma, Nori et al 2014) . . . . .	36
2.12	Compromis biais-variance . . . . .	37
2.13	Illustration du fonctionnement du Dropout (Srivastava, Hinton et al. 2014)	37
2.14	Minimum local, maximum local, point selle (Photo par Rong Ge) . . . . .	38
2.15	Architecture AlexNet . . . . .	41
2.16	Architecture du réseau U-net . . . . .	42
2.17	Architecture du VGG-16 . . . . .	43
3.1	Vizualisation des caractéristiques dans un réseau AlexNet . . . . .	47
3.2	Représentation des hypercolonnes . . . . .	48
3.3	Workflow de la première méthode proposée : ' <i>FCN-VGG-16 adapté comme extracteur de caractéristiques</i> ' . . . . .	49
3.4	Architecture classique d'un réseau siamois . . . . .	50
3.5	Schéma d'un encodeur-décodeur en architecture U . . . . .	51
3.6	Workflow de la deuxième méthode proposée : ' <i>FCN-siamois ED&amp;DI sans maxpooling</i> ' . . . . .	52
4.1	Illustration des étapes suivies pour la création de la base de données . . . . .	55
4.2	Représentation des paires d'images de ' <i>Dataset1</i> ', ' <i>Dataset2</i> ' et ' <i>Dataset3</i> ' respectivement, aux temps T1 et T2 dans les deux premières colonnes, ainsi que leur GT correspondant dans la 3ème colonne. . . . .	56

4.3	Résultats des tests sur <i>Dataset1</i> . Les figures (a), (b) et (c) représentent dans l'ordre l'image 1, l'image 2 et le GT correspondant. Les figures (d), (e) et (f) sont respectivement les résultats des méthodes ' <i>Extracteur de features par VGG-16 classique</i> ', ' <i>VGG-16 adapté avec maxpooling</i> ' et la méthode proposée. . . . .	60
4.4	Résultats des tests sur <i>Dataset2</i> . Les figures (a), (b) et (c) représentent dans l'ordre l'image 1, l'image 2 et le GT correspondant. Les figures (d), (e) et (f) sont respectivement les résultats des méthodes ' <i>Extracteur de features par VGG-16 classique</i> ', ' <i>VGG-16 adapté avec maxpooling</i> ' et la méthode proposée. . . . .	61
4.5	Résultats des tests sur <i>Dataset3</i> . Les figures (a), (b) et (c) représentent dans l'ordre l'image 1, l'image 2 et le GT correspondant. Les figures (d), (e) et (f) sont respectivement les résultats des méthodes ' <i>Extracteur de features par VGG-16 classique</i> ', ' <i>VGG-16 adapté avec maxpooling</i> ' et la méthode proposée. . . . .	62
4.6	Graphes représentant l'évolution de 'loss' et 'précision' respectivement lors de l'entraînement du réseau. . . . .	63
4.7	Résultats des tests sur ' <i>Dataset1</i> '. Les figure (a), (b) et (c) représentent dans l'ordre l'image 1, l'image 2 et le GT correspondant. Les figures (d), (e) (f) et (g) sont respectivement les résultats des méthodes FCN-Siamois ED et FCN-Siamois ED sans maxpooling, FCN-Siamois ED&DI et la méthode proposée. . . . .	64
4.8	Résultats des tests sur ' <i>Dataset2</i> '. Les figure (a), (b) et (c) représentent dans l'ordre l'image 1, l'image 2 et le GT correspondant. Les figures (d), (e) (f) et (g) sont respectivement les résultats des méthodes FCN-Siamois ED et FCN-Siamois ED sans maxpooling, FCN-Siamois ED&DI et la méthode proposée. . . . .	65
4.9	Résultats des tests sur ' <i>Dataset3</i> '. Les figure (a), (b) et (c) représentent dans l'ordre l'image 1, l'image 2 et le GT correspondant. Les figures (d), (e) (f) et (g) sont respectivement les résultats des méthodes FCN-Siamois ED et FCN-Siamois ED sans maxpooling, FCN-Siamois ED&DI et la méthode proposée. . . . .	66
4.10	Illustration des valeurs moyennes des critères d'évaluations pour les différentes variantes testées . . . . .	68

# Liste des tableaux

4.1	Comparaison quantitative des approches sur Dataset1 . . . . .	61
4.2	Comparaison quantitative des approches sur Dataset2 . . . . .	61
4.3	Comparaison quantitative des approches sur Dataset3 . . . . .	62
4.4	L'effet du choix des couches convolutives avec maxpooling à concaténer dans les hypercolonnes pour ' <i>Dataset2</i> ' . . . . .	62
4.5	L'effet du choix des couches convolutives sans maxpooling à concaténer dans les hypercolonnes pour ' <i>Dataset2</i> ' . . . . .	62
4.6	Résultats de la 2ème approche sur ' <i>Dataset1</i> ' . . . . .	63
4.7	Résultats de la 2ème approche sur ' <i>Dataset2</i> ' . . . . .	64
4.8	Résultats de la 2ème approche sur ' <i>Dataset3</i> ' . . . . .	65

# Liste des abréviations

ADAM	Adaptative Moment Estimation
CA	Classification Accuracy
CNN	Convolutional Neural Network ou Réseau de Neurones Convolutif en français.
DC	Détection de Changement.
DCVA	Deep Change Vector Analysis
DI	Difference Image.
DL	Deep Learning ou Apprentissage Profond en français.
ED	Euclidean Distance ou Distance Euclidienne en français.
FC	Fully Connected ou Complètement Connectés en français.
FCN	Fully Convolutional Network ou Réseau Entièrement Convolutif en français
GPU	Graphical Processing Unit ou Processeur Graphique en français.
GT	Ground Truth ou Référence de Changement en français.
IA	Intelligence Artificielle
ILSVRC	Imagenet Large Scale Visual Recognition Challenge.
Kappa	Coefficient Kappa de Cohen
ML	Machine Learning ou Apprentissage Machine en français.
MLP	Multi Layer Perceptron ou Perceptron Multicouche en français.
OSCD	Onera Satellite Change Detection
PCA	Principal Component Analysis ou Analyse en Composantes Principales
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RGB	Red Green Blue
SVM	Support Vector Machine ou Machine à Vecteur de Support en français.
THR	Très Haute Résolution.

# Introduction Générale

L'avènement des satellites de nouvelles générations a révolutionné la science de l'observation de la terre. La possibilité de stocker des données satellites continuellement pour un usage futur, a permis un suivi géographique et temporel précis de l'évolution de la surface de la Terre.

Dans notre travail, nous abordons le domaine de l'exploitation des images satellitaires pour la détection de changement. La DC en télédétection consiste à identifier les changements de la couverture du sol, sur une même zone géographique et entre deux périodes données.

La recherche dans le domaine de la DC en télédétection est exploitée pour la conception de systèmes de haute qualité, destinés à l'analyse et l'aide à la décision et ce, pour plusieurs applications.

On note :

- L'analyse de changements post-catastrophes.
- Le contrôle d'expansion des zones urbaines.
- La surveillance environnementale.
- L'aide à la décision pour le développement durable.
- La quantification des risques naturels.
- Le suivi et la surveillance des cultures.

Les points cités ci-dessus, mettent en évidence la diversité des applications de la tâche de la DC, ce qui démontre la complexité de l'adaptation de ces systèmes. Les systèmes conçus pour une application donnée ne sont pas automatiquement exploitables pour les autres applications. Le temps disponible pour l'analyse présente également une contrainte dans une application, telle que l'analyse des changements post-catastrophes, ce qui amène la recherche dans ce domaine à développer des systèmes de haute précision.

Pour l'obtention de résultats surpassants ceux des méthodes traditionnelles en précision, et manuelles en temps d'analyse, la recherche dans le domaine de la DC s'est orientée récemment vers les techniques d'analyse d'images modernes, notamment l'apprentissage automatique profond. Nous exploitons dans notre travail les réseaux de neurones convolutifs, spécialement conçus pour la classification d'images afin de réaliser deux modèles de DC (supervisé et non supervisé). Le but étant de donner des résultats assez précis, en utilisant peu de données par contrainte de disponibilité de larges bases de données bitemporelles.

Ce document s'articule en quatre chapitres structurés comme suit :

**Chapitre 01** : Dans ce premier chapitre, nous allons présenter le domaine de la DC dans l'imagerie satellitaire, son évolution et les principales approches utilisées

pour la résolution de la problématique de la DC. Nous allons par la suite, donner un aperçu sur la littérature dont les méthodes sont basées sur l'apprentissage machine et l'apprentissage automatique profond.

**Chapitre 02** : Ce chapitre, présente et décrit les CNNs ainsi que la notion d'apprentissage automatique et d'apprentissage par transfert. Enfin, il expose les architectures des CNNs les plus connues dans le cadre de la vision par ordinateur.

**Chapitre 03** : Ce chapitre, consiste à décrire les modèles de DC proposés. La première approche présente une méthode non supervisée, basée sur l'extraction de caractéristiques et exploitant les poids d'un VGG-16 pré-entraîné sur la base de données ImageNet<sup>1</sup> modifié en un FCN sans couches de maxpooling. La deuxième approche est une approche supervisée "end-to-end" mettant en avant un réseau siamois FCN modifié entraîné sur une base de données créée à cette fin.

**Chapitre 04** : Ce dernier chapitre, présente et discute les résultats obtenus pour les deux méthodes proposées. Par la suite, il est question de comparer les résultats des tests, afin de faire valoir l'apport des modèles proposés, et de présenter les avantages et les inconvénients de chacun.

Enfin, Nous finalisons ce document, avec une conclusion qui énonce les challenges rencontrés lors du développement de nos approches, tout en apportant des éventuelles solutions pour l'amélioration des résultats obtenus , ainsi que de nouvelles perspectives à exploiter dans de futurs travaux traitants de la problématique de la DC en Imagerie satellite.

---

1. <http://www.image-net.org/>

# Chapitre 1

## Détection de Changement en Imagerie Satellite

## 1.1 Introduction

La détection de changement est l'une des principales problématiques dans l'analyse d'images d'observation de la terre. Les premiers systèmes de DC ont débuté avec les premières acquisitions d'images satellitaires, et ont évolué en parallèle avec le domaine de la vision par ordinateur. [1] et [2] donnent un aperçu sur l'enchaînement de cette évolution.

Ces dernières années, le progrès des technologies a considérablement impacté le domaine de la télédétection, et a permis à ses applications de dépasser les frontières du militaire et de la défense. Le domaine de la DC en télédétection, a particulièrement évolué grâce à la disponibilité croissante des données satellites d'une part, et l'avancement des techniques d'apprentissage profond d'une autre part.

Dans ce chapitre, nous allons, en premier lieu, présenter l'imagerie satellite et la tâche de la DC, ainsi que ses principales types et approches. Ensuite, une revue de la littérature sera exposée, afin d'avoir un aperçu général sur les techniques utilisées dans ce domaine, et de retracer son développement, notamment avec l'apparition des nouveaux modèles de réseaux de neurones profonds.

## 1.2 Imagerie satellite

Une image satellite, ou satellitaire, est une prise de vue transmise d'un satellite artificiel en orbite. La nouvelle génération des satellites, alliée à la révolution informatique, a fait de la télédétection un outil puissant pour l'analyse, la cartographie et la gestion de l'espace.

Le spectre d'observation est vaste : optique, radar, infrarouge, ultraviolet etc. La résolution atteint actuellement moins d'un mètre pour certaines gammes de fréquences. Celle-ci dépend de la technologie employée et de l'altitude.

L'observation de la terre se fait dans un but scientifique, économique ou militaire. La télédétection par satellite peut fournir des renseignements précis et cohérents sur la surface de la Terre pour une surveillance rentable et rapide des changements environnementaux. Le caractère répétitif des données multi-temporelles et multi-bandes, permet le suivi de l'évolution de beaucoup de phénomènes étudiés (déforestation, expansion urbaine ...) et l'une de ces applications est la DC dans différents milieux : urbains, maritimes, agricoles et forestiers.

### 1.2.1 Types de résolutions en télédétection

Une image numérique est une matrice de pixels (éléments de taille fixe) dont chacun ayant une valeur numérique  $f_{ij}$  où  $(i, j)$  représente les coordonnées spatiales de ce pixel et  $f_{ij}$  son niveau d'intensité lumineuse, comme illustré dans la Figure 1.1.

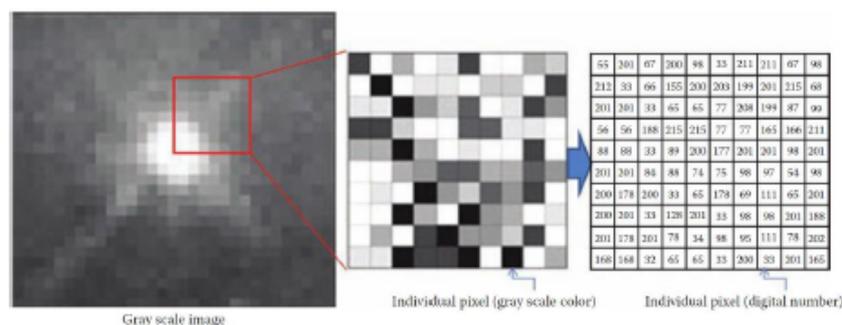


FIGURE 1.1 – Exemple d’une image numérique et sa représentation en matrice de pixels  
Toute image produite par un capteur présente des caractéristiques uniques selon la résolution de ce dernier, qu’elle soit Spatiale, Spectrale, Radiométrique ou Temporelle.

### 1.2.1.1 Résolution Spatiale

La résolution spatiale correspond à la plus petite taille d’un objet ou détail pouvant être représenté dans une image. Elle est souvent définie par l’ouverture du capteur et la plateforme d’altitude (qu’elle soit “*airborne*” ou “*spaceborne*”)

Souvent, pour une application qui requiert une observation globale de la couverture terrestre, une faible résolution est suffisante à cela. Quant aux observations locales, il est nécessaire d’avoir une résolution élevée afin de percevoir plus de détails.

### 1.2.1.2 Résolution Spectrale

Dans les satellites, les capteurs mesurent les radiations Électromagnétiques des différentes portions du spectre électromagnétique. La résolution spectrale est la capacité d’un capteur à distinguer entre les différents intervalles de longueurs d’ondes dans le spectre électromagnétique, et donc représente le nombre de bandes que ce capteur peut détecter.

Cette résolution est le plus souvent utilisée dans des applications de “*mapping*” ou de “*modeling de propriétés biophysiques*” (qualité de l’eau, vigueur des plantes, ..), de sorte que la combinaison de ces bandes de différentes manières, présente l’avantage d’améliorer l’information spectrale de l’image et de rendre certaines caractéristiques et informations biophysiques de la surface de la terre plus discernables.

### 1.2.1.3 Résolution Radiométrique

La résolution radiométrique est la capacité d’un capteur à reconnaître de petites différences dans l’intensité de l’énergie électromagnétique reçue. Plus il en est sensible et plus la résolution est fine.

### 1.2.1.4 Résolution Temporelle

La résolution temporelle représente le temps que prend un satellite pour faire un cycle orbital complet, et pouvoir observer de nouveau le même endroit à partir d’un point de l’espace.

## 1.3 Détection de changement

La DC consiste à identifier les différences d’état d’un objet ou d’un phénomène, en l’observant à différents moments. En télédétection, le processus de détecter un changement

est établi, en observant les différences entre un certain nombre d'images multi-temporelles prises sur le même domaine.

La DC utilisant des images de télédétection multi-temporelles, joue un rôle crucial dans de nombreux domaines d'application, comme la surveillance et la cartographie urbaines, la déforestation, la surveillance de l'environnement et l'évaluation post-catastrophe etc.

Il n'existe pas de méthode unique qui puisse traiter tous les types de problèmes liés à la DC. Différentes applications nécessitent différentes approches, et différents types de données de télédétection exigent des considérations spécifiques.

Dans la littérature, il existe principalement deux types de DC en imagerie satellitaire, à savoir la DC simple et la DC sémantique.

### La détection de changement simple

La DC simple correspond à l'ensemble des méthodes visant à détecter tous les changements dans une zone géographique donnée, à partir d'une paire ou d'une séquence d'images prises à des dates différentes. Ces méthodes ont pour finalité, de générer une image étiquetée où chaque pixel lui est attribué un label binaire correspondant à « changé » ou « inchangé », sans distinguer les catégories de changements. Cependant, ces méthodes ciblent des changements d'intérêts et se réfèrent généralement à des cartes de changements binaires dites « *Ground Truth* », comme le montre la Figure 1.2



FIGURE 1.2 – Exemple de DC simple

### La détection de changement sémantique

La DC sémantique vise à classer les différents changements entre deux ou plusieurs images satellitaires multi-temporelles de la même zone, dans différentes catégories, on cite comme exemples : les changements routiers, naturels et les changements liés à la construction et au développement urbain.

La Figure 1.3 présente un exemple de deux images bi-temporelles et un résultat de DC sémantique :



FIGURE 1.3 – Exemple de DC sémantique

### 1.3.1 Approches de la détection de changements

Les techniques de la DC pour les images satellitaires ont évolué avec les domaines de la vision par ordinateur et le traitement d'images.

Dans un premier temps, les approches basées sur les pixels étaient utilisées en employant des techniques conçues manuellement. Après cette première phase, des descripteurs ont été utilisés dans des applications d'apprentissage standard (PCA-based method, algorithme K-means, image differencing etc). Les descripteurs conçus manuellement se sont avérés insuffisants pour atteindre une haute précision. Avec une plus grande disponibilité de données satellitaires et un avancement colossal dans le domaine de l'intelligence artificielle et de la vision par ordinateur, des techniques d'apprentissage profond plus élaborées omettent ce problème en apprenant de façon hiérarchique. Les CNNs ont permis un grand avancement dans ce domaine, et ont amélioré le résultat de la détection sur des images satellitaires à très haute résolution.

Dans les dernières années, le développement des réseaux de neurones convolutifs, des réseaux de neurones particulièrement adaptés à l'analyse de l'image, a énormément aidé à faire évoluer le domaine de la vision par ordinateur, et ce pour de multiples applications (classification d'images, segmentation, reconnaissance d'objets, etc. . . ). La DC en télédétection est l'un des domaines qui a bénéficié de ces dernières avancées, et l'extraction de *features* de différentes couches a permis l'obtention de nouveaux meilleurs résultats [3]

#### 1.3.1.1 Méthodes basées sur le pixel

Ces méthodes utilisent le pixel comme unité de référence pour la comparaison entre les images, sans prendre en compte le contexte spatial. Toutes les anciennes méthodes de DC et d'analyse d'images en général, prenaient le pixel individuel comme unité de base d'analyse. On cite parmi ces méthodes de DC : la différenciation des images, l'analyse en composantes principales (PCA), le rationnement d'images.

Pour ces méthodes, le pixel comme unité d'analyse est moins approprié pour la DC dans des images satellitaires [2] , car les pixels pourraient être une source d'erreur géométrique, ont des caractéristiques de classification comparables limitées [4] et produisent des changements bruyants et parasites.

A cause de ces limitations, d'autres approches ont été développées, afin de pouvoir mieux définir les changements d'intérêts. Cependant, des méthodes basées sur le pixel et les réseaux de neurones profonds, notamment les convolutifs, ont pu démontrer une efficacité et une robustesse supérieur, ainsi qu'une simplicité d'implémentation.

#### 1.3.1.2 Méthodes basées sur les noyaux

Ces méthodes visent à bénéficier des informations contextuelles, en utilisant les pixels du voisinage. En considérant un voisinage local, l'effet du bruit est réduit. De plus, la texture et la relation contextuelle entre les zones sont exploitées. En effet, ces méthodes sont efficaces pour filtrer le bruit et détecter les changements d'intérêt. Cependant, elles conduisent à des limites floues et à des détails éliminés. Par conséquent, des méthodes basées sur les objets ont vu le jour.

### 1.3.1.3 Méthodes basées sur les objets

La disponibilité croissante de l'imagerie à haute résolution spatiale, a conduit le domaine de DC à recourir vers des méthodes basées sur les objets. La plupart des algorithmes de DC et les méthodes de classifications usuels, prennent le pixel comme base d'analyse, ceci conduit souvent à une faible précision.

Dans les méthodes basées sur les objets, l'information contextuelle est généralement exploitée dans la phase de segmentation. Différentes méthodes de DC basées sur les objets ont été proposées [5] [6], offrant des approches uniques pour exploiter les images THR et une possibilité de la DC sémantique [7].

### 1.3.1.4 Méthodes hybrides

Les méthodes hybrides sont des méthodes dans lesquelles les unités d'analyse sont une combinaison. Une approche plus élémentaire est de se baser sur les pixels pour la comparaison et d'ensuite améliorer le résultat en utilisant des techniques de filtrage ou de segmentation [8]. D'autre part, des méthodes de combinaison de classifications basées sur les pixels et sur les objets ont démontré une grande performance [9].

### 1.3.1.5 Méthodes de fusion de données

La fusion de données dans l'analyse de la télédétection, inclut la fusion d'images satellitaires avec différentes résolutions spatiales et spectrales, mais aussi la fusion de divers types d'informations extraites de la même source. La fusion de différentes variables de changement extraites du même ensemble de données multitemporelles, à l'aide de différents opérateurs de comparaison, a permis l'amélioration de la qualité de l'indicateur de changement final. Elle est notamment utilisée dans des méthodes basées sur les CNNs [10] [6].

## 1.3.2 Modes de comparaison

La DC est souvent approchée par l'un de ces deux modes :

### Comparaison Post-Classification

Ce mode consiste à classifier le contenu de deux images bi-temporelles indépendamment, pour ensuite comparer les cartes de classification pour y identifier les différences. Cependant, cette approche est sujet de résultats erronés suite aux erreurs dues à l'étape de classification dans l'une des deux images, d'où la nécessité d'avoir un taux de précision élevé pour pouvoir recourir à cette méthode.

