



Université ABBES LAGHROUR Khenchela
Faculté des Sciences et de la Technologie
Département de Génie Electrique
جامعة عباس لغرور خنشلة
كلية العلوم والتكنولوجيا
قسم الهندسة الكهربائية



N° Série :.....

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Télécommunications

Spécialité : Systèmes des Télécommunications

THEME

**Application de l'apprentissage profond
au débruitage des images radar à
synthèse d'ouverture (SAR)**

Réalisé par : LATRECH Mouhamed Djihed

Soutenu le : 14/06/2026 Devant le jury composé de:

Dr. DOUAK Fouzi

Dr. MALLEM Ali

Dr. KHEZZAR Zaki Aissam

Encadreur

Président

Examineur

Université Abbes Laghrour-Khenchela

Université Abbes Laghrour-Khenchela

Université Abbes Laghrour-Khenchela

Promotion 2025/2026

*Je dédie le fruit de ce modeste travail et l'essence de ces années de labeur et de
persévérance :*

*À la source éternelle de tendresse, à la prunelle de mes yeux... Ma chère
mère,*

*Qui m'a enveloppé de ses prières sincères au cœur de la nuit, et qui a été mon
soutien et mon refuge face à tous les obstacles.*

*À celui dont je porte fièrement le nom, mon modèle dans la vie... Mon cher
père,*

*qui ne m'a jamais privé de sa bienveillance ni de ses précieux conseils, et qui
m'a tracé le chemin du savoir par ses sacrifices et sa patience.*

À mes piliers et mes soutiens... Mes frères et sœurs,

*Ainsi qu'à tous les membres de ma précieuse famille, chacun en son nom, qui
ont partagé avec moi les moments d'anxiété et de joie.*

*À mes compagnons de route... Mes amis fidèles et mes collègues au sein de
l'Université Abbas Laghrour de Khenchela, avec qui j'ai partagé les
difficultés des études et la douceur de la réussite.*

Mouhamed Djihed

Remerciements

*Au terme de ce travail, je tiens à exprimer ma profonde gratitude et mes plus sincères remerciements à mon encadreur de mémoire, Monsieur le Dr **Douak Fouzi**. Je le remercie chaleureusement pour la confiance qu'il m'a accordée, pour sa rigueur scientifique, ainsi que pour ses précieux conseils qui ont grandement enrichi cette étude. Sa disponibilité constante et ses orientations avisées ont été un moteur essentiel à l'aboutissement de ce travail.*

Mes remerciements s'étendent également aux membres du jury qui ont accepté d'évaluer ce travail et de m'honorer de leur présence.

Enfin, je témoigne toute ma reconnaissance à ma famille et à mes collègues pour leur soutien indéfectible tout au long de mon parcours universitaire.

Résumé

L'imagerie radar à synthèse d'ouverture (SAR) est largement utilisée pour la surveillance des surfaces terrestres, des catastrophes naturelles et de l'environnement, grâce à sa capacité d'acquisition en tout temps, à sa sensibilité aux structures géométriques ainsi qu'à ses propriétés de pénétration. Cependant, le bruit de speckle, inhérent au processus d'acquisition radar, dégrade la qualité visuelle des images et complique considérablement leur analyse et leur interprétation. Par conséquent, une étape préliminaire de réduction du speckle est généralement nécessaire afin de garantir une exploitation efficace et fiable des images SAR.

Parmi les méthodes classiques de débruitage, les filtres adaptatifs tels que ceux de Lee, Kuan et Frost reposent sur des modèles statistiques locaux pour atténuer le bruit. Bien qu'ils permettent d'améliorer la qualité des images, ils entraînent souvent un lissage excessif des structures et une perte des détails fins.

Dans ce mémoire, une approche de débruitage des images SAR basée sur l'apprentissage profond, et plus particulièrement sur les réseaux de neurones convolutionnels (CNN), est proposée pour réduire efficacement le bruit speckle tout en préservant les caractéristiques importantes de l'image. Plusieurs architectures CNN ont été développées et évaluées afin d'analyser l'influence de leur structure sur les performances de débruitage. Une étude comparative a également été réalisée entre les méthodes classiques de débruitage et les approches modernes d'apprentissage profond, dans le but d'identifier les modèles les plus performants pour la réduction du bruit speckle dans les images SAR.

Mots clés: Imagerie SAR, bruit speckle, débruitage, filtres adaptatifs, méthodes non locales, apprentissage profond, réseaux de neurones convolutionnels.

Abstract

Synthetic Aperture Radar (SAR) imaging is widely used for monitoring of land surfaces, natural disasters, and the environment due to its all-time acquisition capability, its sensitivity to geometric structures, and its penetration properties. However, speckle noise, which is inherent to the radar imaging process, degrades the visual quality of SAR images and significantly complicates their analysis and interpretation. Consequently, a preliminary speckle reduction step is generally required to ensure the effective and reliable exploitation of SAR images.

Among the conventional despeckling techniques, adaptive filters such as Lee, Kuan, and Frost are based on local statistical models to suppress noise. Although these methods improve image quality, they often result in excessive smoothing of image structures and the loss of fine details.

In this thesis, a SAR image despeckling approach based on deep learning, and more specifically on Convolutional Neural Networks (CNNs), is proposed to effectively reduce speckle noise while preserving important image features. Several CNN architectures were developed and evaluated to investigate the influence of network design on despeckling performance. Furthermore, a comparative study was conducted between conventional despeckling methods and modern deep learning approaches in order to identify the most effective models for speckle noise reduction in SAR images.

Keywords: SAR imaging, speckle noise, despeckling, adaptive filters, non-local methods, deep learning, convolutional neural networks.