### Analyse comparative

Cette approche est basée sur une analyse comparative pour construire une image de différence (DI) ou une carte de changement qui met en évidence les différences existantes entre deux images d'une même scène à des temps différents. La nature de ces changements est déterminée en analysant l'image résultante de la comparaison. La précision des résultats de la DC dépendra de la précision de la méthode de comparaison et de l'interprétation.

## 1.4 Revue de la littérature

En télédétection, les méthodes conventionnelles pour la DC (différence d'images, rationnement d'images ...) sont des techniques d'analyses qui ne donnent pas de résultats

satisfaisants, notamment sur des images satellitaires à haute résolution contenant beaucoup de détails. Avec l'avènement de l'apprentissage automatique et la reconnaissance de formes, les performances des modèles de la DC basés sur l'apprentissage machine et les réseaux de neurones profonds ont augmenté en comparaison avec les modèles classiques. Beaucoup des travaux existants, notamment les travaux récents, dans la littérature se basent sur le développement de méthodes d'apprentissage ou d'apprentissage profond.

### 1.4.1 Méthodes basées sur le Machine Learning

Le Machine Learning a révolutionné le domaine de la DC. Entre images satellitaires ou photographies simples, le ML a permis d'atteindre de très hauts niveaux de précision. Les algorithmes de "*K-means clustering*", "*fuzzy clustering*", "*Markov Random Field framework*" ou l'utilisation du classifieur SVM pour la classification des zones d'une image sont des méthodes fréquemment utilisées dans la tâche de la DC.

Des méthodes de DC indirectes sont souvent adoptées. On retrouve dans cette catégorie le travail de (Karthik et al. 2017) [11] qui consiste à utiliser un classifieur SVM pour la classification des images bi-temporelles en plusieurs classes : eau, forêts, terres vierges etc, afin de réaliser une différence d'image (DI) entre les deux résultats. Cette méthode a atteint une précision moyenne de 85.2% et est validée sur une paire d'images cartographiques.

### 1.4.2 Méthodes basées sur les réseaux de neurones profonds

Après le succès empirique des réseaux de neurones profonds et l'apprentissage profond dans le domaine de la vision par ordinateur, et la disponibilité croissante des données satellitaires, le recours au Deep Learning devient plus fréquent dans le domaine de la DC. En effet, de nouvelles approches supervisées, non supervisées et semi-supervisées de DC sont apparues sur la base de cette technologie.

Les méthodes basées sur les réseaux de neurones profonds, notamment les CNNs, sont souvent développées en introduisant des techniques d'apprentissage machine comme l'analyse en composantes principales [6] ou le classifieur SVM [12].

Le modèle développé par Larabi Mohammed et al. 2016 [3], fut le premier ayant utilisé les *features* extraits des CNNs, sans recours à l'entraînement. Ce travail, testé sur des images THR, a démontré l'efficacité des CNNs et a surpassé en précision plusieurs méthodes conventionnelles, notamment une méthode basée sur l'analyse en composantes principales.

Depuis 2016, presque tous les travaux sur la DC en télédétection se basent sur les réseaux de neurones profonds, à savoir les CNNs, les réseaux de neurones récurrents et les CNNs modifiés. Ces méthodes ont prouvé leur capacité à augmenter considérablement la précision et la fiabilité des systèmes conçus pour détecter les changements en imagerie satellitaire.

### 1.4.3 Applications des réseaux de neurones convolutifs « CNN » à la détection de changement en Imagerie satellitaire

Une multitude de méthodes et de stratégies sont appliquées pour améliorer les performances de la DC en imagerie satellite. Ces dernières années, les CNNs sont la base de beaucoup de ces méthodes, ayant atteint les meilleures performances.

#### 1.4.3.1 Utilisation des CNNs comme extracteurs de caractéristiques

Larabi et al. [3] ont développé une nouvelle méthode de DC basée sur l'extraction de caractéristiques à partir de plusieurs niveaux du CNN. Ce fut la première approche basée sur les CNNs en DC dans les images satellites de haute résolution. Cette méthode consiste à utiliser un CNN pré-entraîné sur un grand ensemble de données (ImageNet) pour une tâche de classification, afin d'extraire les caractéristiques de la paire d'images à comparer à travers différentes couches convolutives. Ensuite, vient une étape de normalisation et de concaténation des caractéristiques, ce qui permet d'obtenir une carte de caractéristiques de dimension supérieure. La carte de changement finale binaire est obtenue en calculant la distance euclidienne au niveau du pixel, puis en appliquant une méthode de seuillage.

Une autre méthode de DC non supervisée est présentée dans [13] en 2017. L'approche proposée se base sur l'apprentissage par transfert ; un réseau de neurones convolutif VGG-16 performé sur la base de données Imagenet est utilisé avec une technique de segmentation par superpixels en utilisant l'algorithme SLIC. La fusion de ces deux techniques pour la DC a été auparavant introduite dans les travaux de Ken Sakurada et al. [5] pour la DC sur des images bi-temporelles sur la base de données TSUNAMI. L'utilisation des superpixels permet de diminuer l'effet négatif du bruit sur la classification. Elle demande cependant plus de temps d'exécution que la simple extraction de caractéristiques, et ajoute des données redondantes qui pourraient nuire à la précision de la méthode. Pour y remédier des caractéristiques de zoom out ont été introduits [14] , cette technique permet l'extraction de caractéristiques de différents niveaux à proximité d'un superpixel et contribue à la décision de son classement.

Dans son étude, Sudipan Saha [15] propose d'aborder la DC par une approche *context-sensitive*, pour des images THR multi-temporelles et ce, en ayant recours à l'extraction de fonctionnalités basées sur des CNNs pré-entraînés. Cette méthode non supervisée utilise l'approche Deep Change Vector Analysis (DCVA), afin de comparer deux images par l'unité du pixel et tirer les changements. La précision globale (CA) moyenne lors des tests sur la base de données WorldView-2 est aux environs de 97.67%. L'approche DCVA pour la DC à partir d'images THR multi-temporelles, a donc pu démontrer son efficacité à prendre en compte le contexte spatial en utilisant des CNNs pré-entraînés et ce, de manière non-supervisée et sans dépendre de la segmentation d'image.

Récemment, la DC sémantique pour l'imagerie satellitaire est devenue un centre d'intérêt pour beaucoup de chercheurs dans le domaine de la vision par ordinateur. Parmi les approches entreprises, les CNNs sont particulièrement utilisées pour atteindre ce but. De Jong et al. [7] propose une méthode non supervisée, utilisant un réseau de neurones U-Net pré-entraîné pour une tâche de segmentation sémantique. La technique de différenciation d'images est utilisée pour la DC, pour ce fait, les DIs sont générées à partir des *features maps* des deux images. La DI permet au décodeur après une opération de sur-échantillonnage de retrouver les changements. Ces derniers peuvent également être classés en des classes sémantiques qui fournissent des informations sur la nature du changement.

### 1.4.3.2 La stratégie de l'apprentissage par transfert par réglage des CNNs

L'apprentissage par transfert est largement utilisé dans le domaine de la DC au vu du manque de larges bases de données. Mohammed El Amin Larabi [16] adresse le problème de DC en se basant sur les techniques de transfert où il utilise un modèle CNN pré-entraîné, pour un réglage partiel entraîné dans le but de classifier deux patches d'entrées en "changé" ou "inchangé".

Cette méthode a montré sa capacité à *fine-tuning* (section 2.3.4.2) un modèle CNN pré-entraîné sur de petits datasets et à avoir une précision d'environ 83% pour le modèle VGG-16.

Dans [10], un ensemble de réseaux de neurones pré-entraînés est employé pour la tâche de DC. Le réseau pré-entraîné sur ImageNet VGG-16 est utilisé sans les couches fully-connected. En se basant sur cette architecture, trois autres réseaux sont formés en modifiant le décodeur à chaque fois. L'encodeur permet d'obtenir la carte de caractéristiques et le décodeur permet d'obtenir une carte de segmentation qui représenterait la carte de changement. L'entraînement supervisé se fait sur une base de données créée par les auteurs, en ayant manuellement conçu les « *ground truth* » ; quelques paramètres sont initialisés et gelés avec les données du réseau pré-entraîné. La carte de changement finale est obtenue en fusionnant les cartes de sorties des trois réseaux, plus précisément en établissant un seuil pour la moyenne des trois cartes de segmentation.

La méthode proposée en [10], démontre l'efficacité des CNNs en DC sans l'usage de méthodes de prétraitement comme l'orthorectification.

### 1.4.3.3 Entraînement à zéro des CNNs

On retrouve également des travaux de DC utilisant des CNNs entraînés à partir de zéro (en anglais : *from scratch*). En premier lieu, on cite les travaux de Rodrigo Caye Daudt et al. [17], où deux approches sont présentées : la première architecture proposée est la fusion précoce « EF » (en anglais : *Early Fusion*), elle consiste à concaténer les deux images en entrée d'un FCN. La dernière couche est un Softmax avec deux sorties associées aux classes : « changement » et « pas de changement ». La deuxième architecture est une architecture siamoise, acceptant en entrée les deux images. Les données utilisées dans [17], pour l'entraînement à partir de zéro est la base de données conçue et présentée par les auteurs OSCD.

L'utilisation des réseaux siamois pour un entraînement basé sur les patches (en anglais *patch based CNN*) a aussi été introduit pour résoudre la problématique de la DC [16]. L'idée est de traiter chacun des patches en parallèle et de concaténer les sorties. Une autre alternative basée sur cette approche a aussi été présentée en 2018 dans le travail de Faiz Ur Rahman et al [18].

Une proposition supplémentaire à noter est une solution de DC sémantique présentée en 2019 [19]. La méthode consiste à utiliser les CNNs pour la segmentation sémantique en s'appuyant sur le modèle « Deeplab v3+ semantic segmentation model » [20], avec un autre mode de convolution ASPP (en anglais : *Atrous Spatial Pyramid Pooling*), appliquant des filtres avec un taux de dilatation pour réduire les pertes en résolution. La DC finale se fait en calculant les pourcentages de chaque classe pour les images bi-temporelles, à partir des cartes de segmentations sémantiques. C'est donc une DC indirecte.

#### 1.4.3.4 Comparaison des méthodes de ML et DL pour la détection de changement

L'apprentissage automatique ainsi que l'apprentissage automatique profond sont aujourd'hui les piliers du domaine de la DC en imagerie satellitaire. Les deux approches présentent des avantages et des inconvénients.

D'abord, l'utilisation des algorithmes d'apprentissage automatique classiques nécessite une phase d'extraction des caractéristiques au préalable. La phase d'extraction de caractéristiques est utilisée pour améliorer la précision des images, notamment celle des images multi-spectrales. Dans le cas de l'apprentissage profond, la phase d'extraction de caractéristiques séparée n'est pas toujours nécessaire, on cite les méthodes basées sur les CNNs et les réseaux de neurones récurrents comme exemple.

D'autre part, l'apprentissage automatique donne des résultats peu satisfaisants pour les données les plus complexes telles que les images satellitaires des zones urbaines. Il nécessite également un prétraitement des données et des rectifications, ce qui n'est pas toujours le cas avec l'apprentissage automatique profond. Cependant, les réseaux de neurones profonds requièrent une très large quantité de données, ce qui n'est pas aisément trouvé, et peuvent être très coûteux en puissance de calcul à cause de la profondeur des réseaux utilisés.

Le travail de T. Vignesh et al [21] compare différentes méthodes d'apprentissage automatique classique et profond, et conclut d'après les mesures de précision globale validées sur cinq types de données différentes, que les performances en DC sont notablement plus élevées avec les solutions fondées sur l'apprentissage profond. On note que les résultats de son travail sont particulièrement bons avec l'utilisation des réseaux de neurones à convolution.

La Figure 1.4 ([21]) présente la précision globale moyenne obtenue pour les différentes données.

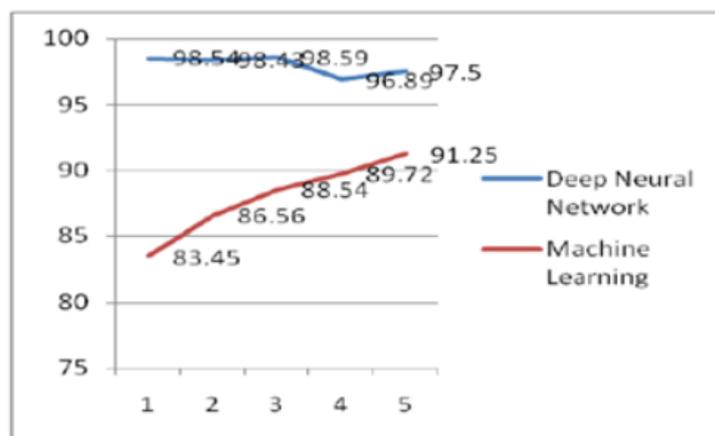


FIGURE 1.4 – Performance des algorithmes ML et méthodes DL pour la tâche de la DC en télédétection

## 1.5 Challenges et problématiques liés à la détection de changement en télédétection

La DC en imagerie satellite est complexe. Elle rencontre plusieurs problèmes d'application en raison du bruit et des distorsions liées à l'imagerie, les conditions atmosphériques et d'humidités à des dates différentes, ainsi que l'orientation de la lumière et la nécessité de distinguer les changements naturels des changements provoqués par les activités humaines, qui sont le plus souvent les changements ciblés en DC.

En général, indépendamment de la zone étudiée, une méthode de DC doit faire la distinction entre les changements non pertinents et les changements d'intérêt. Pour minimiser l'impact des facteurs cités précédemment, il est important d'utiliser des données provenant des mêmes capteurs, avec les mêmes caractéristiques de résolutions, et envisager d'acquérir les données pour des dates quasi identiques, afin d'éliminer les effets des changements de saisons, rayonnements solaires et autres contraintes [6].

Des corrections et du pré-traitement ; comme le réglage de luminosité, élimination du bruit, correction géométrique telle l'orthorectification pour les variations d'orientation et les angles pourraient être nécessaires pour diminuer les effets cités, et faciliter l'opération de la DC par la suite. Cependant, des méthodes récentes ont démontré que l'utilisation des CNNs pourrait donner des résultats satisfaisants sans pré-traitement.

Malgré l'apport gigantesque apporté au domaine de la DC en imagerie satellitaire par l'exploitation des réseaux de neurones profonds, d'autres challenges liés à la problématique de la DC sont apparus. En effet, l'efficacité des réseaux de neurones profonds est fortement liée à la disponibilité de grandes bases de données étiquetées. Or, des bases de données d'images bi ou multi temporelles ne sont pas disponibles en grand nombre. De plus la création des cartes de changement, qui est très coûteuse en temps, est parfois nécessaire pour l'apprentissage supervisé.

Un autre obstacle notable lié à l'apprentissage supervisé en DC, est la représentation des cartes de changements binaires (GT) utilisées comme références pour l'apprentissage et les tests. Ces cartes jouent un rôle crucial dans la conception des systèmes supervisés et dans leur évaluation, et peuvent paradoxalement être une source de problème quant à la fiabilité des systèmes de DC, car pour une même paire d'images bitemporelles, les GT peuvent différer d'un travail à un autre, pour des raisons d'une différence de changement cible à détecter, ou tout simplement pour des raisons relatives à la subjectivité des constructeurs des bases de données.

## 1.6 Conclusion

Ce chapitre a introduit le domaine de la DC en utilisant des images d'observation de la terre. A partir de l'exposition des techniques utilisées en DC, ainsi que la revue de la littérature récente, nous pouvons déduire que le développement des technologies récentes et les méthodes d'apprentissage profond a permis aux systèmes de DC en télédétection d'être plus robustes et précis que les systèmes fondés sur l'analyse statistique et spectrale traditionnelle.

Le chapitre a aussi énoncé les nombreuses problématiques et challenges auxquels le domaine de la DC fait face, et ce malgré les récentes avancées dues à l'apprentissage machine profond.

Enfin, nous concluons ce chapitre pour dire que la problématique de la DC en imagerie satellitaire reste encore aujourd'hui un domaine de recherche active, notamment avec l'introduction des réseaux de neurones profonds, dans le but d'adapter les nouveaux systèmes de sorte à pouvoir contrer certaines contraintes existantes, éventuellement être capable de s'adapter à des données disponibles de différents capteurs, avec une supervision minimale de l'utilisateur.

## Chapitre 2

# Réseaux de Neurones Convolutifs

## 2.1 Introduction

Cette dernière décennie, la sous branche de l'Intelligence Artificielle qu'est l'apprentissage profond a révolutionné le domaine de la vision par ordinateur en introduisant les réseaux de neurones convolutifs dits CNNs (*Convolutional Neural Network*). L'une des premières architectures des CNNs est LeNet5, introduite en 1998 par Yann LeCun [22] pour la reconnaissance de caractères manuscrits, dépassant en performance toutes les méthodes traditionnelles.

Entre 1998 et 2010, les CNNs étaient en incubation. Puis grâce à la double augmentation continue de données numériques et de la puissance de calcul avec l'avènement des processeurs graphiques, les CNNs ont pu retrouver leur place en tant que technique de premier plan dans le domaine de la vision par ordinateur. Le défi de classification annuel basé sur la base de données « ImageNet » [23] a joué un rôle majeur dans l'amélioration des résultats. Le réseau AlexNet, introduit par Alex Krizhevsky et al en 2012 [24] marqua ce point de retour en remportant la compétition ILSVRC-2012.

Les performances des systèmes de DC en télédétection ont également augmenté avec l'utilisation des CNNs. L'étude comparative citée dans la section 1.4.3.4 démontre la supériorité des performances des CNNs pour la résolution des problématiques liées à la DC.

Pour notre travail, nous proposons des méthodes de DC basées sur les CNNs, d'abord comme extracteur de caractéristiques, ensuite pour un système de DC end-to-end. Ce chapitre est donc consacré à expliquer le fonctionnement d'un réseau de neurone artificiel en premier lieu, pour ensuite introduire les CNNs, exposer leur composition détaillée, expliquer l'entraînement ainsi que les stratégies révolutionnaires de l'apprentissage par transfert. Enfin, nous concluons ce chapitre en exposant des modèles de CNNs les plus connus et utilisés dans le domaine de la vision par ordinateur et qui seront adaptés à notre travail par la suite. Ceci permettra d'entamer aisément le chapitre suivant.

## 2.2 Réseaux de Neurones Artificiels

Les réseaux de neurones artificiels sont un système informatique inspiré du fonctionnement des neurones biologiques, dont le premier modèle fut conçu en 1943 par McCulloch et Pitts.

Ce système utilisé pour résoudre divers problèmes statistiques, notamment des problèmes de classification, est composé de neurones généralement répartis en plusieurs couches connectées entre elles ; la sortie d'une couche correspond à l'entrée de la suivante. Chaque neurone reçoit des informations numériques en provenance de neurones voisins, à chacune de ces valeurs est associée un poids représentatif de la force de la connexion. Chaque neurone effectue localement un calcul dont le résultat est transmis ensuite aux neurones avants. Cet empilement de couches définit la sortie finale du réseau comme le résultat d'une fonction différentiable de l'entrée. La dernière couche calcule les probabilités finales en utilisant une fonction d'activation.

### 2.2.1 Le perceptron

Le perceptron est le réseau de neurones le plus simple. Inventé par Rosenblatt, il est formé d'un seul neurone formel tel que présenté dans la Figure 2.1 [25]. Il est constitué d'un nombre indéfini d'entrées et d'une sortie unique, ce qui permet de résoudre des problèmes à caractère linéaire.

Le neurone procède par une somme pondérée de son vecteur d'entrées  $(x_1, x_2, \dots, x_n)$  et de son vecteur de poids synaptique  $(w_1, w_2, \dots, w_n)$ . On y ajoute ensuite le biais  $b$  comme le montre la formule (2.1).

$$S = \sum_{i=0}^n w_i x_i + b \quad (2.1)$$

Enfin, le résultat est produit via une fonction d'activation  $\sigma$  (section 2.2.3.2) et est comparé à un seuil. La sortie prédite du modèle  $\hat{y}$  est donnée par (2.2)

$$\hat{y} = \sigma(S) \quad (2.2)$$

Le perceptron présente deux problèmes majeures :

- Il est incapable de résoudre des problèmes avec une multitude de classes.
- Il n'est pas adapté à séparer des données réparties d'une manière non linéaire.

Ceci a conduit à l'apparition du réseau de neurones multicouches MLP.

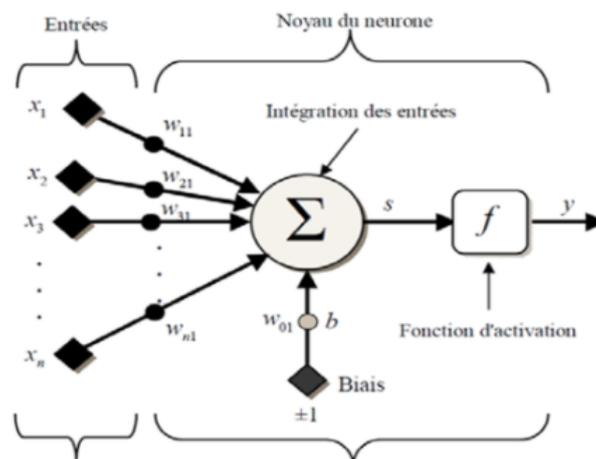


FIGURE 2.1 – Modèle d'un neurone formel (Razafimandimby, Loscri et al. 2016)

### 2.2.2 Perceptron multicouches MLP

Dans un réseau de neurones MLP, les neurones sont arrangés par couche constituant au moins trois couches : une couche d'entrée, une couche de sortie et une ou plusieurs couches cachées. Les neurones d'une même couche n'ont pas de connexion entre eux, et chaque neurone d'une couche est connecté à tous les autres neurones de la couche suivante.

La Figure 2.2 est une représentation d'un MLP à une seule couche intermédiaire.