ملخص

تُستخدم تقنية التصوير بالرادار ذي الفتحة الاصطناعية (SAR) على نطاق واسع في مراقبة الأسطح الأرضية والكوارث الطبيعية والبيئة، وذلك بفضل قدرتها على التصوير في جميع الظروف الجوية وعلى مدار الساعة، إضافةً إلى حساسيتها للبنى الهندسية وخصائصها المتميزة في اختراق بعض الأوساط. غير أن صور SAR تتأثر بضوضاء Speckle، وهي ضوضاء ملازمة لعملية التصوير الراداري، مما يؤدي إلى ضعف الجودة البصرية للصور ويُعقّد بشكل كبير عمليات تحليلها وتفسيرها. لذلك، تُعدّ مرحلة تقليل هذه الضوضاء خطوة أساسية لضمان الاستغلال الفعّال والموثوق لبيانات SAR. من بين طرق إزالة الضوضاء التقليدية، تعتمد المرشحات التكيفية مثل مرشحات Lee و Kuan و Frost على نماذج إحصائية محلية للتقليل من تأثير الضوضاء. وعلى الرغم من قدرتها على تحسين جودة الصور، فإنها غالبًا ما تؤدي إلى تنعيم مفرط للبنى الموجودة في الصورة وفقدان التفاصيل الدقيقة.

في هذه المذكرة، تم اقتراح منهجية لإزالة ضوضاء Speckle من صور SAR بالاعتماد على التعلم العميق، وبشكل خاص على الشبكات العصبية الالتفافية (CNN). تهدف هذه المنهجية إلى تقليل ضوضاء Speckle بفعالية مع المحافظة على الخصائص المهمة للصورة. وقد تم تطوير وتقييم مختلف الشبكات العصبية الالتفافية لدراسة تأثير بنيتها على أداء إزالة الضوضاء. كما أُجريت دراسة مقارنة بين طرق إزالة الضوضاء التقليدية وأساليب التعلم العميق الحديثة بهدف تحديد النماذج الأكثر كفاءة في تقليل الضوضاء في صور الرادار ذي الفتحة الاصطناعية.

الكلمات المفتاحية : الرادار، ضوضاء، إزالة الضوضاء، المرشحات التكيفية، الطرق غير المحلية، التعلم العميق، الشبكات العصبية الالتفافية.

Liste des tableaux

Tableau I. 1. Bandes de fréquence des ondes électromagnétiques [4].....	7
Tableau IV. 1. Evaluation des performances moyennes (PSNR, SSIM, UQI) du filtre de Lee sur 150 images de test pour différents nombres de <i>looks</i> ($L=1, 4$ et 10).	77
Tableau IV. 2. Evaluation des performances moyennes (PSNR, SSIM, UQI) du filtre de Kuan sur 150 images de test pour différents nombres de <i>looks</i> ($L=1, 4$ et 10).	78
Tableau IV. 3. Evaluation des performances moyennes (PSNR, SSIM, UQI) du filtre de Frost sur 150 images de test pour différents nombres de <i>looks</i> ($L=1, 4$ et 10).	78
Tableau IV. 4. Configuration du réseau.	82
Tableau IV. 5. Evaluation des performances moyennes (PSNR, SSIM, UQI) des modèles ID-CNN sur 150 images de test pour différents nombres de <i>looks</i> ($L=1, 4$ et 10).	93

Liste des figures

Figure I. 1. Principe du radar [2].	6
Figure I. 2. Principe de fonctionnement d'un système radar [3].	6
Figure I. 3. (a) Radar SEASAT RSO, (b) Le SIR-C/X-RSO illuminant une région de la terre, (c) Modèle de l'antenne SIR-C/X-RSO.	11
Figure I. 4. Première image SAR acquise en 1957 [10].	11
Figure I. 5. Schéma d'un système SAR [10].	12
Figure I. 6. Illumination d'une scène par un SAR [11].	13
Figure I. 7. Géométrie d'une mesure SAR [11].	14
Figure I. 8. Idéalisation de la trajectoire de section linéaire du ERS-1/2 RSO [6].	16
Figure I. 9. Caractéristique de la diffusion de l'onde RSO.	17
Figure I. 10. Echantillonnage d'une image SAR : résolution en distance et résolution en azimut [13].	18
Figure I. 11. Déformations géométriques induites par le radar imageur à visée latérale [13].	18
Figure I. 12. Acquisition en mode strip-map [14].	19
Figure I. 13. Acquisition en mode Télescope [14].	20
Figure I. 14. Acquisition par affinage Doppler [14].	20
Figure I. 15. Acquisition en mode ScanSAR [14].	21
Figure I. 16. Acquisition en mode Squint [14].	22
Figure I. 17. Acquisition en mode inverse [14].	23
 Figure II. 1. Fonction de densité de probabilité de la distribution de Gamma avec : $\mu = 1$ et $L = 2,3,4$.	30
Figure II. 2. Impact du bruit de speckle sur la qualité des images SAR.	31
Figure II. 3. Restauration par le filtre de Kuan de l'image Clock.tiff: de gauche à droite : originale, corrompue par un bruit multiplicatif, restaurée pour les fenêtres: $W = 3, W = 5, W = 7, W = 9, W = 11, W = 13, W = 15$	34

Figure II. 4. Restauration par le filtre de Lee de l'image Clock.tiff: de gauche à droite : originale, corrompue par un bruit multiplicatif, restaurée pour les fenêtres: $W = 3, W = 5, W = 7, W = 9, W = 11, W = 13, W = 15$	36
Figure II. 5. Restauration par le filtre de Frost de l'image Clock.tiff de gauche à droite: originale, corrompue par un bruit multiplicatif, restaurée pour les fenêtres : $W = 3, W = 5, W = 7, W = 9, W = 11, W = 13, W = 15$	38
Figure II. 6. Principe des méthodes non locales basées sur la recherche de patches similaires.	39
Figure III. 1. Neurone biologique [50].....	45
Figure III. 2. Neurone formel.....	46
Figure III. 3. Perceptron [50].	46
Figure III. 4. Perceptron multicouches.....	47
Figure III. 5. Schéma bloc de l'apprentissage supervisé d'un réseau de neurone.	49
Figure III. 6. Schéma bloc de l'apprentissage non supervisé d'un réseau de neurone.	50
Figure III. 7. Réseaux de neurones convolutifs [54].....	51
Figure III. 8. Ensemble de neurones (cercles) créant la profondeur d'une couche de convolution (bleu). Ils sont liés à un même champ récepteur (rouge).	52
Figure III. 9. Pooling avec un filtre 2x2 et un pas de 2 [54].....	53
Figure III. 10. Schéma représente la fonction d'activation ReLU.	55
Figure III. 11. Schéma représente la fonction d'activation LeakyRelu	55
Figure III. 12. Schéma représente la fonction d'activation PRelu.	56
Figure III. 13. Schéma représente la fonction d'activation SWICH.	57
Figure III. 14. Schéma représente la fonction d'activation Sigmoid.	57
Figure III. 15. Schéma représente la fonction d'activation Tanh.....	58
Figure III. 16. Schéma représente la fonction d'activation Softmax.	58
Figure III. 17. Algorithme de descente de gradient [68].....	59
Figure III. 18. Descente de gradient stochastique [67].	60
Figure III. 19. Minima globaux Vs. Minima locaux [68].	61
Figure IV. 1. Échantillons d'images SAR affectées par le bruit Speckle.....	68