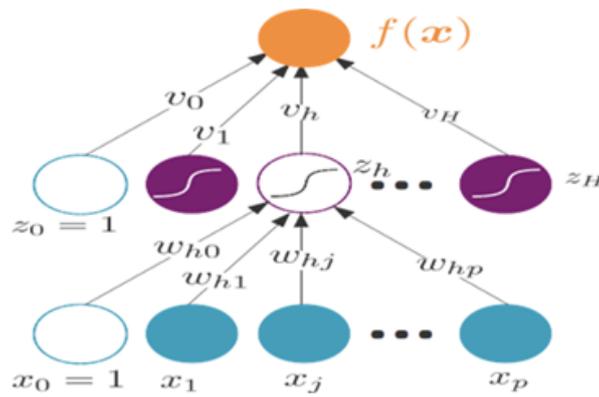


FIGURE 2.2 – MLP à une couche intermédiaire

Dans un MLP, empiler les couches cachées permet d'avoir un réseau avec une performance d'apprentissage très élevée, désigné par le nom « réseau de neurones profond », et c'est de là qu'est apparu le DL.

### 2.2.3 Apprentissage d'un MLP

L'apprentissage d'un réseau de neurone (simple ou profond), revient à estimer le poids lié à chaque neurone de telle sorte que ça minimise une fonction de perte choisie. Parmi les méthodes les plus célèbres et les plus efficaces dans l'apprentissage automatique, notamment dans l'apprentissage profond, on trouve la méthode de la rétropropagation.

Définie par Sapna, Tamilarasi et al. [26], comme une technique d'apprentissage supervisé basée sur la méthode de descente de gradient, dont l'objectif est de minimiser l'erreur du réseau en se référant à une fonction de coût  $J(w, b)$  défini par (2.3). En minimisant la fonction de coût, on obtient les paramètres (poids et biais) optimaux pour les meilleures performances du modèle.

$$J(w, b) = L(y, \hat{y}) \quad (2.3)$$

avec :

- $L$  : fonction de perte (section 2.2.3.1).
- $y$  : la véritable sortie.
- $\hat{y}$  : la sortie prédite par le modèle.

L'algorithme de la descente du gradient est un algorithme d'optimisation de premier ordre, c'est-à-dire qu'il ne prend en compte que la première dérivée lors de la mise à jour des paramètres. Il s'agit de mettre à jour les paramètres à chaque itération, dans le sens opposé du gradient de la fonction de coût  $J(w, b)$  par rapport aux paramètres où le gradient donne la direction de la montée la plus raide.

La phase d'entraînement se constitue de deux phases : Une phase '*Forward*' dans laquelle les sorties des couches intermédiaires sont mises à jour, et une phase '*Backward*', dans laquelle le gradient du coût par rapport aux poids d'une couche est calculé à partir du gradient du coût par rapport aux poids d'une couche supérieure, et ainsi de suite, en commençant par calculer le gradient des sorties de la dernière couche et à itérer jusqu'à la première couche du réseau. La mise à jour des poids se fait suivant (2.4) pour chaque couche  $l$  et dans le but de minimiser la fonction de coût. ( $\alpha$  ici représente le taux d'apprentissage *learning rate*).

$$w_{i,j}^l \leftarrow w_{i,j}^l - \alpha \frac{dJ(w, b)}{dw_{i,j}^l} \quad (2.4)$$

### 2.2.3.1 Fonctions de pertes

La ‘Fonction de Pertes’ (*Loss Function*) est une fonction qui quantifie l’écart entre les prévisions du modèle et les observations réelles du jeu de donnée utilisé lors de l’apprentissage. A partir de cette fonction, le processus d’apprentissage ajustera les paramètres  $w_{ij}$ , de sorte à minimiser cette fonction de perte et avoir un modèle optimal.

Il existe différentes fonctions de pertes utilisées en réseau de neurones, les plus connues sont :

— **L’erreur quadratique moyenne (*Mean Square Error*)**

L’erreur quadratique moyenne est la somme des différences au carré des prédictions du modèle et des observations réelles.

Sa formule est donnée par l’équation (2.5) :

$$MSE = \frac{1}{N} \sum_{(x,y) \in D} (y - prediction(x))^2 \quad (2.5)$$

Où :

- $x$  : ensemble de caractéristiques utilisé par le modèle pour effectuer des prédictions.
- $y$  : la véritable sortie
- $Prediction(x)$  : est la sortie prédite par le modèle pour l’ensemble  $x$ .
- $D$  : ensemble de données étiquetées contenant  $(x, y)$ .
- $N$  : nombre d’exemples dans  $D$ .

— **L’entropie croisée (*Cross Entropy*)**

La perte d’entropie croisée, ou la perte algorithmique, est liée à des mesures de divergences. Elle mesure généralement la performance de modèles de classification, dont la sortie est une probabilité entre 0 et 1. En apprentissage automatique, la fonction utilisée pour un cas d’une classification multi classes est appelée "*Categorical Crossentropy*", et "*Binary Crossentropy*" pour un cas d’une classification binaire.

L’entropie croisée est considérée comme plus performante et permet une meilleure généralisation que l’erreur quadratique moyenne pour les problèmes de classification [27].

### 2.2.3.2 Fonctions d’activation

La fonction d’activation est une fonction mathématique appliquée à un signal en sortie d’un neurone artificiel. Elle est attachée à chaque neurone et détermine si elle doit être activée ou non en fonction de l’entrée. Généralement, les fonctions utilisées sont non linéaires, ce qui amène le réseau à pouvoir traiter des données complexes. On définit dans ce qui suit, quelques unes de ces fonctions d’activation [28] :

— **Sigmoïde**

La fonction d’activation sigmoïde (appelée également fonction logistique) est une fonction non linéaire, permettant la rétropropagation car elle dispose d’une fonction dérivée non nulle liée aux entrées. La fonction sigmoïde est représentée dans la Figure 2.3.

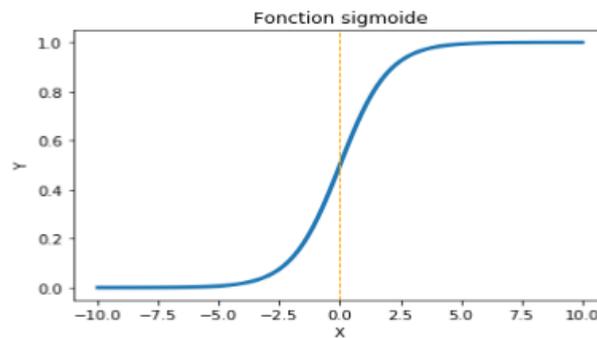
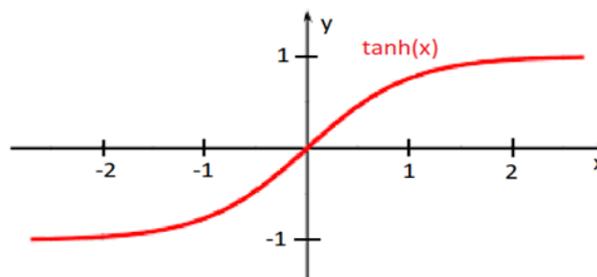


FIGURE 2.3 – Fonction sigmoïde

### — Tangente Hyperbolique

La fonction Tangente Hyperbolique (Figure 2.4) est une variante de la fonction Sigmoïde, La différence avec la fonction Sigmoïde est que la fonction Tanh produit un résultat compris entre -1 et 1. La fonction Tanh est en terme général préférable à la fonction Sigmoïde car elle est centrée sur zéro.

FIGURE 2.4 – Fonction  $\tanh(x)$ 

### — ReLU

La fonction ReLU est interprétée par la formule (2.6)

$$f(x) = \max(0, x) \quad (2.6)$$

Si l'entrée est négative, la sortie est 0 et si elle est positive, alors la sortie est x. Cette fonction d'activation augmente considérablement la convergence du réseau et ne sature pas.

Mais la fonction ReLU n'est pas parfaite. Si la valeur d'entrée est négative, le neurone reste inactif, ainsi les poids ne sont pas mis à jour et le réseau n'apprend pas. La Figure 2.5 donne un aperçu sur la forme de la fonction ReLU.

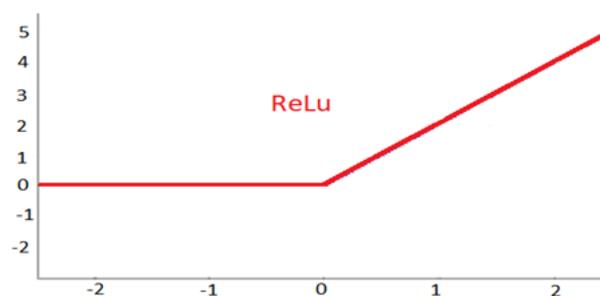


FIGURE 2.5 – Fonction ReLU

### — Leaky ReLU

La fonction Leaky ReLU (Figure 2.6) est interprétée par la formule (2.7) :

$$f(x) = \max(0.1x, x) \quad (2.7)$$

La fonction Leaky ReLU essaye de corriger la fonction ReLU lorsque l'entrée est négative. Son concept est que lorsque l'entrée est négative, il aura une petite pente positive de 0,1. Cette fonction élimine quelque peu le problème d'inactivité de la fonction ReLU pour les valeurs négatives, mais les résultats obtenus avec elle ne sont pas cohérents. Elle conserve tout de même les caractéristiques d'une fonction d'activation ReLU, c'est-à-dire efficace sur le plan des calculs : elle converge beaucoup plus rapidement et ne sature pas dans les régions positives.

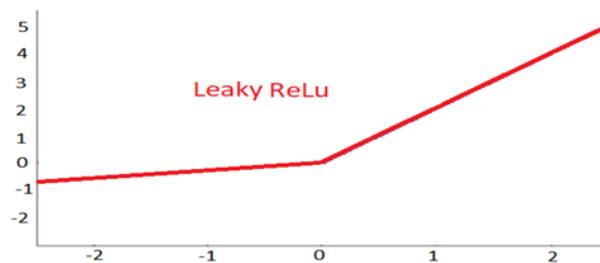


FIGURE 2.6 – Fonction Leaky ReLU

### — Softmax

La fonction Softmax est utilisée dans de nombreux modèles probabilistes, traitant à la fois des données discrètes et continues. Elle produit une sortie qui est sur une plage de valeurs comprise entre 0 et 1. Ainsi, chaque classe de sortie se verra affecter une probabilité, telle que la somme de toutes ces probabilités est égale à 1, et la classe retenue est celle avec la plus grande probabilité.

Avec un nombre de classe égale à  $k$ , la fonction Softmax est donnée par l'équation (2.8) :

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=0}^k e^{x_j}} \quad (2.8)$$

La principale différence entre la Sigmoid et la Softmax est que la Sigmoid est utilisée dans la classification binaire, tandis que la Softmax est utilisée pour les tâches de classification multi classes. La fonction Softmax sert à transformer les valeurs des nœuds de la dernière couche, en probabilité d'appartenance aux différentes classes possibles.

## 2.3 Réseaux de neurones convolutifs

Les CNNs sont une sous-catégorie de réseaux de neurones ; ils présentent donc toutes les caractéristiques citées précédemment. Cependant, ils sont conçus spécialement pour traiter des images en entrée, car leur architecture est construite de telle façon qu'ils prennent en compte la disposition spatiale dans l'image.

Le néocognitron est considéré comme l'ancêtre des réseaux de neurones à convolution. Introduit en 1980 [29], il est le premier réseau à forcer les unités situées en plusieurs positions de partager les mêmes poids. Les CNNs dont l'architecture est inspirée de celui du cortex visuel des vertébrés utilise cette technique et limite le nombre de connexions entre un neurone et les neurones des couches adjacentes, ce qui diminue drastiquement

le nombre de paramètres d'apprentissage [30]. Cette particularité explique la difficulté des MLP à gérer des images de grandes taille, en raison de la croissance exponentielle du nombre de paramètres d'entraînement.

Un CNN présente différentes couches qui œuvrent directement à l'extraction de caractéristiques pouvant décrire au mieux l'image en entrée. Comme il est mentionné dans [31], les CNNs sont largement utilisés pour réaliser des tâches spécifiques, telles que la classification, la segmentation ou la séparation d'images, en raison de leur performance remarquable.

### 2.3.1 Couches d'un CNN

Les réseaux de neurones à convolution présentent une architecture particulière à l'analyse d'image. Ils contiennent plusieurs types de couches, dans le but d'extraire les caractéristiques des images par fragment [32]. Cette sous-section donnera un aperçu sur les différentes couches des CNNs [32], [33].

#### 2.3.1.1 La couche de convolution

La couche de convolution est la composante clé des CNNs. Son rôle est de repérer la position d'un ensemble de caractéristiques dans les images reçues en entrée, et ce en réalisant un filtrage par convolution.

Le principe est de faire "glisser" une fenêtre représentant la caractéristique (*feature*) sur l'image, et de calculer le produit de convolution entre la caractéristique et chaque portion de l'image balayée, la Figure 2.7 explicite cette notion à l'aide d'un exemple avec un filtre de taille 3 x 3. Une caractéristique est alors vue comme un filtre : les deux termes sont équivalents dans ce contexte. En sortie, le résultat pour chaque paire (image, filtre) est une carte d'activation (en anglais : *feature map*), qui nous indique où se situent les caractéristiques dans l'image : les valeurs élevées indiquent la localisation de cette caractéristique dans l'image.

Contrairement aux méthodes traditionnelles, les caractéristiques ne sont pas prédéfinies comme c'est le cas avec les descripteurs, mais apprises automatiquement par le réseau lors de la phase d'entraînement, en s'adaptant au problème posé. Les noyaux des filtres désignent les poids de la couche de convolution. Ils sont initialisés puis mis à jour par rétropropagation du gradient.

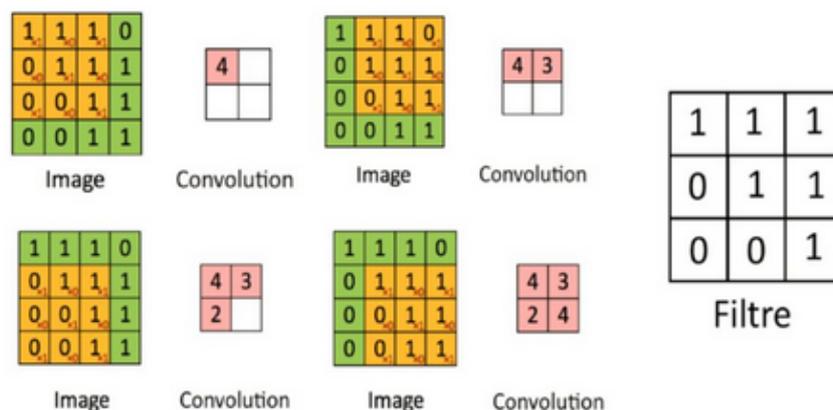


FIGURE 2.7 – Exemple d'une convolution avec un filtre 3 x 3

### 2.3.1.2 Couche Zero Padding

Après avoir procédé à une convolution, la taille de la matrice de caractéristiques se trouve réduite. Etant donné que nous allons enchaîner les convolutions, cette taille ne cessera de diminuer ce qui engendrera :

- Une perte d'informations aux frontières après chaque couche de convolution.
- La diminution de la qualité d'apprentissage.
- La réduction de la taille de la carte de caractéristiques à 1x1 après un certain nombre de convolutions.

Afin de diminuer l'effet négatif de ces derniers points sur l'apprentissage du réseau, la technique de Zero Padding propose d'ajouter une couche de pixels de valeur zéro pour entourer la matrice d'image. De ce fait, les opérations de convolution n'affecteront pas la carte de caractéristiques.

### 2.3.1.3 Couche ReLU

ReLU (*Rectified Linear Units*) désigne la fonction réelle non-linéaire définie par l'équation (2.9)

$$ReLU(x) = \max(0, x) \quad (2.9)$$

Après chaque couche de convolution, il est conventionnel d'appliquer une couche non linéaire. Le but est d'introduire de la non-linéarité dans notre CNN, pour ne pas le limiter à n'apprendre que des problèmes linéaires, étant donné que nous venons de procéder à une convolution qui est une opération linéaire à travers la multiplication et l'addition de matrices par élément. ReLU se chargera de remplacer toutes les valeurs de pixel négatives dans la matrice de caractéristiques par des zéros.

### 2.3.1.4 Couche Pooling

La couche de Pooling reçoit en entrée plusieurs cartes caractéristiques (*feature maps*), et applique à chacune d'entre elles l'opération de *Pooling*. Cette opération consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes.

Les paramètres de l'opération de Pooling sont :

- **La taille de la fenêtre** : Représente la taille de la fenêtre à faire balayer sur la matrice de caractéristique.
- **Le pas du Pooling (*stride*)** : Représente le pas d'avancement après chaque balayage

Pour se faire, nous faisons glisser une fenêtre sur la matrice de caractéristique, et nous créons un élément de la matrice de sortie en fonction des éléments qui se trouvent dans cette fenêtre-là. Les variantes qui peuvent être utilisées sont :

- **Le max Pooling** : Consiste à prendre la valeur maximale dans chaque fenêtre et l'affecter à l'élément en sortie (Figure 2.8).
- **Le min Pooling** : Consiste à prendre la valeur minimale dans chaque fenêtre et l'affecter à l'élément en sortie.
- **La somme Pooling** : Consiste à calculer la somme des éléments présents à l'intérieur de la fenêtre et affecter le résultat à l'élément en sortie.
- **La moyenne Pooling** : Consiste à calculer la moyenne des éléments présents à l'intérieur de la fenêtre et affecter le résultat à l'élément en sortie.

On note que le Pooling le plus utilisé est le *Max Pooling*.

Les avantages de l'opération « *Pooling* » se résument en :

- Réduire le temps d'apprentissage.
- Réduire le nombre de paramètres et de calculs dans le réseau.
- Rendre le réseau invariant par rapport aux petites transformations et dans l'image d'entrée, ce qui par conséquent aide à éviter le sur-apprentissage.

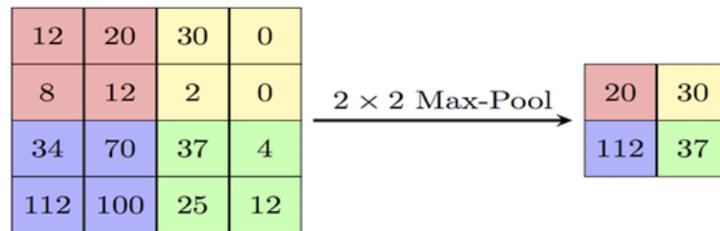


FIGURE 2.8 – Exemple d'une opération de Max-Pooling 2 X 2

### 2.3.1.5 Couche Flattening

La couche Flattening, comme le montre la Figure 2.9, consiste en l'aplatissement des matrices de caractéristiques obtenues en un vecteur à une dimension afin d'adapter les caractéristiques extraites à l'entrée de la couche FC.

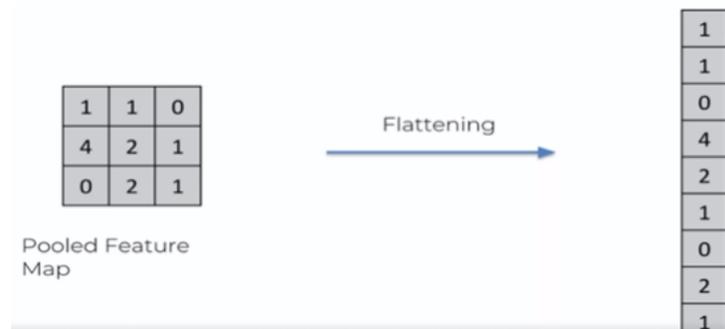


FIGURE 2.9 – Exemple d'une opération de Flattening

### 2.3.1.6 Couche complètement connectée (*Fully-connected*)

La couche FC constitue la dernière partie de tout réseau de neurones, elle n'est pas caractéristique d'un CNN.

Cette couche reçoit en entrée le résultat de l'aplatissement des matrices de caractéristiques extraites des couches ultérieures du CNN. Les valeurs de ce vecteur d'entrée sont multipliées par des poids et passent par une fonction d'activation, comme dans un réseau neuronal artificiel classique. Ils passent ensuite à la couche de sortie, dans laquelle chaque neurone représente une étiquette de classification.

Dans un problème de classification, la couche FC renvoie un vecteur de taille  $N$ , tel que chaque élément du vecteur renvoie une probabilité pour l'image d'appartenir à une classe, avec  $N$  le nombre de classes dans notre problème.

Pour les réseaux CNN, la couche Fully-Connected détermine le lien entre la position des *features* dans l'image et une classe.

Cette couche est schématisée dans la Figure 2.10 :

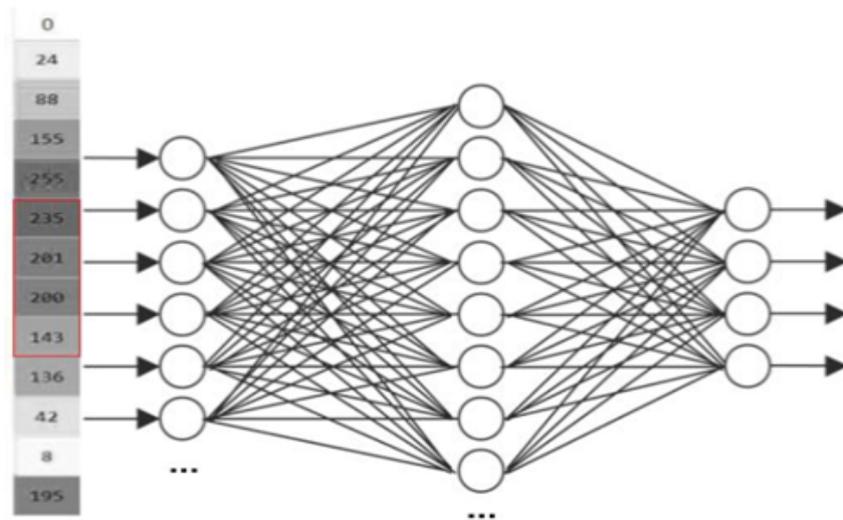


FIGURE 2.10 – Couche complètement connectée de 4 classes

### 2.3.2 Sous-apprentissage et Sur-apprentissage

Le but de l'entraînement d'un réseau de neurones est d'exploiter les données pour fixer un modèle avec les poids adéquats qui permettrait de prédire la sortie pour de nouvelles données non utilisées auparavant lors de la phase d'apprentissage. Cependant, la généralisation rencontre souvent deux problèmes : le sous-apprentissage (*Underfitting*) et le sur-apprentissage (*Overfitting*). La Figure 2.11 [14] démontre visuellement ces deux notions, en comparaison avec une bonne généralisation.