Figure IV. 2. Échantillons d'images de couverture terrestre issues de Sentinel-2.....	69
Figure IV. 3. Génération synthétique du bruit speckle.....	72
Figure IV. 4. Exemple 1, (a) image propre (clean), (b) image bruitée (noisy) avec $L=1$, (c) image bruitée (noisy) avec $L=4$, (d) image bruitée (noisy) avec $L=10$	73
Figure IV. 5. Exemple 2, (a) image propre (clean), (b) image bruitée (noisy) avec $L=1$, (c) image bruitée (noisy) avec $L=4$, (d) image bruitée (noisy) avec $L=10$	73
Figure IV. 6. Architecture du réseau ID-CNN proposée pour le débruitage des images SAR.	80
Figure IV. 7. Architecture ID-CNN-2L implémentée sous Python.....	82
Figure IV. 8. Architecture ID-CNN-3L implémentée sous Python.....	83
Figure IV. 9. Architecture ID-CNN-4L implémentée sous Python.....	84
Figure IV. 10. Architecture ID-CNN-5L implémentée sous Python.....	85
Figure IV. 11. Évolution de la perte euclidienne pour l'architecture ID-CNN-2L durant l'entraînement et la validation, accompagnée du suivi des métriques PSNR, SSIM et UQI : (a) perte euclidienne, (b) PSNR, (c) SSIM, (d) UQI.	90
Figure IV. 12. Évolution de la perte euclidienne pour l'architecture ID-CNN-3L durant l'entraînement et la validation, accompagnée du suivi des métriques PSNR, SSIM et UQI : (a) perte euclidienne, (b) PSNR, (c) SSIM, (d) UQI.	90
Figure IV. 13. Évolution de la perte euclidienne pour l'architecture ID-CNN-4L durant l'entraînement et la validation, accompagnée du suivi des métriques PSNR, SSIM et UQI : (a) perte euclidienne, (b) PSNR, (c) SSIM, (d) UQI.	91
Figure IV. 14. Évolution de la perte euclidienne pour l'architecture ID-CNN-5L durant l'entraînement et la validation, accompagnée du suivi des métriques PSNR, SSIM et UQI : (a) perte euclidienne, (b) PSNR, (c) SSIM, (d) UQI.	91
Figure IV. 15. (a) Image de test propre (clean) et images bruitées par speckle pour différents nombres de <i>looks</i> : (b) $L=1$, (c) $L=4$ et (d) $L=10$	93
Figure IV. 16. Application des filtres (a)–(c) : Lee, Kuan et Frost, et des différentes architectures ID-CNN (d)–(g) sur une image de test pour un nombre de <i>looks</i> $L=1$	95
Figure IV. 17. Application des filtres (a)–(c) : Lee, Kuan et Frost, et des différentes architectures ID-CNN (d)–(g) sur une image de test pour un nombre de <i>looks</i> $L=4$	96
Figure IV. 18. Application des filtres (a)–(c) : Lee, Kuan et Frost, et des différentes architectures ID-CNN (d)–(g) sur une image de test pour un nombre de <i>looks</i> $L=10$	97

Figure IV. 19.	Comparaison entre ID-CNN et les filtres classiques en termes de PSNR.....	98
Figure IV. 20.	Comparaison entre ID-CNN et les filtres classiques en termes de SSIM.....	98
Figure IV. 21.	Comparaison entre ID-CNN et les filtres classiques en termes de UQI.....	99
Figure IV. 22.	Application du modèle ID-CNN-5L sur les images SAR du jeu de test 1. ...	100
Figure IV. 23.	Application du modèle ID-CNN-5L sur les images SAR du jeu de test 2. ...	101

Liste des symboles

AdaDelta : Delta adaptatif.

Adam : Adaptive Moment Estimation (Estimation Adaptative du Moment).

AdaMax : Adaptive Moment Estimation-maximum variant (Estimation Adaptative du Moment Maximal).

B : La largeur de bande du signal radar.

BGD : Batch Gradient Descent (Descente de gradient par lots).

CNN : Les réseaux de neurones convolutionnels.

CONV : La couche de convolution.

CV : Coefficient de variation.

DBS: Doppler Beam Sharpening.

DGS : Stochastic Gradient Descent (Descente de gradient stochastique).

DNN : Deep Neural Network.

EQMM : L'erreur quadratique moyenne minimale.

ESA : Agence Spatiale Européenne.

GD : La descente de gradient.

GPU : Graphics Processing Unit (Processeur graphique).

GRD : Ground Range Detected.

HH : Horizontal-Horizontal (Onde émise horizontalement et reçue horizontalement).

HV : Horizontal-Vertical (Onde émise horizontalement et reçue verticalement).

IEEE : Institute of Electrical and Electronics Engineers.

IW : Interferometric Wide swath.

Leaky ReLU : Leaky Rectified Linear Unit(Unité Linéaire Rectifiée Fuyante).

LOSS : Loss Function (fonction de perte).

LR : Learning Rate (Taux d'Apprentissage adaptatif).

MAPE : Mean Absolute Percentage Error (Erreur absolue moyenne en pourcentage).

MLP : Multilayer Perceptron (Perceptron Multicouches).

MNIST : Modified National Institute of Standards and Technology.

MSLE : Mean Square Logarithmic Error (Erreur logarithmique quadratique moyenne).