Le sous-apprentissage sous-entend que le modèle prédictif ne s'adapte pas convenablement avec le *Training set*, ce qui conduira à une mauvaise généralisation sur d'autres données. D'autre part, le sur-apprentissage désigne un apprentissage générant un modèle de prédiction qui s'adapte tellement bien au *Training set* qu'il prenne en compte des éléments secondaires (ex : le bruit), ce qui amènera le modèle à planter lors d'une prédiction sur de nouvelles données.

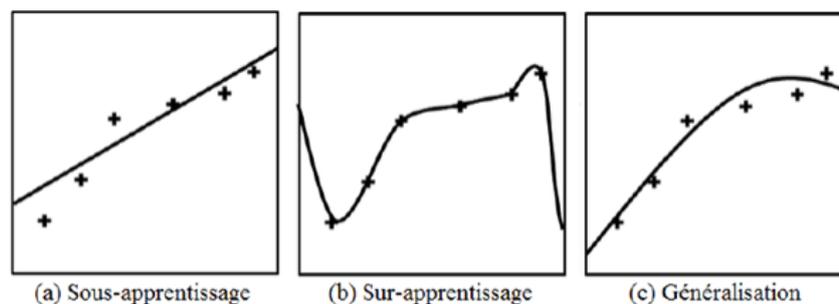


FIGURE 2.11 – Sous-apprentissage, sur-apprentissage et généralisation. (Sharma, Nori et al 2014)

#### 2.3.2.1 Compromis biais-variance

Un modèle simple (à variance simple), risque le sous-apprentissage compte tenu du biais élevé sur les données. D'autre part, un modèle présentant une variance élevée risque le sur-apprentissage, ceci s'explique par le biais faible sur les données d'entraînement mais élevé sur les nouvelles données.

Le concept du compromis biais-variance est représenté par la Figure 2.12 [34].

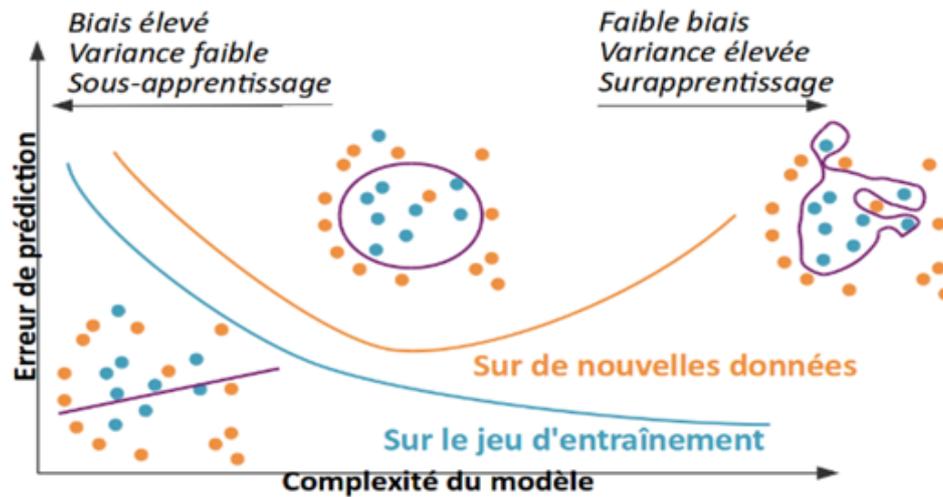


FIGURE 2.12 – Compromis biais-variance

### 2.3.2.2 Méthodes pour éviter le sur-apprentissage

- **Arrêt précoce** : Arrêt précoce ou « *Early Stopping* » en anglais est la technique qui vise à arrêter l'entraînement au point de moindre erreur par rapport au jeu de données de validation.

- **Dropout** : L'idée clé derrière le *dropout* est de déconnecter momentanément et au hasard des neurones et leurs connexions du réseau pendant l'entraînement (Figure 2.13 [35]). La désactivation de chaque neurone suit une loi de probabilité de Bernoulli de paramètre  $\rho$ . Par conséquent chaque neurone du réseau a une probabilité  $\rho$  d'être déconnecté, et ce indépendamment des autres neurones.

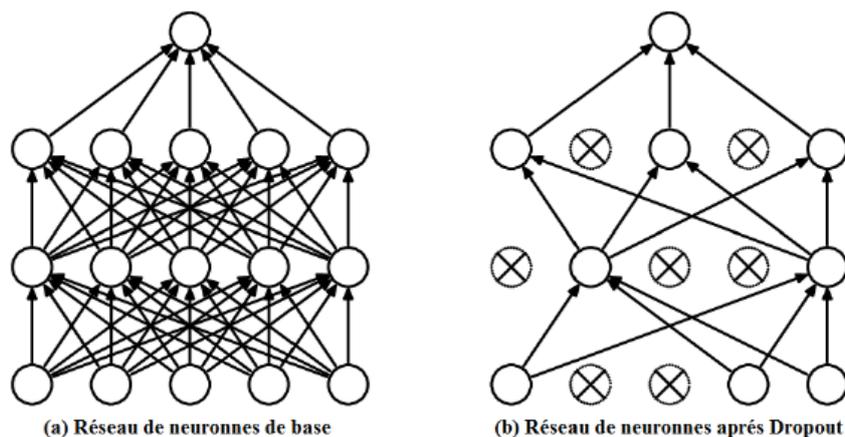


FIGURE 2.13 – Illustration du fonctionnement du Dropout (Srivastava, Hinton et al. 2014)

### 2.3.2.3 Validation croisée

Il s'agit de mélanger le jeu de données pour ajouter de l'aléatoire et éviter ainsi les possibles corrélations entre les données.

- Diviser le jeu de données en  $k$  groupes.
- Pour chaque groupe  $i$  ( $i \leq k$ ), exécuter un entraînement du réseau de neurones  $i$

en prenant ce groupe là comme jeu de données de test et les  $(k - 1)$  autres groupes comme jeu de données d'entraînement et calculer l'erreur de prédiction  $E_i$  de ce modèle.

- En procédant de la sorte, nous aurons à la fin  $k$  modèles entraînés séparément et nous obtiendrons l'erreur de prédiction de notre modèle indépendamment du jeu de données.

### 2.3.3 Optimisation de l'apprentissage par des algorithmes d'optimisation avancés

Les techniques avancées sont des adaptations de l'équation de la descente de gradient standard dans le but d'optimiser la tâche d'apprentissage.

Pour les modèles d'apprentissage contenant des millions de paramètres, la fonction de coût est optimisée sur des millions de dimensions. Les fonctions de coût pour ce type de modèles sont souvent très complexes et l'objectif reste de trouver le minimum global de la fonction.

Pour la descente de gradient, nous faisons un pas le long du gradient dans chaque dimension, lorsqu'on se retrouve au niveau d'un minimum local (Figure 2.14), l'algorithme se bloque et la position oscille car la taille du pas dans une direction donnée est déterminée par la valeur du gradient.

Les points selles (Figure 2.14) sont également un problème pour l'apprentissage, ils peuvent ralentir le processus ou l'arrêter dans le cas où la descente de gradient converge vers une valeur sous-optimale.

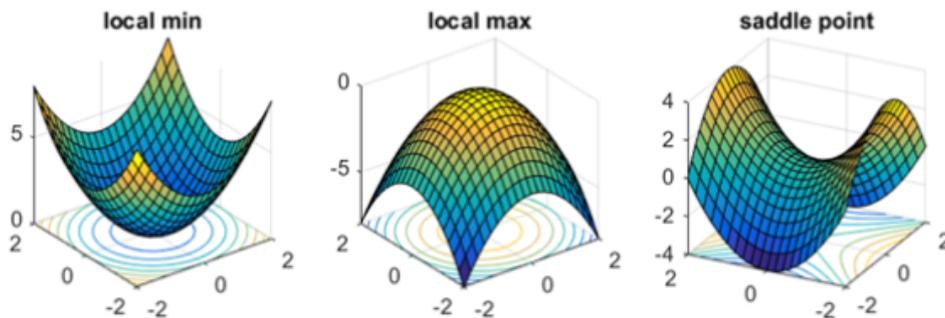


FIGURE 2.14 – Minimum local, maximum local, point selle (Photo par Rong Ge)

Des techniques avancées de calcul de gradient ont été développées pour réduire l'impact des problématiques citées auparavant et pour l'optimisation du temps d'apprentissage.

Le choix d'algorithme dépend du type de tâche et de données. Les optimisateurs avancés les plus utilisés spécialement pour les réseaux de neurones profonds sont les suivants :

#### — Momentum

L'approche du Momentum consiste à utiliser un gradient moyen au lieu du gradient immédiat à chaque pas de temps. En utilisant un gradient moyen sur  $n$  pas, l'élan est maintenu à l'approche de l'optima local. Comme on calcule un gradient moyen, le ralentissement n'est pas aussi grand comparé à la descente du gradient standard. Cependant, il est possible de rester bloqué sur un point selle.

L'avantage qu'offre l'optimisateur Momentum est la réduction des oscillations en se focalisant sur la voie de descente la plus raide et donc l'accélération de la recherche du minimum global.

#### — RMSProp

RMSProp est un schéma d'optimisation alternatif introduit par Geoff Hinton. Cette technique vise à empêcher les optima locaux de ralentir le processus de convergence en échelonnant de manière adaptative le taux d'apprentissage dans chaque dimension en fonction de la moyenne pondérée exponentielle du gradient. Cela a pour effet d'amortir les fortes pentes afin de réduire les oscillations dans les ravins pendant le chemin vers l'optimum global.

#### — ADAM

ADAM qui signifie « *Adaptive Moment Estimation* », combine les deux idées précédentes du Momentum (utilisation d'une moyenne mobile de gradient) et RMSProp (taux d'apprentissage adaptatif). L'estimation du moment adaptatif (ADAM) consiste à calculer le taux d'apprentissage adaptatif pour chaque paramètre, et stocker une moyenne à décroissance exponentielle des gradients carrés passés. « Si on considère l'élan comme une balle descendant une pente, Adam se comporte comme une balle lourde avec frottement, qui préfère donc des minimas plats dans la surface d'erreur » [36].

### 2.3.4 Apprentissage par Transfert

En pratique, entraîner un CNN peut s'avérer très coûteux, en terme de temps et de puissance de calcul. En effet, la majorité des problématiques nécessitent une très grande quantité de données d'entraînement et une grande puissance de calcul. C'est pour cela que la notion de l'apprentissage par transfert (*Transfer Learning* en Anglais), est très utilisée pour faire face à ces deux inconvénients.

Le principe de l'apprentissage par transfert [37] est d'utiliser des connaissances apprises par un réseau de neurones, les valeurs de poids et de biais dans ce cas, lors de la résolution d'une première tâche avec un réseau formé A, pour la résolution d'une autre tâche, plus ou moins similaire, en transférant les caractéristiques apprises par le premier réseau vers un réseau B, le jeu de données pouvant être différent de celui de la source. C'est donc une technique qui utilise un modèle pré-entraîné sur d'autres données afin de nous éviter de construire un nouveau modèle et de faire l'entraînement à partir de zéro. Ce processus fonctionne dans les cas où l'apprentissage du réseau A est bien généralisé.

#### 2.3.4.1 Catégories de l'apprentissage par transfert

Les catégories d'apprentissage par transfert peuvent être classées en fonction du type d'algorithmes d'apprentissage traditionnels concernés [38] :

- **L'apprentissage par transfert inductif** : Dans ce scénario, les domaines source et cible sont les mêmes, mais les tâches source et cible sont différentes l'une de l'autre. Les algorithmes tentent d'utiliser les biais inductifs du domaine source pour améliorer la tâche cible. Selon que le domaine source contient des données étiquetées ou non, il peut être divisé en deux sous-catégories, similaires respectivement à l'apprentissage multitâche et à l'apprentissage autodidacte.
- **Apprentissage par transfert non supervisé** : Ce paramètre est similaire au transfert inductif lui-même, avec un accent sur les tâches non supervisées dans le

domaine cible. Les domaines source et cible sont similaires, mais les tâches sont différentes. Dans ce scénario, les données étiquetées ne sont disponibles dans aucun des domaines.

- **Apprentissage par transfert transductif** : Dans ce scénario, il y a des similitudes entre les tâches sources et cibles, mais les domaines correspondants sont différents. Dans ce cas, le domaine source contient beaucoup de données étiquetées, tandis que le domaine cible n'en contient aucune. Il est possible de classer ces données en sous-catégories, en se référant aux paramètres où les espaces de caractéristiques sont différents ou aux probabilités marginales.

#### 2.3.4.2 Stratégies d'apprentissage par transfert

Il est nécessaire de choisir avec soin la partie du modèle pré-entraîné à utiliser dans le nouveau modèle. C'est là qu'apparaît la notion de « gel ». Geler une couche reviendrait à fixer ses paramètres à ceux qui ont été appris du modèle pré-entraîné sans changement possible. Il reste tout de même possible de permettre aux couches un nouveau processus d'apprentissage ayant pour poids initiaux ceux de l'ancien modèle.

De ce fait, un compromis doit être établi pour savoir quelles sont les couches qui seront gelées et quelles sont celles qui ne le seront pas. Ce qui donne naissance à différents types d'apprentissage par transfert.

Les stratégies d'apprentissage par transfert à adopter est liée aux différents contextes décrits précédemment. On peut résumer ces stratégies comme suit :

##### - **Extracteur de caractéristiques** :

Cette méthode est utilisée dans le cas où l'ensemble des données cible (Target dataset) est petit et similaire à la base de données d'origine. Vu le nombre restreint de données, il n'est pas intéressant d'entraîner les couches du CNN afin d'éviter le 'sur-apprentissage'. Cependant, étant donné que les données soient similaires à ceux d'origine, les caractéristiques tirées des dernières couches du réseau devraient être pertinentes pour cet ensemble de données. Pour cela, il serait meilleur de supprimer la dernière couche entièrement connectée, puis geler le reste des paramètres du réseau et le considérer comme un extracteur de caractéristiques pour la nouvelle base de données.

##### - **Réglage complet** :

Cette approche est considérée dans le cas où le nouveau dataset est volumineux et similaires à celui du modèle pré-entraîné. Elle consiste d'une part à remplacer la couche entièrement connectée du réseau pré-entraîné en fonction des besoins du nouveau modèle, et de l'autre à entraîner les poids des précédentes couches en continuant la rétropropagation. Ceci est faisable compte tenu du nombre important de données disponibles, effaçant ainsi tout risque de sur-apprentissage. Aussi, vu que les paramètres de toutes les couches sont initialement ceux du réseau pré-entraîné, la phase d'apprentissage sera faite plus rapidement que pour une initialisation aléatoire.

### - Réglage partiel :

Cette méthode est appliquée dans le cas où le dataset est restreint et très différent de celui d'origine. Le nombre de données étant petit, il n'est préférable de réentraîner que la couche entièrement connectée afin d'éviter un sur-apprentissage. Cependant, les caractéristiques extraites des couches du réseau convolutif ne seraient pas représentatives des nouvelles données vu leur différence avec ceux d'origine. Ainsi, la solution serait de geler les paramètres des premières couches (qui sont plus générales) et d'entraîner les dernières couches du réseau convolutif (qui sont plus spécifiques) ainsi que la nouvelle couche entièrement connectée.

### - Entraînement à zéro :

Cette méthode est utilisée dans le cas où l'ensemble de données est volumineux et très différent de l'ensemble de données original. Vu le nombre important et suffisant de données dont nous disposons, un entraînement du réseau de neurone peut être fait à partir de zéro. Néanmoins, en pratique, l'initialisation avec les poids d'un modèle pré-entraîné est toujours plus utile.

## 2.3.5 Architectures des CNNs les plus utilisées

### 2.3.5.1 AlexNet

AlexNet est une architecture de CNNs profonds développée par Alex Krizhevsky en 2012. Entraînée sur ImageNet, c'est une maquette gagnante de son concours de reconnaissance visuelle à grande échelle 2012 (ILSVRC-2012).

Ce modèle est composé de 5 couches de convolution où la première, la seconde et la cinquième sont suivies par des couches de pooling, et trois couches fully-connected sont ajoutées en dernier comme le montre la Figure 2.15.

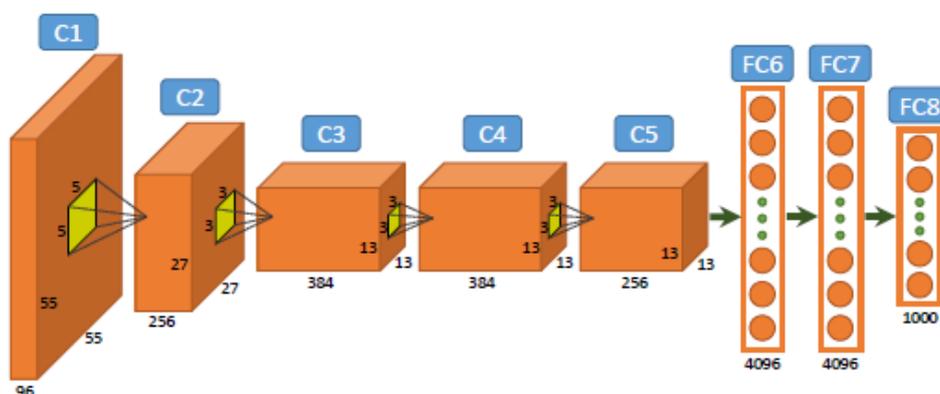


FIGURE 2.15 – Architecture AlexNet

Les particularités de l'architecture AlexNet en comparaison avec les anciennes architectures se résument dans quelques astuces pratiques, à savoir, les unités linéaires rectifiées (ReLU), l'augmentation des données, la non-linéarité, et la technique du dropout. Ces particularités ont permis au réseau AlexNet de devenir une architecture de base des CNNs modernes.



Le VGG-16 accepte en entrée une image RGB de taille  $224 \times 224$ , l'entrée est ensuite passée dans un ensemble de couches convolutives qui représentent les 13 principales couches du réseau. Les filtres de convolution sont de taille  $3 \times 3$  avec un *padding* de 1 pixel. La taille des fenêtres du *maxpooling* est de  $2 \times 2$  avec un *stride* de 2. Après les 13 premières couches, on obtient une carte de caractéristiques de l'image de taille  $7 \times 7 \times 512$ . Après l'opération de l'aplatissement dite « *Flattening* » on obtient un vecteur de caractéristiques. Ce vecteur s'exposera à 3 couches FC, la dernière ayant 1000 neurones de sorties représentant le nombre de classes de la base de données ImageNet.

La Figure 2.17 présente l'architecture générale du réseau VGG-16.

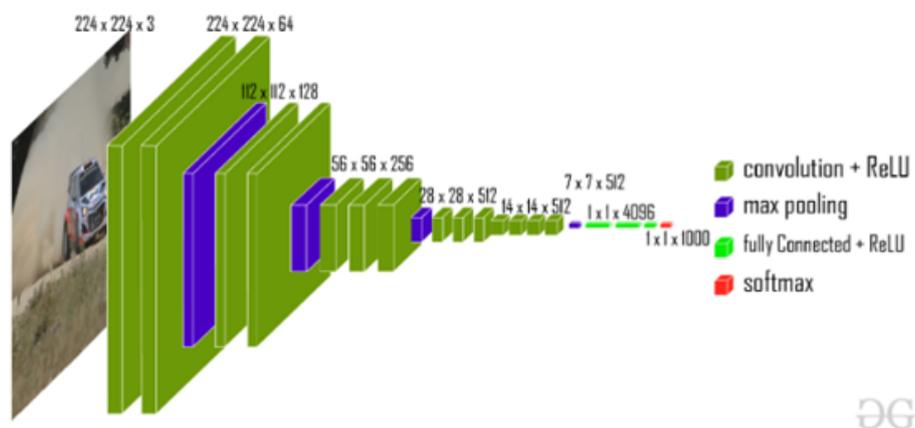


FIGURE 2.17 – Architecture du VGG-16

## 2.4 Conclusion

Dans ce chapitre nous avons commencé par expliquer le fonctionnement des réseaux de neurones artificiels tout en résumant l'historique de leur apparition. Nous avons introduit les principales notions gravitant autour des CNNs, et nous avons exposé l'architecture type d'un réseau de neurone convolutif, ainsi que le rôle de chaque couche et son fonctionnement.

Par la suite, nous nous sommes intéressées à l'apprentissage par transfert et ses principales stratégies, largement utilisées pour l'entraînement d'un CNN. Cette section explique le contexte d'utilisation de chacune de ces techniques et leur utilité.

Le chapitre est conclu par la présentation des modèles de CNNs les plus utilisés et les plus adaptés à la DC, ce qui facilitera la compréhension du chapitre suivant.

## Chapitre 3

# Réseaux de Neurones Convolutifs Adaptés pour la Détection de Changement

## 3.1 Introduction

Ce chapitre est consacré à la présentation de deux approches suivies pour l'élaboration de solutions au problème de la DC en télédétection en utilisant des FCN.

La première partie est un bref aperçu sur les FCN. Cette section permet de comprendre les modèles adoptés dans notre travail et l'intérêt qu'apporte cette méthode pour les deux solutions décrites dans ce chapitre. La deuxième partie du chapitre expose le modèle d'un FCN basé sur un VGG-16 pré-entraîné sur la base de données ImageNet. Cette méthode consiste à utiliser le réseau comme extracteur de caractéristiques et ces dernières sont utilisées pour mesurer la similarité entre les images bi-temporelles de type RGB en exploitant les Hypercolonnes construites à partir des couches intermédiaires du réseau.

La troisième partie du chapitre décrit une méthode basée sur l'entraînement end-to-end d'un réseau siamois entièrement convolutif utilisant les « long skip-connections » afin d'exploiter les résultats de la distance euclidienne et la différence d'image de certaines couches du réseau.

Pour l'élaboration de ces méthodes, l'indisponibilité de grandes bases de données étiquetées et la limitation d'accès aux ressources ont été prises en compte.

## 3.2 Les réseaux de neurones entièrement convolutifs

Un réseau entièrement convolutif « FCN » (en anglais : *Fully Convolutional Neural Network*) [40] utilise un réseau neuronal convolutif pour transformer les pixels d'une image en catégories de pixels. Contrairement aux réseaux neuronaux convolutifs, un FCN transforme la hauteur et la largeur de la carte des caractéristiques de la couche intermédiaire pour revenir à la taille de l'image d'entrée à travers la couche de convolution transposée « *Conv2DTranspose* » [41], de sorte que les prédictions ont une correspondance bi-univoque avec l'image d'entrée dans la dimension spatiale. Pour une position donnée sur la dimension spatiale (hauteur et largeur), la sortie est une prédiction de catégorie du pixel correspondant à l'emplacement.

Pour la première approche adoptée, le modèle produit des sorties spatiales mais la prédiction n'entre pas en jeu, car le réseau FCN est uniquement utilisé pour l'extraction de caractéristiques, la mesure de similarité étant une étape ultérieure.

Les CNNs classiques nécessitent des entrées de taille fixe. Pour permettre au modèle d'accepter des entrées avec des tailles variantes, les couches entièrement connectées sont supprimées de l'architecture. Cette démarche permet en plus pour la première approche d'optimiser le temps d'exécution et l'espace mémoire. En effet, pour les CNNs profonds, les couches entièrement connectées contiennent plus de 70% des poids du réseau [40].

Pour la deuxième approche, une architecture siamoise particulière entièrement convolutive est adoptée (section 3.4.2). La technique entièrement convolutive est utilisée pour obtenir des sorties spatiales de même taille que les images d'entrées. La prédiction se fait en classifiant chaque pixel dans une catégorie : 'changé' ou 'inchangé'.

## 3.3 L'extraction de caractéristiques pour la détection de changement

L'extraction des caractéristiques des CNNs profonds est une stratégie de transfert d'apprentissage exploitée pour la résolution de plusieurs problématiques dans le domaine de l'IA. Il existe plusieurs solutions pour la problématique de la DC en télédétection basées sur cette technique (section 1.4.3.1). Le modèle proposé dans cette section est en lien avec le travail de Larabi et al [3] qui fut le premier à avoir utilisé des CNNs pour la DC en imagerie satellitaire. L'approche introduit une modification d'architecture du réseau VGG-16 et est appliquée sans avoir recours à un entraînement. Cette section permet de comprendre l'approche et l'objectif de l'utilisation des caractéristiques de plusieurs couches du réseau, et expose les étapes du processus suivi pour la génération de la carte de changement.

### 3.3.1 La compréhension des caractéristiques des CNNs

En vision par ordinateur, le terme caractéristiques ou *features* d'une image désigne les zones d'intérêts de cette dernière. Ça peut correspondre à des bords, des coins ou des régions d'intérêts de toutes sortes (en anglais ROI : *Region Of Interest*). Les CNNs ont révolutionné le domaine de la vision par ordinateur, notamment la reconnaissance d'objets, grâce à leur capacité à entraîner le réseau de telle manière à ce qu'il soit apte à détecter les régions d'intérêts nécessaires à l'accomplissement d'une tâche particulière.

Les nombreuses architectures de CNNs existantes présentent un phénomène en commun : les premières caractéristiques sortantes des premières couches ressemblent aux caractéristiques des filtres de Gabor et aux blobs de couleurs, ce qui rend le gel des premières couches pour l'apprentissage par transfert très efficace pour une multitude de tâches et de données différentes. D'autre part, nous savons que les dernières couches d'un CNN sont au contraire en très forte corrélation avec les données et la tâche à réaliser [42] . Pour résumer, on peut distinguer trois différents types de caractéristiques : haut, moyen et bas niveau.

Pour une tâche comme la DC, l'exploitation des caractéristiques des couches intermédiaires est la plus adaptée [6] , compte tenu de l'information non spécifique de ces dernières mais qui définit plus clairement les zones d'intérêt comparé aux premières couches.

La Figure 3.1 [42] représente la visualisation de caractéristiques de 9 activations à partir de la 2<sup>ème</sup> à la 5<sup>ème</sup> couche de l'architecture AlexNet.



FIGURE 3.1 – Vizualisation des caractéristiques dans un réseau AlexNet

### 3.3.2 Architecture basée sur l'extraction de features

Cette section va d'abord décrire le modèle adopté pour la tâche de la DC, puis elle exposera les étapes détaillées de la méthode proposée.

Le modèle adopté pour la tâche de la DC est un VGG-16 (section 2.3.5.3) modifié. Les couches FC sont supprimées, ce qui rend le réseau entièrement convolutif. Cette modification permet d'extraire les caractéristiques nécessaires en introduisant des images de tailles différentes sans les redimensionner pour les adapter à une taille d'entrée unique. Comme le montre la Figure 3.3, l'architecture du réseau est différente par rapport au CNN classique qui contient dans sa première partie des couches de maxpooling. Ces couches seront exclues du modèle adopté, car la méthode de mesure de similarité est basée sur le pixel et la réduction de la dimensionnalité des cartes avec l'opération de Pooling diminuera la précision des caractéristiques extraites.

Les étapes du système de DC non supervisé basée sur le transfert de caractéristiques se présentent ainsi :

#### 3.3.2.1 La construction des Hypercolonnes

Pour un pixel donné, la définition d'une hypercolonne est le vecteur d'activation de toutes les unités CNN au-dessus de ce pixel [43], comme indiqué dans la Figure 3.2 [43].

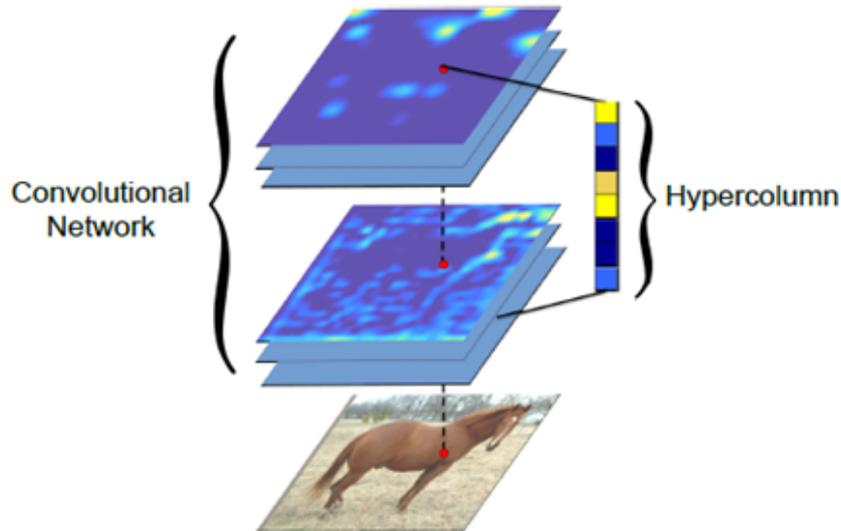


FIGURE 3.2 – Représentation des hypercolonnes

Pour cette approche, qu'on nommera "*FCN-VGG-16 adapté comme extracteur de caractéristiques*", la solution pour la DC se base sur l'extraction des hypercolonnes des deux images bi-temporelles, afin de les utiliser pour prédire le changement entre les deux images et ce, sans entraîner le réseau davantage, et sans prétraitement des images.

La construction des hypercolonnes se fait par l'extraction des cartes de caractéristiques appelées aussi « *feature maps* » de plusieurs couches intermédiaires et intermédiaires supérieures de notre FCN. Etant donnée que les opérations de maxpooling sont omises et que l'on emploie une opération de *ZeroPadding* avant chaque convolution, le problème de taille des cartes de caractéristiques ne se pose pas et la concaténation de ces cartes peut se faire afin de former l'hypercolonne.

### 3.3.2.2 La génération de la carte de changement :

La carte de changement est une image offrant une visibilité sur des changements produits dans une même zone en se basant sur deux images bi-temporelles. Dans notre cas, les pixels correspondants à un changement sont représentés en blanc, les pixels inchangés en noir.

Pour cette approche, la carte de changement est calculée en passant par la distance euclidienne définie par la formule (3.1) :

$$d_{ij} = \sum_{k=1}^k (\mu_i^k - \mu_j^k)^2 \quad (3.1)$$

Avec  $k$  la dimension de la *feature map*,  $\mu_i^k$  et  $\mu_j^k$  les valeurs des *features* pour la  $k^{eme}$  dimension pour les positions  $(i, j)$ . Pour obtenir la carte de changement finale une binarisation avec la méthode de seuillage OTSU est appliqué à  $d$ .

### 3.3.2.3 La méthode de seuillage OTSU

Cette méthode développée par N. OTSU [44] est une méthode de seuillage dont l'objectif est la binarisation d'images à niveaux de gris. L'algorithme de la méthode OTSU nécessite le calcul de l'histogramme de l'image au préalable. Les probabilités de chaque niveau de gris sont donc calculées à partir de cette information.

OTSU classe les pixels de l'image en deux classes : la région d'intérêt et l'arrière-plan. La probabilité d'appartenir à la classe 1 ou 2 selon un seuil  $T$  est calculée selon (3.2) et (3.3) respectivement :

$$\omega_1(T) = \sum_{k=1}^T P(k) \tag{3.2}$$

$$\omega_2(T) = \sum_{k=T+1}^{256} P(k) \tag{3.3}$$

Tel que  $P(k)$  est la probabilité d'appartenir au niveau de gris  $k$ .

La méthode OTSU est une méthode itérative qui vise à prédire un seuil  $T$  optimal pour la binarisation de l'image, elle est utilisée à la fin du système pour l'obtention d'une carte de changement plus précise et homogène.

Le schéma présenté dans la Figure 3.3 donne un aperçu général de l'architecture et les étapes essentielles de l'approche non-supervisée proposée.

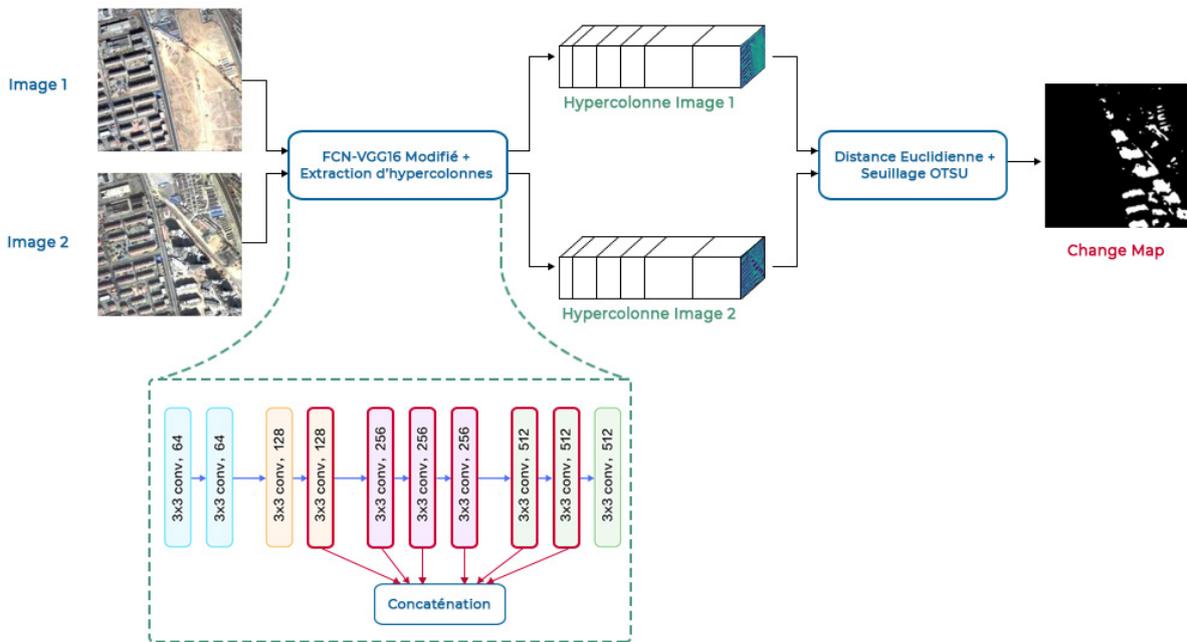


FIGURE 3.3 – Workflow de la première méthode proposée : 'FCN-VGG-16 adapté comme extracteur de caractéristiques'

### 3.4 Réseaux siamois entièrement convolutifs pour la détection de changement

Contrairement à la méthode précédente, où il est question d'utiliser un CNN pré-entraîné pour extraire les *features* d'une paire d'images et seuiller la différence entre les deux hypercolonnes formées pour en obtenir la carte de changement, cette deuxième méthode, qu'on nommera "*FCN-siamois ED&DI sans maxpooling*", est un réseau siamois modifié entièrement convolutif entraîné de façon end-to-end et utilisant les « *skip-connections* » afin d'obtenir en sortie la carte de changement où chaque pixel correspond à l'une des deux classes : "changé" ou "inchangé"

### 3.4.1 Les réseaux siamois

L'architecture d'un réseau siamois, également appelés réseaux jumeaux (en anglais *Siamese Network*) est présentée comme deux réseaux d'encodage différents qui partagent des poids, le but étant d'apprendre comment encoder des images pour quantifier le degré de différence de deux entrées données. La Figure 3.4 montre l'architecture générique d'un réseau siamois. En réalité, l'architecture siamoise est un réseau utilisé deux fois avant de faire la rétropropagation [45].

Les premiers travaux ayant recours à une architecture siamoise sont les travaux de Bromley et al [46] pour la vérification de signatures. Ce type d'architecture a la particularité d'apprendre une mesure de similarité entre deux entrées indépendantes partageant une relation de similarité. La méthode utilisée est la comparaison des caractéristiques de la sortie des deux réseaux, la comparaison en question peut se faire avec plusieurs fonctions de perte, à savoir la perte de triplet, la perte contrastive et le pseudo étiquetage avec perte d'entropie croisée [45].

Les architectures siamoises sont très souvent adoptées pour des problématiques de comparaison d'images en utilisant les CNNs. On cite le travail sur la comparaison d'images par patch de S. Zagoruyko [40] . Pour la problématique de la DC, les architectures siamoises ont également été introduites récemment, à l'exemple des travaux de R. Daudt [47], pour la résolution de cette tâche. Les réseaux siamois sont utilisés dans d'autres domaines comme la segmentation, l'identification faciale et le suivi des objets.

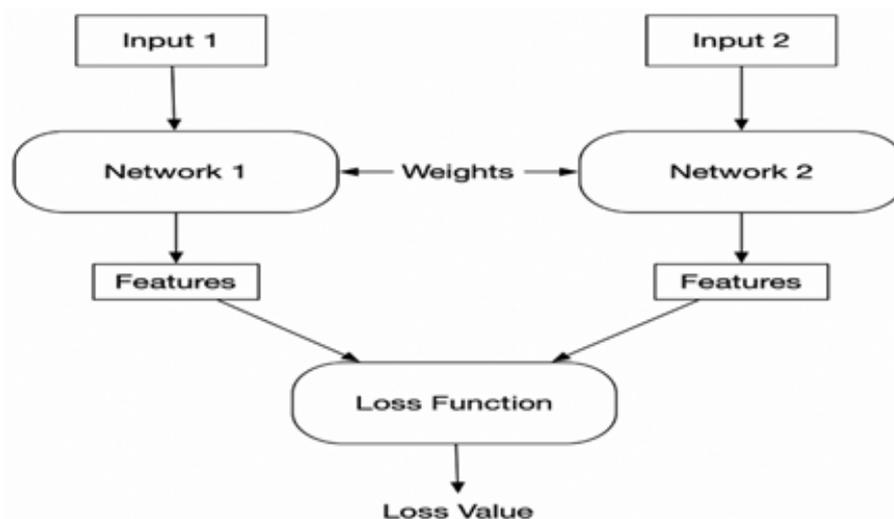


FIGURE 3.4 – Architecture classique d'un réseau siamois

### 3.4.2 Architecture du modèle adopté

Pour cette approche "*FCN-siamois ED&DI sans maxpooling*", nous avons adopté un modèle basée sur l'architecture siamoise, pouvant ainsi accepter deux images en entrée pour un entraînement end-to-end. Dans ce qui suit, nous allons introduire la notion de '*Skip Connections*' et détailler l'architecture du modèle par la suite.

### 3.4.2.1 Skip Connections

Les skip-connections sont des liaisons entre des couches de la même échelle de sous-échantillonnage dans un modèle encodeur-décodeur. L'architecture du schéma d'encodage et de décodage, ainsi que les « *long skip connections* », est souvent appelée forme en U (Figure 3.5). Elle est utilisée pour des tâches dont la prédiction a la même dimension spatiale que l'entrée, telles que la segmentation de l'image, la prédiction vidéo, ou comme dans notre cas la DC. Les *skip-connections* sont utilisées pour faire passer les caractéristiques du chemin du codeur au chemin du décodeur afin de récupérer les informations spatiales perdues lors du sous-échantillonnage [48]. Ces connexions permettent donc d'ajouter des détails spatiaux présents dans les premières couches du réseau convolutif aux couches supérieures, plus abstraites.

Pour notre modèle, nous adaptons les *skip-connections* à la DC en modifiant la façon dont elles sont utilisées : au lieu pour concaténer les deux connexions prises des deux flux du codeur, nous concaténons la valeur absolue de leur différence ainsi que leur distance euclidienne. De cette manière, nous obtenons, en plus des détails spatiaux, une meilleure représentation de la différence entre les deux images en sortie.

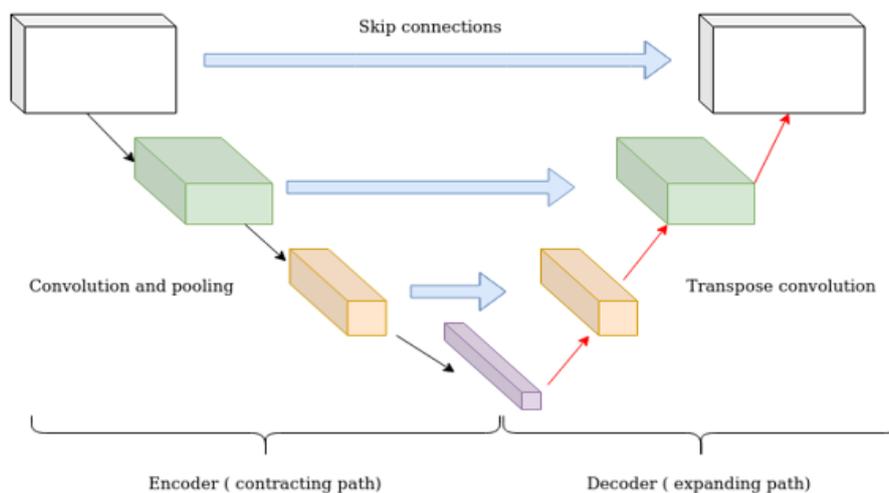


FIGURE 3.5 – Schéma d'un encodeur-décodeur en architecture U

### 3.4.2.2 Description de l'architecture

L'architecture proposée de la méthode "*FCN-siamois ED&DI sans maxpooling*" est inspirée des modèles codeurs-décodeurs existants, elle est cependant modifiée afin d'être adaptée à notre problématique.

Le modèle est composé de deux blocs principaux : le premier bloc est sous forme d'une architecture siamoise, étant séparé en deux flux de structures identiques et de poids partagés, il accepte deux images en entrée. Chaque image est donc traitée en parallèle par 6 couches de convolutions (Figure 3.6) sans avoir recours aux opérations de maxpooling qui réduisent la précision des caractéristiques spatiales extraites dans les réseaux conventionnels.

Les *feature maps* de 3 blocs de convolutions seront utilisées ultérieurement dans les *skip-connections* qui, différentes de celles qu'on retrouve habituellement dans les modèles

de type Unet, seront exploitées pour la concaténation de la distance euclidienne et de la différence d'images entre lesdites *feature maps* (section 3.4.2.1).

Par la suite, la sortie de l'image 1 passe par deux autres couches convolutives avant d'entrer dans le deuxième bloc du réseau qui, symétrique au premier, est composé également de 6 couches convolutives où chaque deux niveaux sont concaténés avec les deux distances euclidiennes et différences absolues issues à partir du premier bloc, à l'aide de la *skip-connection* correspondante.

L'architecture est conclue par une couche convolutive 1 x 1 et la fonction d'activation 'sigmoïde' donnant ainsi en sortie la carte de changement binaire correspondante à la paire d'images en entrée, où chaque pixel de l'image sortante est classé en « changé » et « inchangé ».

La Figure 3.6 présente un schéma descriptif de l'architecture du modèle siamois entièrement convolutif modifié adopté dans notre travail.