Nadam : Nesterov-accelerated Adaptive Moment Estimation (Estimation Adaptative du Moment Accélérée par Nesterov).

PRelu : Parametric Rectified Linear Unit (Unité Linéaire Rectifiée Paramétrique).

ReLU : Rectified Linear Unit (Unité Linéaire Rectifiée).

RF : Radio-Fréquences.

RMSProp : Root Mean Square Propagation (Propagation de la Racine Carrée Moyenne).

RNA : Artificial Neural Network (Réseau de Neurones Artificiels).

RSO : Le radar à synthèse d'ouverture (l'abréviation française).

SAR : Synthetic Aperture Radar (L'abréviation internationale (anglais)).

SIR : Shuttle Imaging Radar.

SIR-A : Shuttle Imaging Radar-A.

SIR-B : Shuttle Imaging Radar-B.

SIR-C : Shuttle Imaging Radar-C.

Softmax : Soft Maximum Function (fonction maximum adoucie).

SWISH : Smooth Weighted Input by Sigmoid.

TV : La variation totale.

UIT : Union Internationale des Télécommunications.

VH : Vertical-Horizontal (Onde émise verticalement et reçue horizontalement).

VV : Vertical-Vertical (Onde émise verticalement et reçue verticalement).

Table des matières

Introduction Générale.....	1
-----------------------------------	----------

Chapitre I

Fondements de l'imagerie SAR

I.1	Introduction	5
I.2	Télédétection et radar à synthèse d'ouverture.....	5
I.2.1	Radar	5
I.2.1.1	Principe du radar.....	5
I.2.1.2	Principe de détection	6
I.2.1.3	Principe de fonctionnement	7
I.2.2	Radar à Synthèse d'Ouverture	8
I.2.2.1	Historique du radar à synthèse d'ouverture.....	8
I.2.2.2	Définition du radar à synthèse d'ouverture	11
I.2.2.3	Concepts de base du Radar à Synthèse d'Ouverture	12
I.2.2.4	Géométrie d'une mesure SAR.....	14
I.2.2.5	Résolution d'un radar	15
I.3	Modes d'acquisition SAR	19
I.3.1	Mode Strip-Map.....	19
I.3.2	Mode Télescope.....	19
I.3.3	Affinage Doppler à antenne en rotation.....	20
I.3.4	Mode ScanSAR	21
I.3.5	Mode Squint (dépointé)	21
I.3.6	Mode inverse.....	22
I.3.7	Mode multivue.....	23
I.4	Radiométrie des images SAR.....	23
I.5	Équation du radar	24
I.6	Principaux domaines d'application des images SAR.....	24
I.7	Conclusion.....	25

Chapitre II

Caractérisation et méthodes de filtrage classiques

II.1 Introduction.....	27
II.2 Bruit	27
II.2.1 Bruit de speckle	28
II.2.2 Restauration et suppression des bruits	29
II.3 Modélisation statistique du Speckle.....	29
II.3.1 Distribution Gamma	29
II.3.2 Nombre de looks (L).....	30
II.3.3 Impact sur la qualité d'image.....	31
II.4 Méthodes Classiques de Débruitage	32
II.4.1 Filtres adaptatifs.....	33
II.4.2 Méthodes non locales.....	39
II.4.3 Approches variationnelles (Total Variation)	41
II.4.4 Synthèse et limites des approches traditionnelles.....	42
II.5 Conclusion	43

Chapitre III

Apprentissage profond pour la restauration d'images

III.1. Introduction	45
III.2. Fondements des réseaux de neurones artificiels.....	45
III.2.1. Perceptron	46
III.2.2. Réseaux multicouches	47
III.3. Apprentissage des réseaux de neurones.....	49
III.3.1. Apprentissage supervisé.....	49
III.3.2. Apprentissage non-supervisé	49
III.4. Apprentissage profond.....	50
III.4.1. Architectures de réseaux de neurones profonds.....	50
III.4.2. Couche de convolution (CONV).....	52
III.4.3. Couche de pooling.....	52
III.4.4. Couche entièrement connectée(FC)	53
III.4.5. Couche de Couche de perte (LOSS)	54

III.4.6. Fonction d'activation	54
III.5. Optimisation	58
III.5.1. Descente de gradient (GD).....	59
III.5.2. Descente de gradient par lots (BGD)	59
III.5.3. Descente de gradient stochastique (SGD).....	60
III.5.3. Descente de gradient stochastique (SGD).....	60
III.5.3.1. Momentum	62
III.5.3.2. Nesterov Momentum.....	62
III.5.4. Descente de gradient adaptative (AdaGrad)	62
III.5.5. Delta adaptatif (AdaDelta)	62
III.5.6. Propagation racine carrée moyenne (RMSProp).....	63
III.5.7. Estimation adaptative du moment (Adam)	63
III.5.8. Estimation du moment adaptatif maximal (AdaMax).....	63
III.5.9. Estimation du moment adaptatif accélérée par Nesterov (Nadam).....	63
III.6. Choix des hyperparamètres	64
III.6.1. Nombre de filtres.....	64
III.6.2. Forme de filtres	64
III.7. Avantages et les inconvénients Les réseaux de neurones CNNs.....	64
III.8. Conclusion	65

Chapitre IV

Résultats et Discussion

IV.1. Introduction	67
IV.2. Constitution des bases de données	67
IV.2.1. Images naturelles pour l'apprentissage.....	70
IV.2.2. Génération synthétique du bruit speckle.....	71
IV.3. Critères d'évaluation	74
IV.4. Méthode proposée	75
IV.4.1. Méthodes classiques.....	75
IV.4.1.1. Filtre de Lee	76
IV.4.1.2. Filtre de Kuan.....	77
IV.4.1.3. Filtre de Frost	78
IV.4.2. Approche basée sur les réseaux de neurones convolutifs (CNN)	79

IV.4.2.1. Conception du modèle neuronal CNN	81
IV.4.2.2. Apprentissage des modèles CNN	86
IV.4.2.2.1. Fonction de perte	86
IV.4.2.2.2. Stratégie et paramètres d'apprentissage	86
IV.4.2.3. Résultats et discussion des modèles CNN.....	89
IV.5. Comparaison entre ID-CNN et les filtres classiques.....	93
Conclusion générale.....	104
<i>Bibliographie.....</i>	<i>107</i>