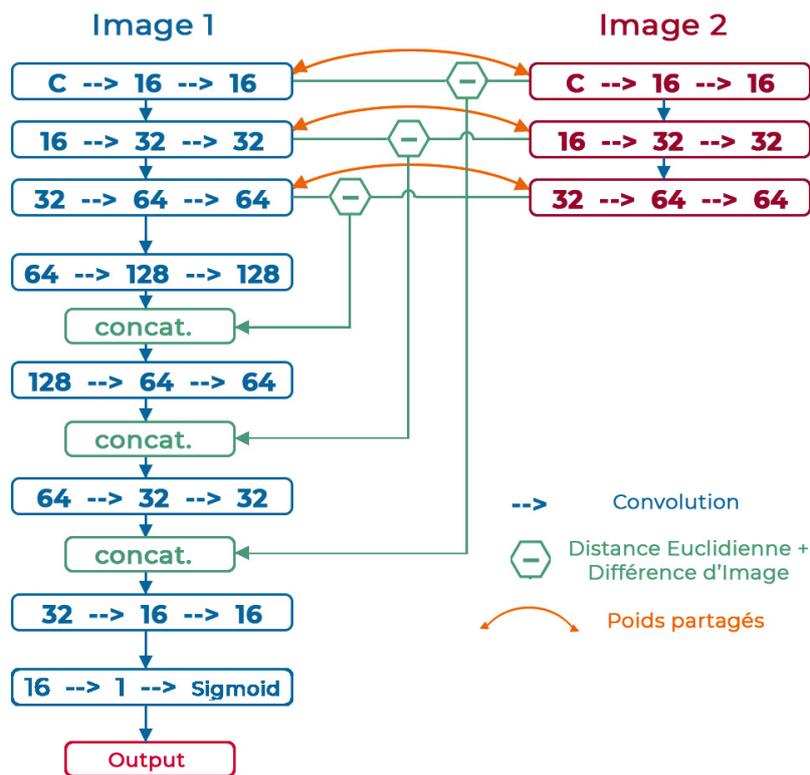


FIGURE 3.6 – Workflow de la deuxième méthode proposée : 'FCN-siamois ED&DI sans maxpooling'

## 3.5 Conclusion

Dans ce chapitre, nous avons introduit les FCNs en expliquant l'intérêt de leur utilisation dans nos approches.

Nous avons également montré l'importance de l'exploitation des caractéristiques extraites des différentes couches du réseau dans la résolution de certaines problématiques, notamment la DC.

Par la suite, nous avons exposé une première architecture basée sur l'extraction de *features* et détaillé ses différentes étapes en passant par les notions d'hypercolonnes et de seuillage.

Nous concluons le chapitre en présentant une deuxième approche basée sur les réseaux siamois entièrement convolutifs, et où les skip connections ont été réadaptées de sorte à faciliter davantage la DC.

# Chapitre 4

## Implémentation des Méthodes et Résultats

## 4.1 Introduction

Ce dernier chapitre est dédié à l'implémentation et à l'analyse des performances des systèmes de DC proposés dans ce travail. Dans un premier temps, il est question de présenter la base de données utilisée pour l'entraînement et les tests effectués pour l'évaluation, ainsi que les différents outils informatiques et bibliothèques employés.

Par la suite, nous exposerons les résultats des tests effectués pour les deux méthodes afin de les interpréter séparément et de comparer les différentes variantes de chaque méthode.

Ce chapitre sera conclu par la comparaison des deux principales approches entre elles et la présentation des avantages et inconvénients de chacune.

## 4.2 Base de données

Malgré les nombreux programmes d'observation de la terre et l'abondance d'images satellitaires brutes, il y a peu de données annotées en libre accès nécessaires au développement de méthodes d'apprentissage supervisées [47].

Pour remédier à cela, nous avons créé notre propre base de données en utilisant les images disponibles de la Dataset ONERA Satellite Change Detection<sup>1</sup>. Elle contient un total de 14 paires d'images bi-temporelles prises par les satellites Sentinelles-2, de tailles différentes et accompagnées de leurs cartes de changement manuellement étiquetées. Elle comporte des images de zones urbaines dont la résolution varie de 10m à 60m et où seuls les changements urbains sont identifiés (les changements naturels ne sont pas pris en compte lors de l'étiquetage).

S'ajoute à cela 4 paires d'images de SZTAKI Air Change Dataset, représentant une zone agricole, ainsi qu'une autre paire d'images de la ville de Beijing prise par le satellite Quick-Bird [6].

Etant donnée que les architectures proposées ne contiennent pas de couches *Fully Connected*, la taille des images d'entraînement et de test peuvent être différentes. En effet, comme illustré dans la Figure 4.1, pour créer les paires d'images de test (*Testing\_Set*), nous découpons le côté droit du bas de quelques images de sorte qu'il ait la moitié de la taille d'origine. Le reste de l'image est découpé en petites images chevauchées de 128x128 pixels chacune, formant ainsi les données d'entraînement (*Training\_Set*).



FIGURE 4.1 – Illustration des étapes suivies pour la création de la base de données

1. <https://iee-dataport.org/open-access/oscd-onera-satellite-change-detection>

Nous obtenons au final une base de données contenant quelques paires d'images de test de tailles différentes et 5118 paires d'images d'entraînement de taille 128x128 pixels. Ces dernières seront augmentées par la suite lors de l'entraînement.

Dans la suite de ce document, nous nous appuyerons sur les 3 paires d'images test illustrées dans la Figure 4.2, pour exposer les résultats de nos méthodes. Elles seront dénotées par "*Dataset1*", "*Dataset2*" et "*Dataset3*" de tailles (424x424), (392x504) et (424x312) pixels respectivement.

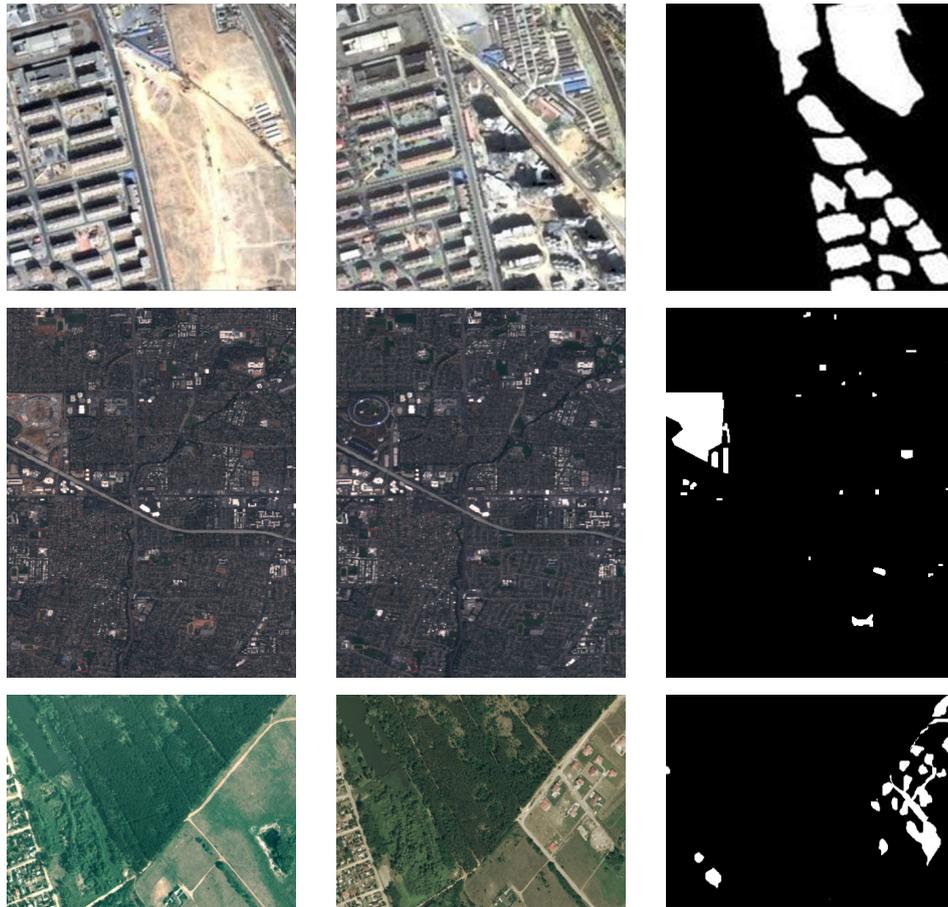


FIGURE 4.2 – Représentation des paires d'images de '*Dataset1*', '*Dataset2*' et '*Dataset3*' respectivement, aux temps T1 et T2 dans les deux premières colonnes, ainsi que leur GT correspondant dans la 3ème colonne.

## 4.3 Bibliothèques et Outils

Dans ce qui suit, nous présentons les divers outils et bibliothèques utilisés pour mettre en oeuvre les étapes décrites dans le chapitre précédent, et implémenter nos solutions.

### 4.3.1 Python

Python est un langage de programmation orienté objet, de haut niveau et open source d'utilisation générale. Il est conçu pour être utilisé dans de multiples plateformes, pour des programmes simples ou complexes, et est doté de plusieurs bibliothèques utiles à la programmation dans différents contextes et applications, notamment dans l'apprentissage machine et l'Intelligence Artificielle.



Python utilise l'indentation pour délimiter les blocs d'un programme au lieu des accolades et autres délimiteurs, ce qui fait de lui un langage lisible et concis.

Nous avons utilisé Python version 3.6 pour l'implémentation de la totalité de nos méthodes.

### 4.3.2 TensorFlow

Tensorflow est une bibliothèque de logiciels open source conçue pour l'apprentissage automatique et les réseaux de neurones profonds. Elle est développée par *Google Brain Team* en 2011 et s'appelait à l'origine *DistBelief*. Google la modifie et la renomme Tensorflow en 2015, l'année où elle est rendue ouverte au public.

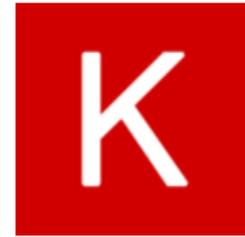
Tensorflow regroupe une multitude d'algorithmes qui permettent de développer et d'exécuter des applications de *ML* et de *DL*. Ses outils offrent la possibilité de résoudre des problèmes mathématiques extrêmement complexes et permet donc au développeur de se focaliser sur la logique générale de l'application.



Nous avons utilisé la version 2.3.0 de Tensorflow dans la conception des réseaux de neurones profonds.

### 4.3.3 Keras

Keras est une bibliothèque Open Source de haut niveau, écrite en Python (pour le développement en Python.) et conçue pour la conception, la configuration et l'entraînement des modèles de réseaux de neurones. Elle permet le prototypage facile et supporte à la fois les CNNs et récurrents. Keras permet d'interagir avec d'autres bibliothèques de plus bas niveau comme *TensorFlow* ou *Theano*.



Nous avons utilisé Keras version 2.4.3 pour faciliter la conception de nos réseaux de neurones profonds.

### 4.3.4 OpenCV

OpenCV (*Open Source Computer Vision Library*) est une bibliothèque graphique Open Source initiée par Intel en 2000. Elle regroupe des algorithmes spécialisés dans le traitement d'images en temps réel, mais également l'apprentissage automatique.



OpenCV permet la création d'applications simples à partir de données brutes jusqu'au développement d'interfaces graphiques. Elle est disponible pour plusieurs langages de programmation dont Python, Java et C++.

Nous avons utilisé OpenCV version 4.1.2 pour la préparation, le traitement et la visualisation de nos données.

### 4.3.5 Google Colaboratory

Google Colaboratory, souvent réduit en Google Colab, est un service gratuit de *Google Research* qui permet l'écriture et l'exécution de codes en Python sur le navigateur sans configuration requise. Il est basé sur *Jupyter Notebook* et est destiné spécialement à la recherche dans le domaine de l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles d'apprentissage automatique et profond directement sur Cloud sans avoir recours à l'installation d'autres outils que le navigateur. Les ressources de Google Colab en RAM et en puissance de calcul restent cependant limitées.



Nous avons utilisé Google Colab pour l'écriture et l'exécution de nos programmes.

## 4.4 Critères d'évaluation

Afin d'évaluer les performances des architectures proposées, nous faisons appel à certains critères standards adaptés aux algorithmes liés à la DC et qui sont la *CA*, le coefficient *Kappa* et la *Précision*.

### 4.4.1 Classification Accuracy

L'accuracy de classification est défini comme le rapport entre le nombre de prédictions correctes et le nombre total de prédictions Elle est formulée selon l'équation (4.1) :

$$CA = \frac{VP + VN}{VP + VN + FP + FN} \quad (4.1)$$

où :

**VP** : Vrai positif, est une mesure représentant le nombre de pixels correctement classifiés comme "changés".

**VN** : vrai négatif, est une mesure représentant le nombre de pixels correctement classifiés comme "inchangés".

**FP** : Faux positif, est le nombre de pixels "inchangés" dans le GT mais qui sont classifiés comme "changés".

**FN** : Faux négatif, est le nombre de pixels "changés" dans le GT mais qui sont classifiés comme "inchangés".

### 4.4.2 Coefficient Kappa

Le coefficient Kappa est un indice statistique variant entre 0 et 1 qui peut s'interpréter comme une proportion d'accord, *ie* la proportion d'éléments classés dans la même catégorie par deux entités, le GT et la prédiction dans notre cas.

Il se calcule suivant la formule (4.2) :

$$Kappa = \frac{CA - PRE}{1 - PRE} \quad (4.2)$$

avec :

**CA** : Classification Accuracy

**PRE** : correspond à la probabilité d'un accord aléatoire formulé selon (4.3) :

$$PRE = \frac{(VP + FP).(VP + FN)}{(VP + VN + FP + FN)^2} + \frac{(FN + VN).(FP + VN)}{(VP + VN + FP + FN)^2} \quad (4.3)$$

### 4.4.3 Précision

La précision correspond à la proportion de prédictions positives qui étaient correctes. Elle se formule selon (4.4) :

$$Precision = \frac{VP}{VP + FP} \quad (4.4)$$

## 4.5 Expériences et Résultats

Afin d'évaluer l'efficacité de nos deux modèles, '*FCN-VGG-16 adapté comme extracteur de caractéristiques*' et '*FCN-siamois ED&DI sans maxpooling*', nous utiliserons les jeux de données '*Dataset1*', '*Dataset2*' et '*Dataset3*' créés dans notre base de données.

Nous détaillerons dans ce qui suit l'implémentation de nos méthodes et discuterons les résultats obtenus, tout en évoquant les différents tests que nous avons mené au cours de notre travail, et qui nous ont permis d'améliorer la conception des modèles finaux.

### 4.5.1 Implémentation et résultats de la méthode "*FCN-VGG-16 adapté comme extracteur de caractéristiques*"

Pour cette première approche, nous compilons le modèle '*FCN-VGG-16 adapté comme extracteur de caractéristiques*' en utilisant l'optimiseur "*ADAM*" et la fonction de perte "*binary\_crossentropy*".

Les résultats obtenus, ainsi que ceux résultants du même réseau avec maxpooling et d'un réseau VGG-16 classique, sont illustrés dans les figures (4.3, 4.4 et 4.5) pour chacune des paires d'images de test respectivement.

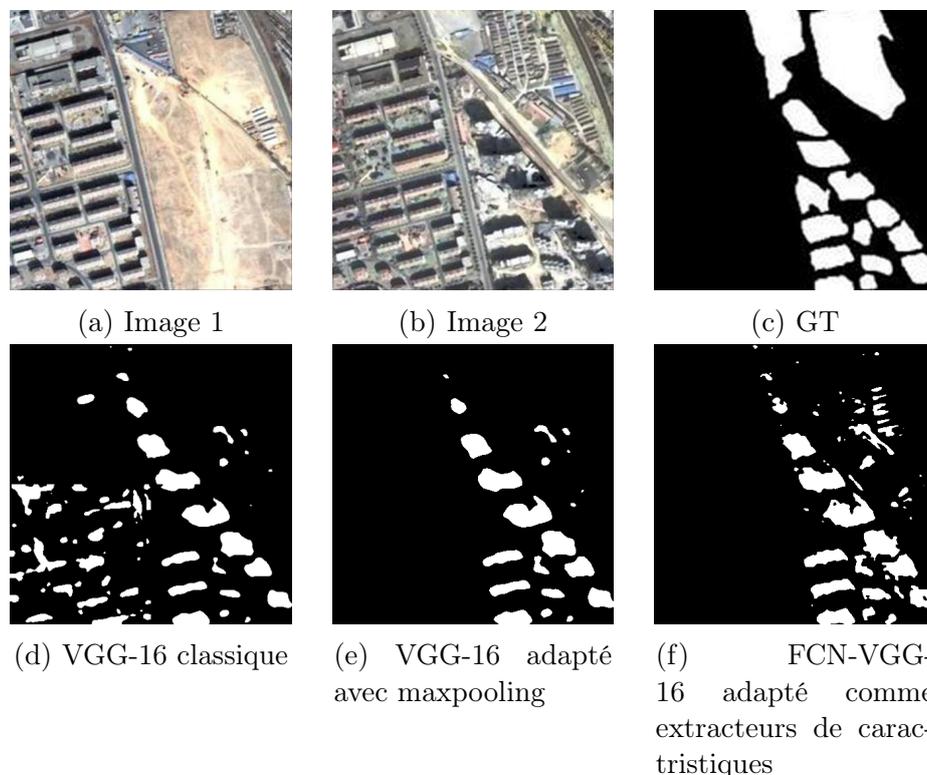


FIGURE 4.3 – Résultats des tests sur Dataset1. Les figures (a), (b) et (c) représentent dans l'ordre l'image 1, l'image 2 et le GT correspondant. Les figures (d), (e) et (f) sont respectivement les résultats des méthodes '*Extracteur de features par VGG-16 classique*', '*VGG-16 adapté avec maxpooling*' et la méthode proposée.

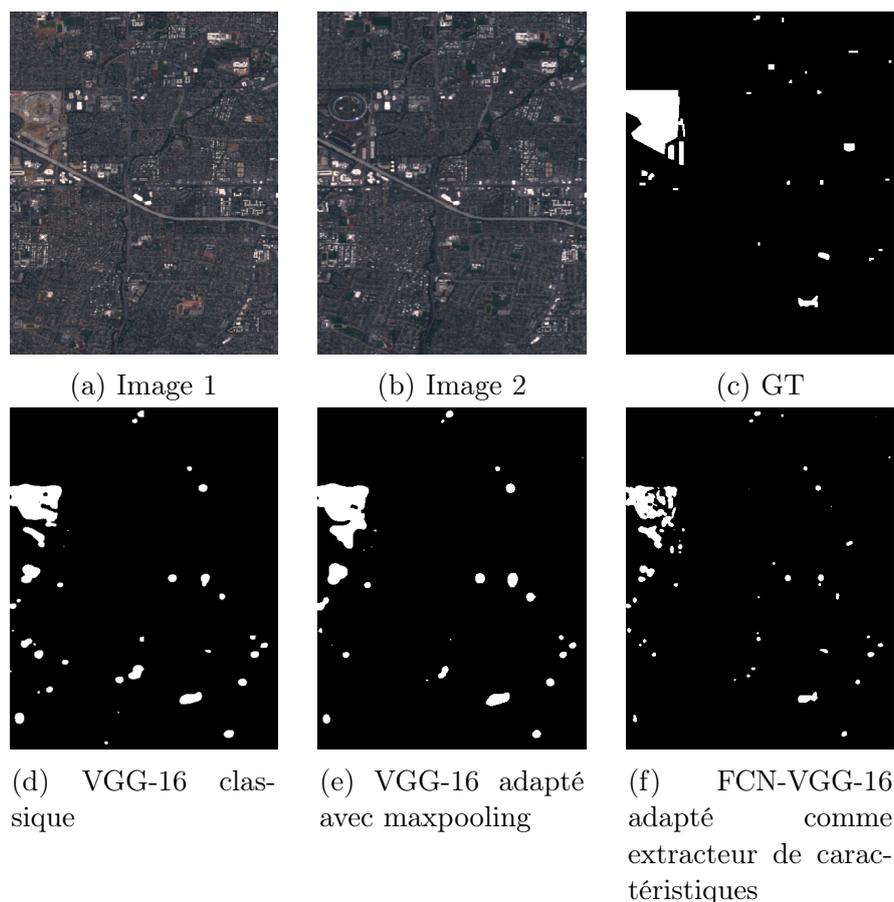


FIGURE 4.4 – Résultats des tests sur *Dataset2*. Les figures (a), (b) et (c) représentent dans l'ordre l'image 1, l'image 2 et le GT correspondant. Les figures (d), (e) et (f) sont respectivement les résultats des méthodes '*Extracteur de features par VGG-16 classique*', '*VGG-16 adapté avec maxpooling*' et la méthode proposée.

Les tableaux suivants (Table 4.1, Table 4.2 et Table 4.3) contiennent l'évaluation quantitative de notre architecture suivant les critères d'évaluation, à savoir le CA, le coefficient Kappa et la précision pour chacune des paires d'images précédentes, respectivement. Ils montrent également les résultats obtenus à partir de la méthode par VGG-16 classique, ainsi que l'effet des couches de maxpooling sur le modèle proposé.