Introduction générale

Introduction générale

La télédétection est devenue un outil essentiel pour l'observation et l'analyse de la surface terrestre. Elle permet de collecter des données précises sur l'environnement, les ressources naturelles et les infrastructures urbaines à partir de capteurs embarqués sur des plateformes aériennes ou satellitaires. Parmi les techniques de télédétection, l'imagerie radar à synthèse d'ouverture (Synthetic Aperture Radar, SAR) occupe une place particulière grâce à sa capacité à fournir des images haute résolution spatiale, indépendamment des conditions météorologiques et de l'éclairement solaire [1]. Cette technologie repose sur l'émission et la réception d'ondes électromagnétiques cohérentes, permettant ainsi une observation détaillée des surfaces terrestres et maritimes.

Les images SAR sont utilisées dans des domaines variés : surveillance environnementale (suivi de la déforestation, fonte des glaciers, inondations), agriculture de précision (estimation de la biomasse, humidité des sols), gestion des catastrophes naturelles (cartographie rapide des zones sinistrées), urbanisme et cartographie (détection des changements d'occupation des sols), ainsi que dans les applications de défense et sécurité [2], [3]. Leur capacité à fournir des données fiables de jour comme de nuit, quelles que soient les conditions atmosphériques, en fait un outil stratégique pour la recherche scientifique et l'aide à la décision.

Malgré leurs nombreux avantages, les images SAR sont affectées par un bruit particulier appelé *bruit speckle*. Ce phénomène, de nature multiplicative, résulte des interférences cohérentes entre les ondes radar réfléchies par les différentes structures présents dans la scène observée [4], [5]. La présence du speckle réduit la qualité visuelle des images, masque les détails fins et complique l'interprétation des données. Son atténuation constitue donc un enjeu majeur pour l'exploitation efficace des données SAR.

Afin de réduire l'impact du speckle, de nombreuses méthodes de débruitage ont été développées au cours des dernières décennies. Les filtres adaptatifs, tels que ceux de Lee, Kuan et Frost, s'appuient sur des modèles statistiques locaux pour réduire le bruit tout en préservant les caractéristiques principales de l'image. Toutefois, ils entraînent souvent un lissage excessif des structures et une perte de détails. Les approches variationnelles, notamment celles fondées sur la régularisation par variation totale, permettent une meilleure conservation des contours, mais leur efficacité dépend fortement du choix des paramètres de régularisation. Les méthodes non locales exploitent quant à elles la redondance des motifs présents dans l'image afin d'améliorer les performances de débruitage, au prix d'une complexité computationnelle plus élevée [6], [7].

L'apprentissage profond a révolutionné le traitement d'images en permettant l'apprentissage de représentations complexes directement à partir des données. Les réseaux de neurones convolutionnels (Convolutional Neural Networks, CNN) se distinguent par leur capacité à réduire efficacement le speckle tout en préservant les structures fines et les textures [8], [9], [10]. Leur adaptabilité aux différentes bases de données et leur robustesse face à la variabilité des scènes en font une alternative prometteuse aux méthodes classiques.

L'objectif principal de ce mémoire est de développer et évaluer une approche de débruitage des images SAR basée sur les réseaux de neurones convolutifs. Afin d'analyser de manière rigoureuse les performances de cette approche, deux catégories de méthodes sont étudiées.

La première catégorie regroupe les techniques classiques de débruitage, notamment les filtres de Lee, de Kuan, et de Frost. Ces méthodes, largement utilisées dans la littérature scientifique, servent de référence pour l'évaluation comparative des résultats obtenus.

La seconde catégorie est basée sur des approches d'apprentissage profond basées sur les CNN. Ces réseaux sont capables d'extraire automatiquement les caractéristiques spatiales pertinentes des images SAR et de modéliser efficacement la relation entre les images bruitées et leurs versions débruitées.

Plusieurs travaux récents ont démontré l'intérêt des CNN pour la réduction du speckle. En particulier, Chierchia et al. [11] ont proposé une architecture composée de couches convolutionnelles, appelé SAR-CNN, intégrant des blocs Conv-BN-ReLU ainsi qu'une couche résiduelle basée sur une division élément par élément pour estimer le bruit speckle. Afin de prendre en compte la nature multiplicative du bruit, cette approche adopte une méthode homomorphique fondée sur l'application successive de transformations logarithmiques et exponentielles, associée à une fonction de coût adaptée à la distribution statistique du speckle.

De leur côté, Wang et al. [12] ont développé une architecture proche de DnCNN, composée de huit blocs Conv-BN-ReLU. Dans cette méthode, la fonction de perte classique basée sur l'erreur quadratique moyenne (MSE) est remplacée par une combinaison d'une perte euclidienne et d'une perte de variation totale, permettant d'obtenir des images restaurées plus lisses tout en conservant les structures importantes.

Par ailleurs, la référence [13] a proposé une méthode de débruitage fondée sur un réseau dense résiduel (Residual Dense Network). Cette architecture prend en entrée une image polarimétrique multi-canaux et produit en sortie l'image résiduelle correspondante, permettant

ainsi une meilleure exploitation des informations de polarisation et une amélioration des performances de débruitage.

Dans le cadre de ce travail, une architecture de type Image Despeckling Convolutional Neural Network (ID-CNN), inspirée des approches proposées dans [11], [12], [13], est développée et évaluée pour le débruitage des images SAR. L'architecture proposée repose sur une cascade de couches convolutionnelles organisées selon différentes configurations afin d'extraire efficacement les caractéristiques spatiales de l'image et de modéliser le bruit speckle. De plus, à l'instar de l'approche présentée dans [13], une stratégie d'apprentissage résiduel est adoptée. Au lieu d'estimer directement l'image débruitée, le réseau apprend à prédire la composante de bruit présente dans l'image d'entrée, ce qui facilite le processus d'apprentissage et améliore les performances de restauration. L'image débruitée est ensuite reconstruite à partir de l'image bruitée et du résidu estimé par le réseau. Les différentes architectures étudiées ont été implémentées sur la plateforme Google Cloud, en exploitant les ressources de calcul GPU afin d'accélérer les phases d'apprentissage, d'optimisation et de validation des réseaux de neurones convolutifs.