	CA	Kappa	Précision
VGG-16 classique	75,37%	17,82%	46,05%
VGG-16 adapté avec MaxPooling	81,36%	32,13%	<b>85,22%</b>
FCN-VGG-16 adapté comme extracteur de caractéristiques	<b>81,65%</b>	<b>35,69%</b>	78,94%

TABLE 4.1 – Comparaison quantitative des approches sur Dataset1

	CA	Kappa	Précision
VGG-16 classique	96,48%	51,04%	52,65%
VGG-16 adapté avec MaxPooling	97,45%	<b>64,75%</b>	65,45%
FCN-VGG-16 adapté comme extracteur de caractéristiques	<b>97,71%</b>	61,31%	<b>80,28%</b>

TABLE 4.2 – Comparaison quantitative des approches sur Dataset2

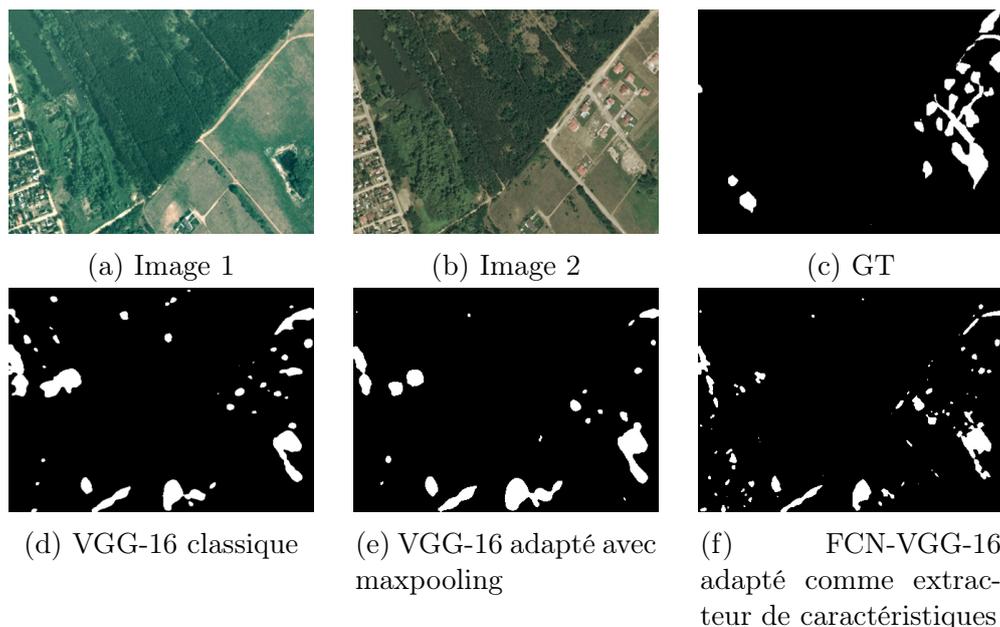


FIGURE 4.5 – Résultats des tests sur *Dataset3*. Les figures (a), (b) et (c) représentent dans l'ordre l'image 1, l'image 2 et le GT correspondant. Les figures (d), (e) et (f) sont respectivement les résultats des méthodes '*Extracteur de features par VGG-16 classique*', '*VGG-16 adapté avec maxpooling*' et la méthode proposée.

	CA	Kappa	Précision
VGG-16 classique	90,49%	14,68%	18,80%
VGG-16 adapté avec MaxPooling	92,09%	18,79%	22,09%
FCN-VGG-16 adapté comme extracteur de caractéristiques	<b>93,82%</b>	<b>28,48%</b>	<b>34,82%</b>

TABLE 4.3 – Comparaison quantitative des approches sur *Dataset3*

Pour ces deux prochains tableaux (Table 4.4 et Table 4.5), on compare les résultats de notre modèle ainsi que le modèle avec maxpooling sur '*Dataset2*', en choisissant différentes couches convolutives pour former les hypercolonnes : les premières layers, les layers intermédiaires et les layers supérieures.

	CA	Kappa	Précision
Premières couches	97,43%	54,42%	<b>77,33%</b>
Couches intermédiaires	<b>97,45%</b>	<b>64,75%</b>	65,45%
Couches supérieures	96,07%	55,80%	47,97%

TABLE 4.4 – L'effet du choix des couches convolutives avec maxpooling à concaténer dans les hypercolonnes pour '*Dataset2*'

	CA	Kappa	Précision
Premières couches	97,33%	51,42%	76,86%
Couches intermédiaires	97,71%	61,31%	80,28%
Couches supérieures	<b>98,08%</b>	<b>68,71%</b>	<b>84,02%</b>

TABLE 4.5 – L'effet du choix des couches convolutives sans maxpooling à concaténer dans les hypercolonnes pour '*Dataset2*'

### 4.5.2 Implémentation et résultats de la méthode *FCN-siamois ED&DI sans maxpooling*

Pour notre 2<sup>ème</sup> approche, nous entraînons le modèle siamois avec les images du *Training\_Set* créé dans notre base de données et dont les images sont augmentées en variant l'éclairage dans un intervalle [0.6, 1.0]. Nous utilisons également la technique de *dropout* pour éviter le surapprentissage. Nous compilons le modèle à travers un optimiseur *ADAM* de *learning\_rate* = 0.001, de *batch\_size* = 16 et de fonction de perte "*binary\_crossentropy*" et entamons l'entraînement pour environ 50 *epochs* dont les graphes de '*loss*' et de '*precision*' sont illustrés dans la Figure 4.6.

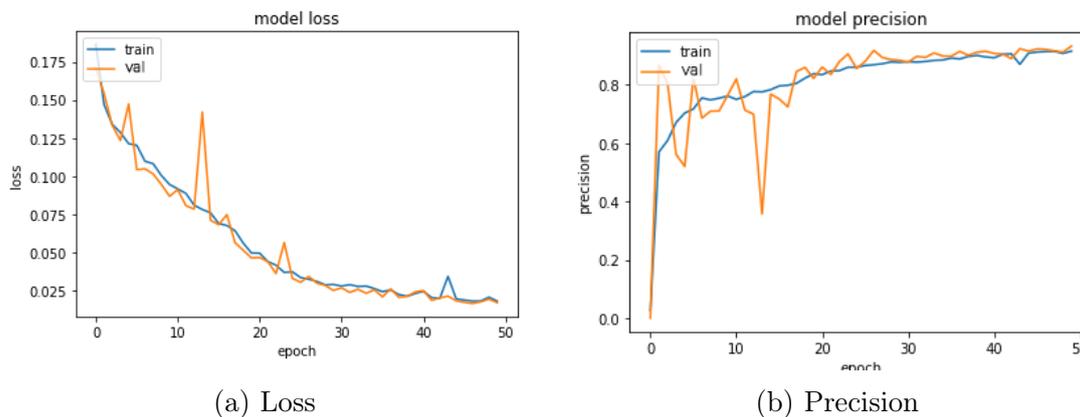


FIGURE 4.6 – Graphes représentant l'évolution de '*loss*' et '*précision*' respectivement lors de l'entraînement du réseau.

Les Figures (4.7, 4.8 et 4.9) et les tableaux (4.6, 4.7 et 4.8) suivants présentent les résultats obtenus pour notre modèle, ainsi que ceux issus d'autres alternatives, à savoir :

- Modèle FCN siamois utilisant que la Distance Euclidienne dans les skip connections et avec maxpooling.
- Modèle FCN siamois utilisant que la Distance Euclidienne dans les skip connections et sans maxpooling.
- Modèle FCN siamois concaténant la Distance Euclidienne et la Différence d'Image dans les skip connections et avec utilisation des couches de maxpooling.

Ces cartes de changement binaires permettent ainsi une meilleure interprétation des valeurs résultantes et une visibilité sur l'apport de chaque méthode.

	CA	Kappa	Précision
FCN-Siamois ED avec MaxPooling	77,84%	39,95%	52,99%
FCN-Siamois ED sans MaxPooling	78,21%	47,55%	52,76%
FCN-Siamois ED&DI avec MaxPooling	77,07%	37,00%	51,51%
FCN-Siamois ED&DI sans MaxPooling	<b>82,18%</b>	<b>53,62%</b>	<b>60,42%</b>

TABLE 4.6 – Résultats de la 2ème approche sur '*Dataset1*'

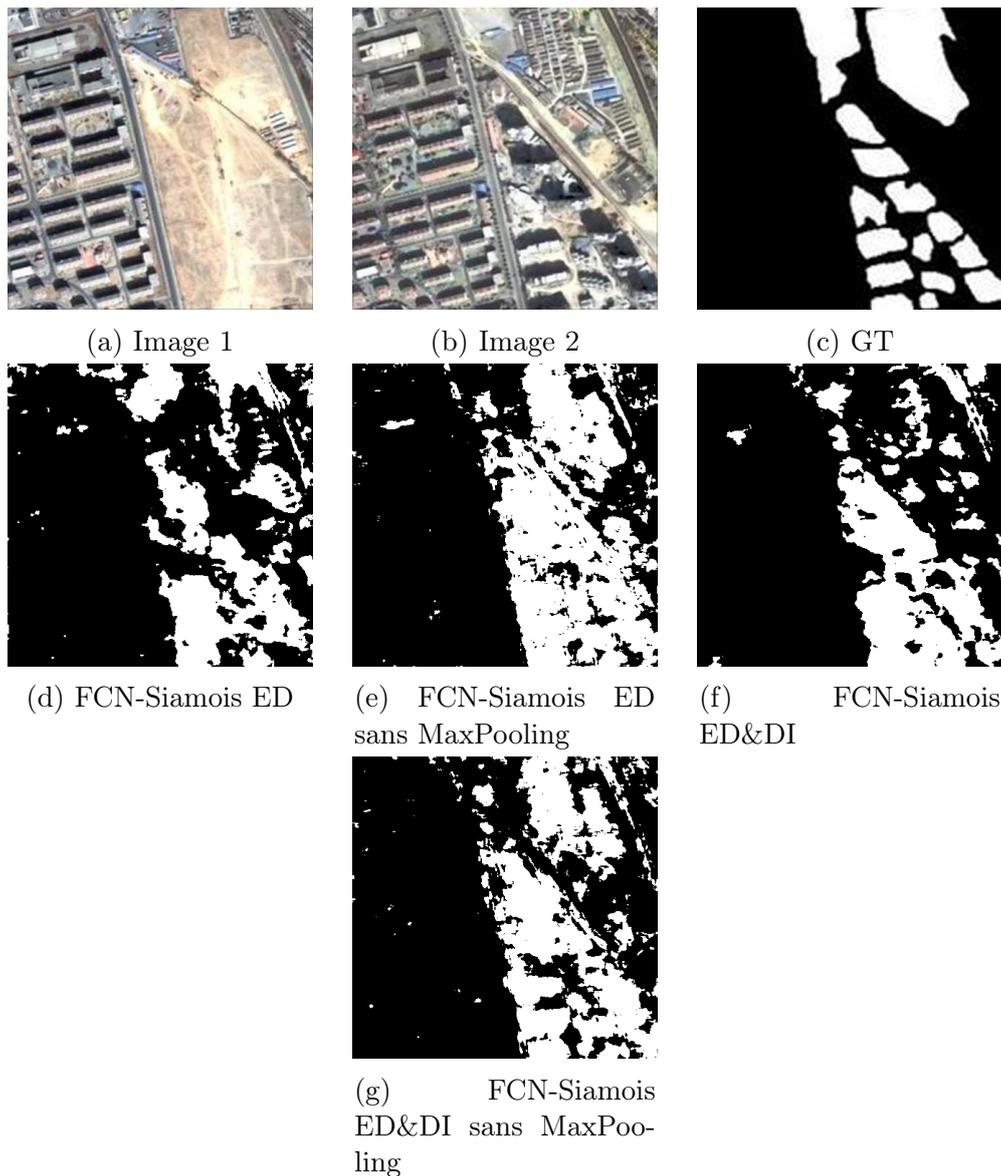


FIGURE 4.7 – Résultats des tests sur 'Dataset1'. Les figure (a), (b) et (c) représentent dans l'ordre l'image 1, l'image 2 et le GT correspondant. Les figures (d), (e) (f) et (g) sont respectivement les résultats des méthodes FCN-Siamois ED et FCN-Siamois ED sans maxpooling, FCN-Siamois ED&DI et la méthode proposée.

	CA	Kappa	Précision
FCN-Siamois ED avec MaxPooling	96,38%	19,62%	55,76%
FCN-Siamois ED sans MaxPooling	<b>96,73%</b>	24,46%	<b>84,01%</b>
FCN-Siamois ED&DI avec MaxPooling	96,33%	05,05%	66,01%
FCN-Siamois ED&DI sans MaxPooling	<b>96,73%</b>	<b>24,61%</b>	83,54%

TABLE 4.7 – Résultats de la 2ème approche sur 'Dataset2'

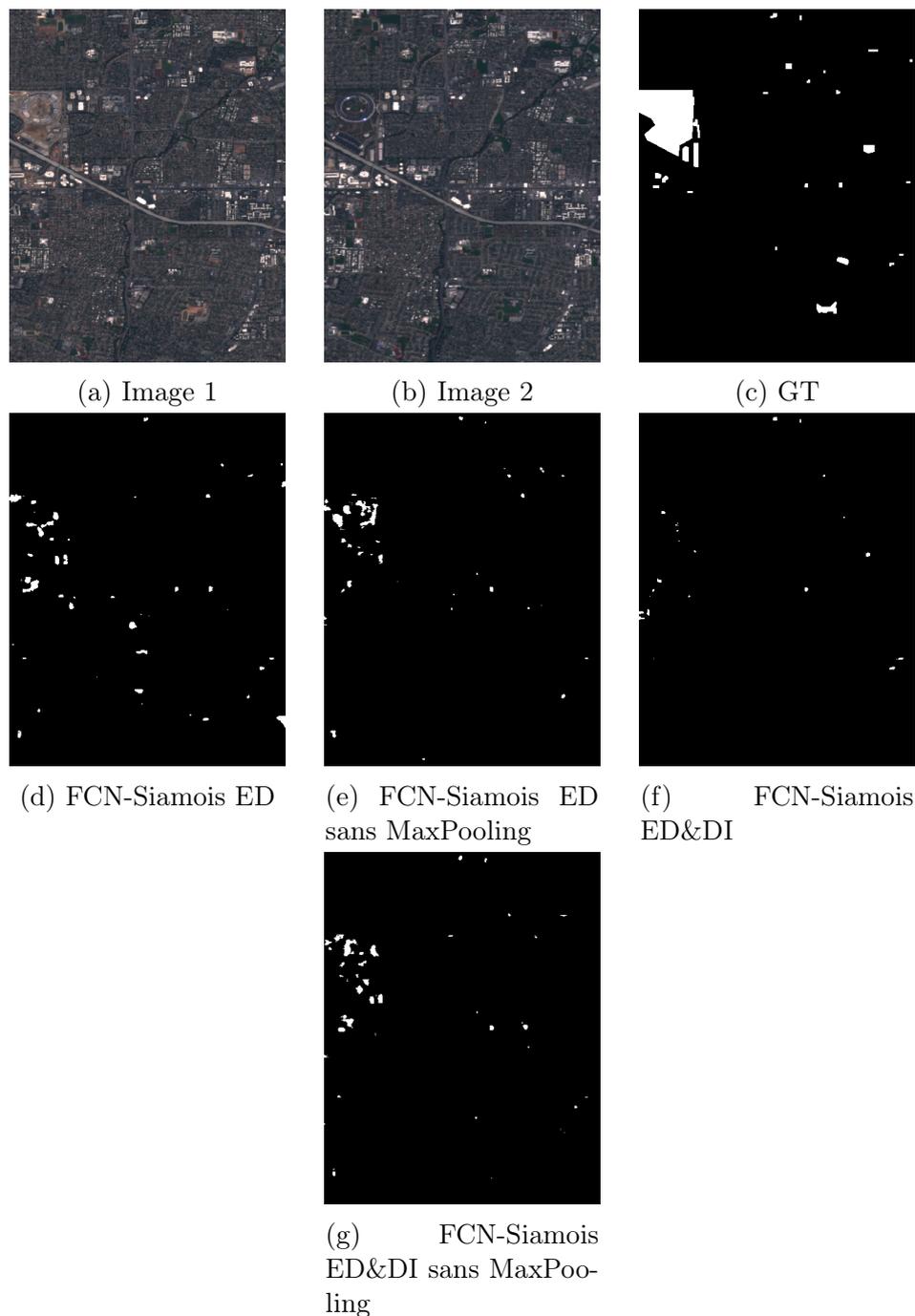


FIGURE 4.8 – Résultats des tests sur 'Dataset2'. Les figure (a), (b) et (c) représentent dans l'ordre l'image 1, l'image 2 et le GT correspondant. Les figures (d), (e) (f) et (g) sont respectivement les résultats des méthodes FCN-Siamois ED et FCN-Siamois ED sans maxpooling, FCN-Siamois ED&DI et la méthode proposée.

	CA	Kappa	Précision
FCN-Siamois ED avec MaxPooling	95,01%	57,82%	49,72%
FCN-Siamois ED sans MaxPooling	95,73%	60,81%	55,15%
FCN-Siamois ED&DI avec MaxPooling	95,42%	60,01%	52,51%
FCN-Siamois ED&DI sans MaxPooling	<b>95,82%</b>	<b>61,91</b>	<b>55,77%</b>

TABLE 4.8 – Résultats de la 2ème approche sur 'Dataset3'

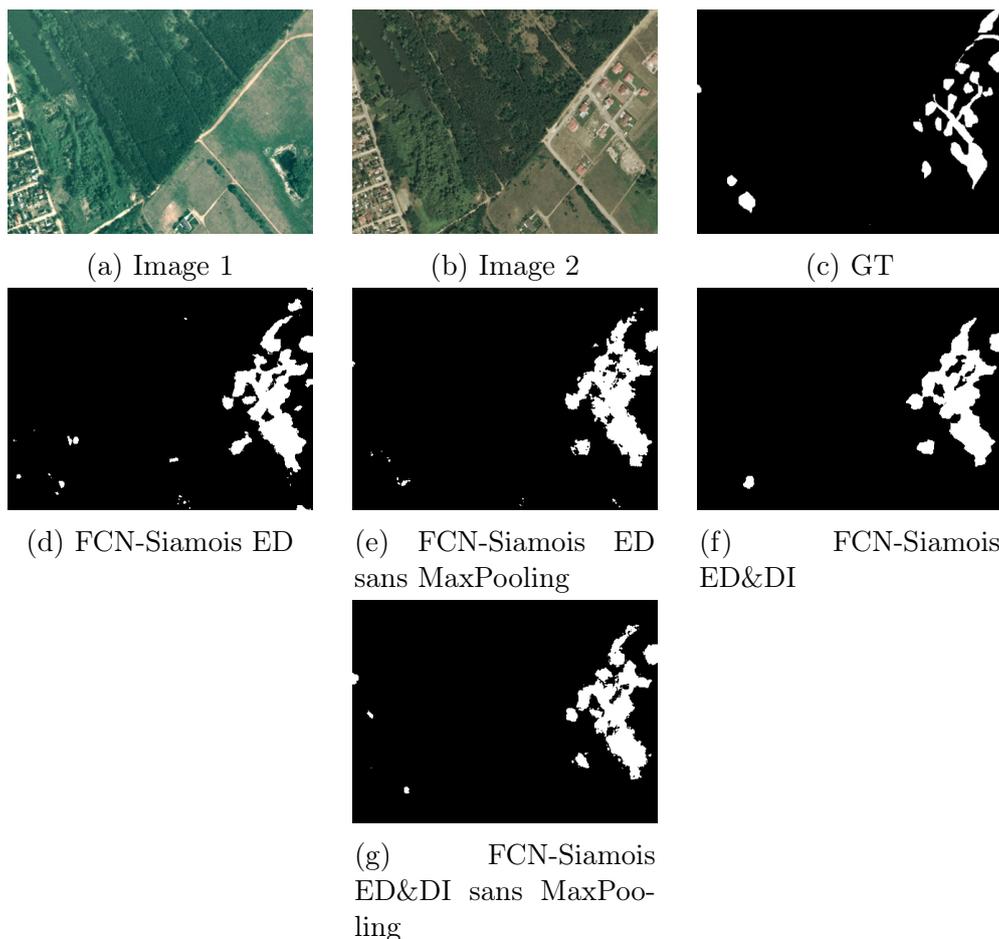


FIGURE 4.9 – Résultats des tests sur 'Dataset3'. Les figure (a), (b) et (c) représentent dans l'ordre l'image 1, l'image 2 et le GT correspondant. Les figures (d), (e) (f) et (g) sont respectivement les résultats des méthodes FCN-Siamois ED et FCN-Siamois ED sans maxpooling, FCN-Siamois ED&DI et la méthode proposée.

### 4.5.3 Interprétation des résultats

Dans ce qui suit, les différents résultats obtenus des tests effectués seront analysés pour chaque méthode. Nous discuterons également de l'apport de ces dernières à la résolution de la problématique de la DC en imagerie satellitaire.

#### 4.5.3.1 1ère Méthode | FCN-VGG-16 adapté comme extracteur de caractéristiques :

Les résultats relatifs à cette première méthode montrent que l'extraction de caractéristiques à partir du VGG-16 modifié en FCN, n'ayant pas recours aux couches de maxpooling atteint des résultats satisfaisants, dépassant en moyenne ceux des variantes basées sur le réseau VGG-16 classique [3] ou encore celle utilisant un FCN avec les couches de maxpooling et ceci, principalement en terme de CA mais aussi pour la précision et le coefficient de kappa moyens.

Dans les tableaux (4.4 et 4.5), on remarque l'effet des différentes couches convolutives de l'hypercolonne sur les résultats finaux. En effet, dans le cas où les opérations de maxpooling sont exploitées, l'utilisation de couches intermédiaires dans l'hypercolonne mène vers une certaine amélioration des résultats. Ceci est dû au fait que les caractéristiques spatiales et sémantiques soient prises en compte lors du calcul de la distance euclidienne.

Quant à notre cas, où les opérations de maxpooling ont été omises, l'utilisation des couches supérieures présente de meilleurs résultats comparés aux autres. Car, en plus de l'information plus spécifique se trouvant dans les caractéristiques extraites de ces dernières couches, la précision spatiale y est aussi conservée, ce qui aboutit à une meilleure mesure de similarité pour former la carte de changement finale.

Cependant, au vu de la RAM limitée, nous nous sommes contentées d'utiliser les couches intermédiaires, reliant ainsi les meilleures caractéristiques pour les différentes variantes de cette méthode non supervisée.

#### 4.5.3.2 2ème Méthode | FCN-siamois ED&DI sans maxpooling :

Avec son architecture siamoise plus adaptée à la comparaison de paires d'images, et l'avantage des *skip-connections* permettant l'exploitation de la différence absolue et de la distance euclidienne pour la mesure de similarité au niveau du pixel, cette deuxième méthode supervisée nommée '*FCN-siamois ED&DI sans maxpooling*' améliore notablement les résultats et démontre son efficacité par rapport aux autres alternatives des FCN siamois testées pour les mêmes images.