Ce mémoire est organisé en quatre chapitres principaux :

- Le premier chapitre présente les principes fondamentaux de l'imagerie SAR, son fonctionnement ainsi que ses principaux domaines d'application.
- Le deuxième chapitre est consacré à l'analyse du bruit speckle et aux méthodes classiques de débruitage utilisées dans le contexte des images SAR.
- Le troisième chapitre introduit les concepts fondamentaux de l'apprentissage profond et des réseaux de neurones convolutifs appliqués à la restauration et au débruitage des images.
- Le quatrième chapitre décrit la méthodologie proposée, présente les résultats expérimentaux obtenus et réalise une comparaison approfondie avec les approches classiques Lee, Kuan, et Frost.

Enfin, une conclusion générale récapitule les principales contributions de ce travail et présente les perspectives de recherche envisageables pour améliorer davantage les performances du débruitage des images SAR.

Chapitre I

Fondements de l'imagerie SAR

I.1 Introduction

La télédétection constitue aujourd'hui un outil incontournable pour l'observation et l'analyse de la surface terrestre. Parmi les différentes technologies existantes, le radar à synthèse d'ouverture (SAR) occupe une place importante grâce à sa capacité à acquérir des images indépendamment des conditions climatiques et de l'éclairage. Contrairement aux systèmes optiques, le SAR utilise des ondes électromagnétiques, ce qui lui permet de fonctionner de jour comme de nuit et à travers les nuages.

Le principe fondamental du SAR repose sur l'émission d'un signal radar et l'analyse de l'écho réfléchi par les objets observés. Grâce à des techniques avancées de traitement du signal, il est possible de reconstruire des images à haute résolution, en exploitant le mouvement du capteur pour simuler une antenne de grande taille.

Dans ce contexte, ce chapitre vise à présenter les fondements théoriques de l'imagerie SAR. Il aborde le principe de fonctionnement du radar, la formation des images ainsi que les notions essentielles de résolution radiale et azimutale. Ces éléments sont indispensables pour comprendre les méthodes de traitement et d'amélioration des images SAR qui seront développées dans la suite de ce travail.

I.2 Télédétection et radar à synthèse d'ouverture

I.2.1 Radar

Le mot radar est une abréviation de l'anglais «Radar Detection Ranging» c'est-à-dire détection par radio et mesure de la distance. Ce mot est utilisé pour décrire un système d'acquisition d'images à des longues distances. Le principe de fonctionnement du système radar se base sur l'éclairage artificiel de la scène, et l'analyse des ondes réfléchies.

Nous distinguons deux types de systèmes radar : le radar à ouverture réelle et le radar à ouverture synthétique (Synthetic Aperture Radar ou SAR). C'est ce dernier type de radar (SAR) qui a attiré l'attention des chercheurs en télédétection ces dernières années.

Ce système produit des images de bonne qualité dans des conditions difficiles (nuages, longue distance). Cependant, les images produites sont affectées par un bruit, et nécessitent un prétraitement avant d'être interprétées [14].

I.2.1.1 Principe du radar

Le principe utilisé par le radar est similaire à celui de la réflexion des ondes sonores, à l'image d'un vol de la chauve souris. Par exemple dans le cas d'un radar à impulsion ce dernier utilise des impulsions électromagnétiques (signal hyperfréquence) qui se propage dans

l'espace en générale à ligne droite et à vitesse constante. Ces derniers sont en fonction de l'indice de réfraction qui dépend des conditions météorologiques.

Comme illustre la Figure I.1, un signal hyperfréquence est émis en direction de la cible. Une partie de cette énergie est alors réfléchiée par la cible dans la direction du radar, cette énergie renvoyée par la cible est appelée écho, est exploitée pour déterminer la direction ainsi que la distance de l'objet réflecteur [15].

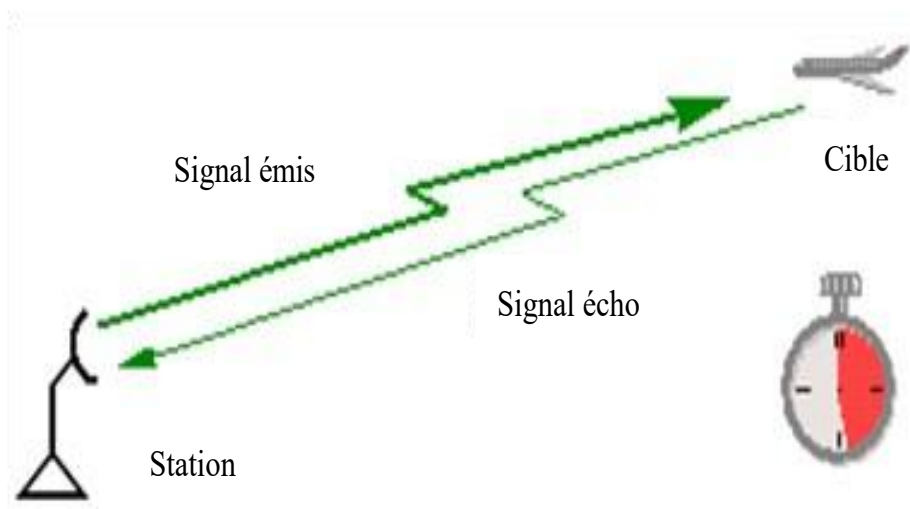


Figure I. 1. Principe du radar [15].

Les ondes électromagnétiques se propagent dans l'air approximativement à la vitesse de la lumière dans le vide : 3.10^8 m/s [15].

I.2.1.2 Principe de détection

Les systèmes radars permettent de mesurer des informations sur la distance ainsi que la vitesse des objets présents dans leur environnement. Comme illustré à la Figure I.2, un système radar monocanal comporte une antenne d'émission et une antenne de réception.

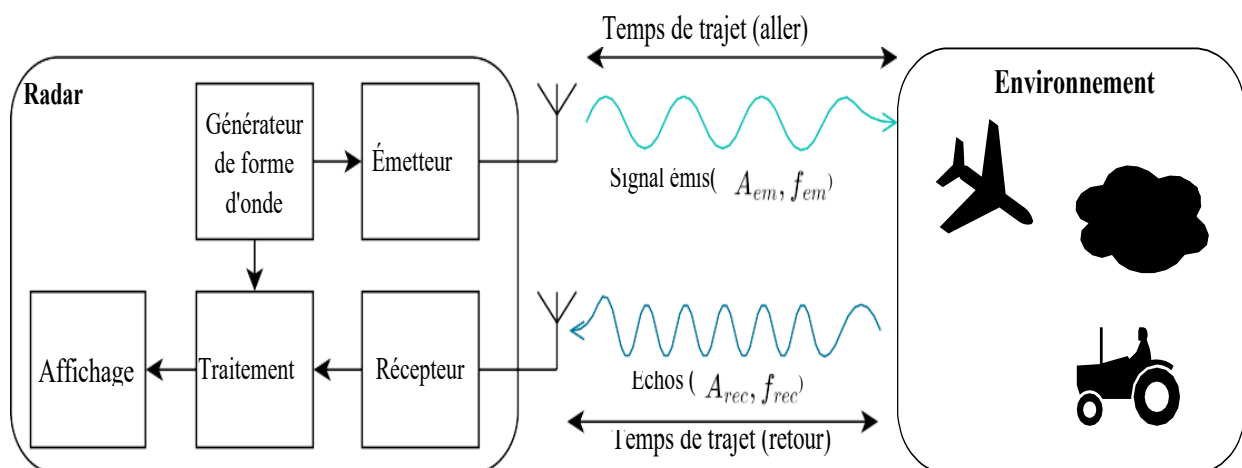


Figure I. 2. Principe de fonctionnement d'un système radar [16].

Des ondes électromagnétiques émises se réfléchissent sur les objets environnants. Comme mentionné précédemment, l'onde réfléchie est appelée écho. Le temps de propagation de l'onde, ainsi que le décalage de fréquence provoqué par l'effet Doppler-Fizeau, mesurés par le système radar, permettent de déterminer la distance et la vitesse des objets ayant réfléchi l'onde incidente. Par ailleurs, l'amplitude de l'écho subit une atténuation en fonction de l'environnement traversé et la bande de fréquence de l'onde émise [16].

I.2.1.3 Principe de fonctionnement

En dehors de son utilisation de base concernant la détection et la localisation d'un objet, l'une des exploitations phares d'un radar est l'extraction à distance des informations sur un objet (fixe ou en mouvement) observé. Pour ce faire, il se base sur le principe de l'émission et la réception des ondes électromagnétiques ou plus exactement sur les ondes Radio Fréquences (RF). Ces ondes sont émises dans la direction de la zone d'intérêt et se caractérisent par leurs longueurs d'onde (ou leurs fréquences). Les principales bandes de fréquence des radars sont données par le Tableau I.1. La bande de fréquence radar adoptée doit être conforme avec les standards IEEE (Institute of Electrical and Electronics Engineers). Tandis que la fréquence centrale et la largeur de bande du signal sont soumises aux réglementations de l'UIT (Union Internationale des Télécommunications). Lorsque les ondes électromagnétiques rencontrent un objet distant, elles sont réfléchies en direction du radar. Le signal réfléchi est généralement porteur d'informations sur l'objet illuminé. Les informations généralement extraites incluent la distance, la vitesse, la position ou autres caractéristiques. Et de plus en plus les systèmes radar d'aujourd'hui sont exploités dans divers secteurs pratiques, incluant le domaine automobile, la reconnaissance de cibles mais aussi des applications en télédétection [17].

Tableau I. 1. Bandes de fréquence des ondes électromagnétiques [17].

Bandes	Longueur d'ondes (cm)	Fréquences (GHz)
Ka	0.75 à 1.1	26.5 à 40
K	1.1 à 1.67	18 à 26.5
Ku	1.67 à 2.4	12.5 à 18
X	2.4 à 3.75	8 à 12.5
C	3.75 à 7.5	4 à 8
S	7.5 à 15	2 à 4
L	15 à 30	1 à 2
P	30 à 100	0.3 à 1

D'une manière simplifiée, un radar est composé des éléments suivants :

- a) **Émetteur:** Il génère les impulsions hyperfréquences à la fréquence et la puissance désirées par l'utilisateur. Ces impulsions sont destinées à être diffusées par l'antenne. Le type de l'émetteur peut être par exemple un oscillateur de puissance.
- b) **Antenne :** C'est la composante la plus visible du radar. Dans l'émission, l'antenne concentre l'énergie émise par le radar dans une direction déterminée. Lors de la réception, l'antenne intercepte l'énergie réfléchi par les objets aux alentours et la transmet au récepteur. Dans le cas d'une configuration monostatique, cette antenne est commune pour l'émission et la réception.
- c) **Modulateur :** C'est un circuit électronique qui permet d'assurer le stockage de l'énergie pendant les périodes séparant des émissions successives et sa restitution ensuite pendant le temps très bref de l'émission radar.
- d) **Récepteur :** Il est le cœur de la chaîne du traitement radar vu qu'il fournit les données nécessaires à exploiter pour extraire les informations nécessaires concernant la cible. Son rôle principal est l'amplification du signal. Cela doit être effectué sans déformation du signal initial. C'est pour cela que d'autres tâches sont associées à ce dispositif à l'instar du filtrage. Le récepteur doit veiller à maximiser le rapport signal/bruit du signal réfléchi afin de faciliter son traitement dans la suite de la chaîne.
- e) **Duplexeur :** Il est utilisé principalement dans la configuration mono-statique. Il est le responsable de l'utilisation d'une seule antenne pour l'émission et la réception. Pour ce faire, il transmet le signal d'émission à l'antenne et le signal capté au récepteur.
- f) **Synchronisateur:** En se basant sur une horloge d'une très grande stabilité, il génère des signaux qui définissent les instants d'émission. Il permet aussi de délivrer des signaux nécessaires à des opérations en temps réel.
- g) **Traitement et exploitation des signaux :** En traitant le signal brut capté par l'antenne de réception, cette étape vise à fournir des signaux exploitables pour différentes applications (détection, reconnaissance, ...).