La première variante testée consistait à réutiliser la même mesure de similarité que la première méthode 4.5.1, ie la distance euclidienne. Nous avons amélioré les résultats en ôtant les couches de maxpooling et en concaténant la DI avec la ED, ce qui a formé l'architecture finale proposée comme deuxième solution.

Effectivement, pour les Dataset 1, 2 et 3, on constate dans les tableaux (4.6, 4.7 et 4.8) que notre méthode surpasse notablement les autres variantes en terme de CA, de précision et de coefficient de Kappa. Ces mesures prouvent que l'association des DI aux ED calculées augmentent la performance de la méthode.

### 4.5.4 Bilan et Comparaison

Dans les sections précédentes, deux solutions pour la DC en imagerie satellite ont été présentées avec les résultats interprétés correspondants à chacune ; la première est une méthode non supervisée basée sur l'extraction de caractéristiques et la deuxième est une méthode supervisée qui se base sur l'entraînement d'un réseau siamois FCN adapté sur une base de données étiquetées.

Les architectures adoptées pour les deux méthodes sont des architectures entièrement convolutives, ne possédant pas de couches de Pooling. Pour les deux, la suppression de ces dernières augmente considérablement la précision et la robustesse comme le montre les critères d'évaluation. Par ailleurs, cette suppression augmente le temps d'exécution et d'entraînement à cause de la non diminution des dimensions et donc du nombre de paramètres du réseau. Ces architectures restent toutefois plus optimales en terme de temps grâce à l'élimination des couches FC.

La première méthode '*FCN-VGG-16 adapté comme extracteur de caractéristiques*' permet la DC avec des scores satisfaisants, notamment pour les images satellitaires des zones urbaines comme on peut le voir dans les résultats et ce, sans avoir recours à une quelconque base de données étiquetées ni à un entraînement préalable à l'utilisation. Elle reste toutefois lente et demande un temps d'exécution assez long pour une seule paire d'images comparé à la seconde méthode.

La méthode supervisée, '*FCN-siamois ED&DI sans maxpooling*', propose une solution nettement plus rapide que la première basée sur l'extraction des caractéristiques. En effet, grâce au GPU accordé par Google Colab, l'entraînement se fait en un temps raisonnable et permet par la suite de prédire la carte de changement d'une paire d'images en l'espace de quelques secondes pour un résultat satisfaisant, soit en moyenne 40x plus rapidement que le FCN-VGG-16 adapté. Dans le cas des tests choisis, le «*FCN-siamois ED&DI sans Maxpooling*» a également démontré être plus robuste que la première méthode, et donne des résultats satisfaisants sur différents types d'images, on cite les images tests des zones agricoles et urbaines.

La Figure 4.10 illustre les résultats moyens obtenus pour les différents critères d'évaluation CA, précision et kappa, respectivement, pour toutes les variantes testées sur nos paires d'images de test. Nos modèles finaux démontrent leur supériorité en terme de précision et de robustesse au VGG-16 classique comme extracteur de caractéristiques choisi comme référence. De plus, elles optimisent notablement le temps nécessaire pour la génération de la carte de changement en comparaison avec cette référence.

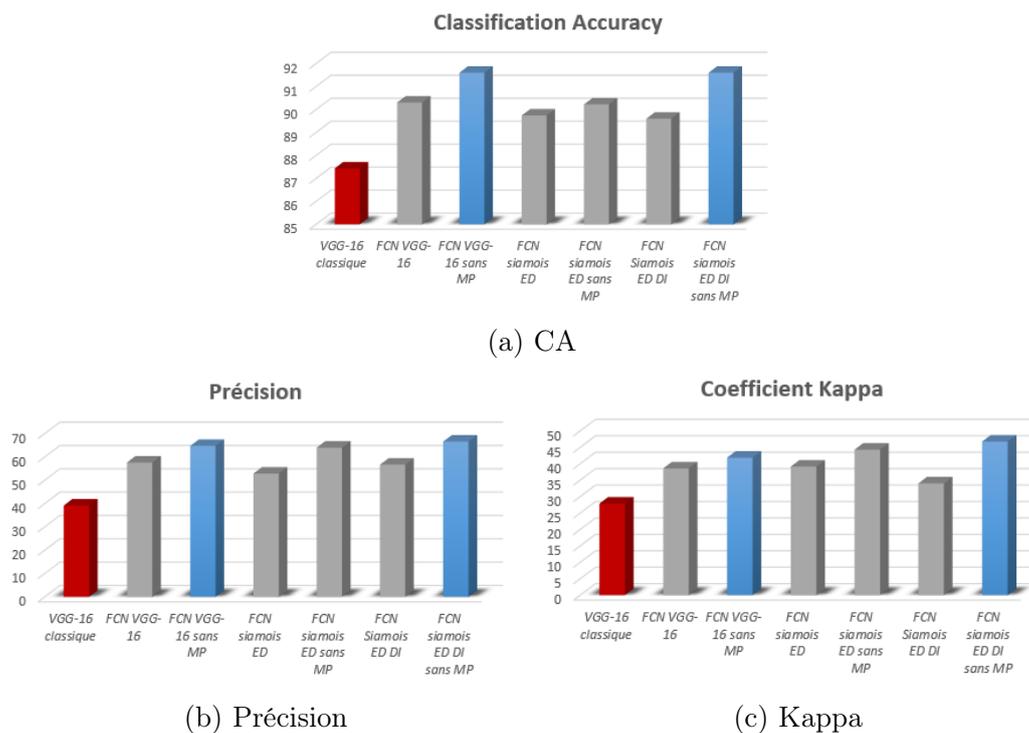


FIGURE 4.10 – Illustration des valeurs moyennes des critères d'évaluations pour les différentes variantes testées

Depuis la Figure 4.10, on constate également que les résultats du '*FCN-VGG-16 adapté comme extracteur de caractéristiques*' et du '*FCN-siamois ED&DI sans maxpooling*' sont considérablement proches. Possédant l'une comme l'autre des avantages et inconvénients, le choix de l'approche dépend donc du contexte auquel est confronté l'utilisateur ou le constructeur d'un des systèmes, on cite le type et la disponibilité des données, la contrainte du temps ainsi que l'objectif ciblé par la DC. On note toutefois, que le deuxième système expose les concepteurs à un grand obstacle, celui d'avoir recours à une très large base de données annotées et diversifiées, afin de pouvoir obtenir la meilleure généralisation possible lors de l'entraînement. Nous estimons donc que les résultats de cette deuxième

méthode s'amélioreraient davantage dans le cas de l'élargissement et de la diversification de la base de données utilisée, et surpasseraient par conséquent ceux de la première.

## 4.6 Conclusion

Au cours de ce chapitre, nous avons présenté la base de données sur laquelle nous avons entraîné certaines approches et testé leurs résultats, et introduit les différents outils qui nous ont permis de concrétiser notre travail.

Par la suite, nous avons expliqué l'implémentation de nos méthodes et exposé leurs résultats. Nous avons pu également interpréter ces derniers en s'appuyant sur des critères d'évaluation ainsi que sur les figures illustrant les cartes de caractéristiques résultantes.

Nous terminons le chapitre sur un bilan comparatif des deux méthodes, à travers lequel nous estimons que la deuxième approche, '*FCN-siamois ED&DI sans maxpooling*', soit plus adaptée et plus robuste pour la DC, et délivre, en dépit du manque de données d'entraînement, de meilleurs résultats par rapport à la 1ère méthode '*FCN-VGG-16 adapté comme extracteur de caractéristiques*' et ce, en un temps considérablement réduit.

# Conclusion Générale

La détection de changement en imagerie satellitaire est une problématique complexe, qui, jusqu'à aujourd'hui fait face à de multiples challenges. Comme nous l'avons exposé auparavant dans ce document, l'apprentissage automatique profond a contribué à l'avancement de ce domaine. Des systèmes basés sur le transfert d'apprentissage, la fusion de données et les modèles convolutifs ont été adoptés dans plusieurs travaux.

Dans notre travail, nous proposons deux approches de DC testées sur les mêmes paires d'images, ne nécessitant aucun prétraitement ou régulation : la première est une méthode non supervisée qui se base sur l'extraction de caractéristiques d'un modèle VGG-16 adaptée entièrement convolutif, pré-entraîné sur la base de données ImageNet. La deuxième est une méthode adoptant une architecture siamoise de poids partagés, entièrement convolutive également, qui se base sur l'entraînement à partir de zéro, en exploitant une base de données d'images satellitaires urbaines de type RGB.

A partir des résultats de notre travail, nous concluons les points suivants :

- Les modèles pré-entraînés sur les grandes bases de données telles que ImageNet peuvent être adaptés à des tâches plus complexes, différentes de la classification comme la DC. En apportant au réseau les modifications nécessaires et en ajoutant une mesure de similarité, ces modèles peuvent donner des résultats très satisfaisants grâce à leur profondeur.
- Les réseaux siamois sont plus robustes pour une tâche de comparaison d'images bitemporelles, compte tenu de leur architecture adaptée à la mesure de similarité. Cependant, le manque de larges bases de données diminue leur capacité à la généralisation.
- Pour les deux approches, la modification des CNNs en supprimant les couches de maxpooling, améliorent les résultats visuels et analytiques. Ceci s'explique par le fait que les deux approches se basent sur la comparaison au niveau du pixel, ainsi comme les deux systèmes comparent les images au niveau du pixel et génèrent des sorties spatiales, la suppression des couches de maxpooling aide donc à avoir une meilleure précision spatiale.
- La problématique de la DC reste toujours complexe et relative. L'efficacité et la robustesse des méthodes varient pour différentes zones étudiées, et dépendent des bases de données utilisées pour l'entraînement, car les types de changements détectés par les systèmes conçus pour la DC restent relatifs à ces dernières.

Pour la réalisation de ce travail, nous avons rencontré plusieurs difficultés, on cite :

- L'indisponibilité d'un grand nombre d'images satellitaires bi-temporelles.
- L'indisponibilité de grandes bases de données de référence, étiquetées accessibles et adaptés à la tâche de la DC.
- L'accès limité à des GPU puissants et la limitation de la RAM à 12.5 GB sur Google Colab, ont fait que le choix des hyper paramètres pour la deuxième approche soit conditionné, ainsi que le choix du nombre des couches utilisées pour l'extraction des caractéristiques pour la première approche.

Enfin, Nous concluons notre travail par des recommandations et des remarques pour la possibilité d'amélioration des modèles présentés, ainsi que pour des futurs travaux indépendants dans le domaine :

Pour le premier modèle adopté, l'extraction des caractéristiques d'un réseau profond pré-entraîné était le centre d'intérêt de l'approche. L'utilisation d'un réseau pré-entraîné sur une large base de données d'images satellitaires pour une tâche de segmentation est envisageable pour l'amélioration des résultats du premier système.

Deuxièmement, l'utilisation des réseaux siamois plus profonds sur des images de plus haute résolution et en plus grand nombre, amélioreraient la précision et la robustesse du système. Par ailleurs, on peut aussi envisager de changer la stratégie d'entraînement, en utilisant des fonctions de pertes adaptées à la mesure de similarité, à savoir, la perte contrastive et la perte de triplet.

Nous voulons également suggérer un autre type de couches convolutives qui pourrait être efficace pour l'entraînement des réseaux siamois adaptés à la DC. Ces couches, appelées les « *Atrous Convolutions* », sont généralement utilisées pour la segmentation d'images. Nous considérons que de futurs travaux peuvent exploiter cette notion pour la DC, notamment pour la DC sémantique.

En dernier lieu, nous espérons adapter notre travail à des problèmes concrets en Algérie, et pouvoir contribuer à l'analyse et à la prise de décision, en particulier dans le domaine de l'aménagement du territoire et de la régulation et contrôle de l'expansion des zones urbaines, où des solutions y sont de plus en plus sollicitées.

# Références Bibliographiques

- [1] A. SINGH, “Review article digital change detection techniques using remotely-sensed data”, *International journal of remote sensing*, t. 10, n° 6, p. 989-1003, 1989.
- [2] M. HUSSAIN, D. CHEN, A. CHENG, H. WEI et D. STANLEY, “Change detection from remotely sensed images : From pixel-based to object-based approaches”, *ISPRS Journal of photogrammetry and remote sensing*, t. 80, p. 91-106, 2013.
- [3] A. M. EL AMIN, Q. LIU et Y. WANG, “Convolutional neural network features based change detection in satellite images”, in *First International Workshop on Pattern Recognition*, International Society for Optics et Photonics, t. 10011, 2016, 100110W.
- [4] G. CHEN, G. J. HAY, L. M. CARVALHO et M. A. WULDER, “Object-based change detection”, *International Journal of Remote Sensing*, t. 33, n° 14, p. 4434-4457, 2012.
- [5] K. SAKURADA et T. OKATANI, “Change Detection from a Street Image Pair using CNN Features and Superpixel Segmentation.”, in *BMVC*, t. 61, 2015, p. 1-12.
- [6] *Mohammed El Amin Larabi "Multi-Temporal Remote Sensing Change Detection Using Deep Learning ", Université de Beihang, Pékin, CHINE, 2018.*
- [7] K. L. de JONG et A. S. BOSMAN, “Unsupervised change detection in satellite images using convolutional neural networks”, in *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2019, p. 1-8.
- [8] Y. BAZI, F. MELGANI et H. D. AL-SHARARI, “Unsupervised change detection in multispectral remotely sensed imagery with level set methods”, *IEEE Transactions on Geoscience and Remote Sensing*, t. 48, n° 8, p. 3178-3187, 2010.
- [9] J. AGUIRRE-GUTIÉRREZ, A. C. SEIJMONSBERGEN et J. F. DUIVENVOORDEN, “Optimizing land cover classification accuracy for change detection, a combined pixel-based and object-based approach in a mountainous area in Mexico”, *Applied Geography*, t. 34, p. 29-37, 2012.
- [10] K. LIM, D. JIN et C.-S. KIM, “Change detection in high resolution satellite images using an ensemble of convolutional neural networks”, in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2018, p. 509-515.
- [11] B. SHIVAKUMAR et al., “Change detection using image differencing : A study over area surrounding Kumta, India”, in *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, IEEE, 2017, p. 1-5.
- [12] C. CAO, S. 114 DRAGIĆEVIĆ et S. LI, “Land-use change detection with convolutional neural network methods”, *Environments*, t. 6, n° 2, p. 25, 2019.
- [13] A. M. EL AMIN, Q. LIU et Y. WANG, “Zoom out CNNs features for optical remote sensing change detection”, in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, IEEE, 2017, p. 812-817.

- [14] R. SHARMA, A. V. NORI et A. AIKEN, "Bias-variance tradeoffs in program analysis", *ACM SIGPLAN Notices*, t. 49, n° 1, p. 127-137, 2014.
- [15] S. SAHA, F. BOVOLO et L. BRURZONE, "Unsupervised multiple-change detection in VHR optical images using deep features", in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2018, p. 1902-1905.
- [16] M. E. A. LARABI, S. CHAIB, K. BAKHTI et M. S. KAROUI, "Transfer Learning for Changes Detection in Optical Remote Sensing Imagery", in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2019, p. 1582-1585.
- [17] R. C. DAUDT, B. LE SAUX, A. BOULCH et Y. GOUSSEAU, "Urban change detection for multispectral earth observation using convolutional neural networks", in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2018, p. 2115-2118.
- [18] F. RAHMAN, B. VASU, J. VAN COR, J. KEREKES et A. SAVAKIS, "Siamese network with multi-level features for patch-based change detection in satellite imagery", in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, 2018, p. 958-962.
- [19] R. SU et R. CHEN, "Land Cover Change Detection via Semantic Segmentation", *arXiv preprint arXiv :1911.12903*, 2019.
- [20] *Review : DeepLabv3+ — Atrous Separable Convolution (Semantic Segmentation)*, <https://medium.com/@sh.tsang/review-deeplabv3-atrous-separable-convolution-semantic-segmentation-a625f6e83b90>, Consulté le : 07-06-2020.
- [21] *Change Detection using Deep Learning and Machine Learning Techniques for Multispectral Satellite Images*, [https://www.researchgate.net/publication/338013906\\_Change\\_Detection\\_using\\_Deep\\_Learning\\_and\\_Machine\\_Learning\\_Techniques\\_for\\_Multispectral\\_Satellite\\_Images](https://www.researchgate.net/publication/338013906_Change_Detection_using_Deep_Learning_and_Machine_Learning_Techniques_for_Multispectral_Satellite_Images), Consulté le : 01-09-2020.
- [22] Y. LECUN, L. BOTTOU, Y. BENGIO et P. HAFFNER, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, t. 86, n° 11, p. 2278-2324, 1998.
- [23] J DENG, W DONG, R SOCHER, L. LI, K LI et L. FEI-FEI, "2009 IEEE conference on computer vision and pattern recognition", 2009.
- [24] Y. ZHAN, K. FU, M. YAN, X. SUN, H. WANG et X. QIU, "Change detection based on deep siamese convolutional network for optical aerial images", *IEEE Geoscience and Remote Sensing Letters*, t. 14, n° 10, p. 1845-1849, 2017.
- [25] C. RAZAFIMANDIMBY, V. LOSCRI et A. M. VEGNI, "A neural network and IoT based scheme for performance assessment in internet of robotic things", in *2016 IEEE first international conference on internet-of-things design and implementation (IoTDI)*, IEEE, 2016, p. 241-246.
- [26] S SAPNA, A TAMILARASI, M. P. KUMAR et al., "Backpropagation learning algorithm based on Levenberg Marquardt Algorithm", *Comp Sci Inform Technol (CS and IT)*, t. 2, p. 393-398, 2012.
- [27] C. M. BISHOP, *Pattern recognition and machine learning*. springer, 2006.
- [28] AIT GANA Zakaria "Classification des phases de la marche humaine par apprentissage profond – Application au domaine médical ", *Ecole Supérieure d'Informatique, Algérie, 2019*.

- [29] *Neocognitron : A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position*, <https://www.rctn.org/bruno/public/papers/Fukushima1980.pdf>, Consulté le : 01-09-2020.
- [30] E. POISSON, C. VIARD-GAUDIN et P. LALLICAN, “Réseaux de neurones à convolution : reconnaissance de l’écriture manuscrite non contrainte”, *Revue VALGO*, In *Valgo*, n° 01-02, p. 726-730, 2001.
- [31] M. COURBARIAUX, Y. BENGIO et J.-P. DAVID, “Binaryconnect : Training deep neural networks with binary weights during propagations”, in *Advances in neural information processing systems*, 2015, p. 3123-3131.
- [32] *Charles Crouspeyre*, *Comment les réseaux de neurones à convolution fonctionnent*, <https://medium.com/@CharlesCrouspeyre/comment-les-réseaux-de-neurones-à-convolution-fonctionnent-b288519dbcf8>, Consulté le : 01-04-2020.
- [33] *Prabhu*, *Understanding of Convolutional Neural Network (CNN) - Deep Learning*, <https://medium.com/@CharlesCrouspeyre/comment-les-réseaux-de-neurones-à-convolution-fonctionnent-b288519dbcf8>, Consulté le : 08-04-2020.
- [34] *Chloé-Agathe Azencott*, *"Évaluez les performances d’un modèle de machine learning"*, <https://openclassrooms.com/fr/courses/4297211-evaluez-et-ameliorer-les-performances-d-un-modele-de-machine-learning>, Consulté le : 19-04-2020.
- [35] N. SRIVASTAVA, G. HINTON, A. KRIZHEVSKY, I. SUTSKEVER et R. SALAKHUTDINOV, “Dropout : a simple way to prevent neural networks from overfitting”, *The journal of machine learning research*, t. 15, n° 1, p. 1929-1958, 2014.
- [36] M. HEUSEL, H. RAMSAUER, T. UNTERTHINER, B. NESSLER et S. HOCHREITER, “Gans trained by a two time-scale update rule converge to a local nash equilibrium”, in *Advances in neural information processing systems*, 2017, p. 6626-6637.
- [37] J. YOSINSKI, J. CLUNE, Y. BENGIO et H. LIPSON, “How transferable are features in deep neural networks?”, in *Advances in neural information processing systems*, 2014, p. 3320-3328.
- [38] D. SARKAR, *A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning*, 2018.
- [39] K. SIMONYAN et A. ZISSERMAN, “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint arXiv :1409.1556*, 2014.
- [40] J. LONG, E. SHELHAMER et T. DARRELL, “Fully convolutional networks for semantic segmentation”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, p. 3431-3440.
- [41] *Marios Aspris*, *Is the Transposed Convolution layer and Convolution layer the same ? Experimenting with Concepts using PyTorch, 2018*. <https://towardsdatascience.com/is-the-transposed-convolution-layer-and-convolution-layer-the-same-thing-8655b751c3a1>, Consulté le : 14-08-2020.
- [42] M. D. ZEILER et R. FERGUS, “Visualizing and understanding convolutional networks”, in *European conference on computer vision*, Springer, 2014, p. 818-833.
- [43] *Sik-Ho Tsang*, *Review :Hypercolumn(Instance Segmentation), 2019*, <https://towardsdatascience.com/review-hypercolumn-instance-segmentation-367180495979>, Consulté le : 21-05-2020.
- [44] N OTSU, “AA threshold selection method from grey scale histogram”, *IEEE Transactions on Systems Man and Cybernetics*, 1979.

- 
- [45] *Contrastive Loss Explained*, <https://towardsdatascience.com/contrastive-loss-explained-159f2d4a87ec>, Consulté le : 16-08-2020.
- [46] J. BROMLEY, I. GUYON, Y. LECUN, E. SÄCKINGER et R. SHAH, "Signature verification using a " siamese" time delay neural network", in *Advances in neural information processing systems*, 1994, p. 737-744.
- [47] R. C. DAUDT, B. LE SAUX et A. BOULCH, "Fully convolutional siamese networks for change detection", in *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2018, p. 4063-4067.
- [48] *Intuitive Explanation of Skip connections in Deep Learning*, <https://theaisummer.com/skip-connections/>, Consulté le : 01-09-2020.