Dans l'application de reconnaissance visée dans ce travail, et après différents traitements de base, les signaux exploitables sont donnés sous forme d'images radar [17].

I.2.2 Radar à Synthèse d'Ouverture

I.2.2.1 Historique du radar à synthèse d'ouverture

La première expérience réussie de transmission d'hyperfréquences et de réflexion sur différents objets fut réalisée par Heinrich Hertz en 1886. Cette découverte a constitué une étape fondamentale dans le développement des technologies radar. Au début du 20^{ème} siècle, les premiers radars rudimentaires ont été conçus principalement pour la détection des navires.

Durant les années 1920 à 1930, des radars au sol à impulsions expérimentaux ont été développés pour détecter des objets éloignés. Ces systèmes étaient essentiellement destinés à la localisation et au suivi des avions et des navires. Pendant la seconde guerre mondiale, les besoins militaires en reconnaissance à distance et en observation tous temps ont accéléré l'évolution des systèmes radar. Les premiers radars de cartographie aéroportés firent alors leur apparition. Toutefois, la qualité des images obtenues restait limitée en raison de la faible taille des antennes installées sur les avions.

Pour améliorer la résolution des images, les chercheurs britanniques et américains ont développé, au début des années 1950, des antennes plus larges, installées suivant l'axe longitudinal de l'avion. Cette innovation a conduit à la première génération des radars à vision latérale (SLAR, Side Looking Airborne Radar), largement utilisée durant les années 1960 pour l'observation et la cartographie terrestre.

Parallèlement, des travaux théoriques menés aux États-Unis en 1951 ont établi les principes d'une nouvelle génération de radars : les radars à synthèse d'ouverture (RSO), ou SAR (Synthetic Aperture Radar). Contrairement aux radars conventionnels, ces systèmes permettent d'obtenir une haute résolution spatiale malgré la taille réduite de leur antenne, grâce à la synthèse numérique d'une ouverture équivalente beaucoup plus grande. Les premiers prototypes opérationnels furent développés aux États-Unis à partir de 1958, tandis que des essais en vol furent réalisés en Angleterre et en France durant les années 1970.

L'application spatiale du radar à synthèse d'ouverture a débuté à la fin des années 1970 avec le lancement du satellite SEASAT-A, le 28 juin 1978 (Figure I.3-(a)). Il s'agissait du premier satellite civil embarquant un radar SAR destiné à l'observation de la Terre. Bien que sa mission ait été interrompue en octobre de la même année à la suite d'un court-circuit, SEASAT a démontré l'intérêt majeur de cette technologie pour l'océanographie et la télédétection active [18], [19].

Dans les années 1980, plusieurs programmes spatiaux ont poursuivi le développement de cette technologie. En 1982, la navette spatiale américaine Columbia embarqua le Shuttle Imaging Radar (SIR-A), qui permit d'acquérir des images de nombreuses régions du globe. Cette mission fut suivie par SIR-B, puis par SIR-C/X-SAR, lancé finalement en 1994 à bord de la navette Endeavour, après le report de la mission initialement prévue en 1988 (Figures I.3-(b) et I.3-(c)) [19].

Un tournant important vers les systèmes radar spatiaux opérationnels fut marqué par le lancement du satellite ERS-1 par l'Agence Spatiale Européenne (ESA) le 16 juillet 1991. Équipé d'un radar SAR offrant une résolution d'environ 30 mètres, ce satellite était également

doté d'instruments complémentaires destinés à l'étude globale des océans, des vents, des vagues et de la température de surface [19].

Peu après, d'autres satellites ont renforcé l'intérêt mondial pour les systèmes SAR, notamment JERS-1 (Japon, 1992), ERS-2 (Europe, 1995), RADARSAT-1 (Canada, 1995) et ENVISAT (ESA, 2002). Ces plateformes ont considérablement amélioré les capacités d'acquisition d'images radar, avec des résolutions variées et des modes d'observation multifréquences et multipolarisations [18], [19].

Parallèlement, plusieurs systèmes radar aéroportés ont été développés, tels que E-SAR (DLR, Allemagne), AIRSAR (JPL, États-Unis), RAMSES (ONERA, France), PISAR (Japon) et CONVAIR (Canada). De même, les systèmes spatiaux comme XSAR, ALOS-PALSAR, RADARSAT-2 et PALSAR-2 (2013) ont contribué à l'évolution des techniques de télédétection radar [18], [20].

Les satellites ERS-1/2, RADARSAT et ENVISAT portent un SAR. Cette ouverture synthétique est une technique qui permet d'augmenter la résolution dans une des directions du plan perpendiculaire au rayonnement du faisceau. L'amélioration de la résolution azimutale est obtenue à partir de l'intégration cohérente de l'énergie réfléchie par la cible. Le terme ouverture synthétique fait référence à la distance que le radar parcourt pendant le temps que la cible reste éclairée par le lobe de l'antenne émettrice [19].

Aujourd'hui, les images SAR complètent efficacement les données optiques issues de satellites comme SPOT et Landsat, Sentinel-2, en permettant l'observation de la Terre indépendamment des conditions météorologiques et de l'éclairement. Grâce à leur capacité de pénétration et à leur haute résolution, elles sont devenues essentielles dans de nombreux domaines, tels que la cartographie, la surveillance environnementale, l'océanographie, la gestion des catastrophes naturelles, l'agriculture de précision ainsi que les applications militaires.

Nombreux satellites fournissent activement des images SAR et optiques pour la recherche académique. L'un d'eux est Sentinel. Alors que le satellite Sentinel-1, basé sur la technologie radar, acquiert des images SAR, le satellite Sentinel-2 repose sur l'imagerie optique et capture des images multispectrales. Les images issues des satellites Sentinel sont utilisées dans diverses applications, telles que l'agriculture [21] et l'évaluation des catastrophes naturelles [22].

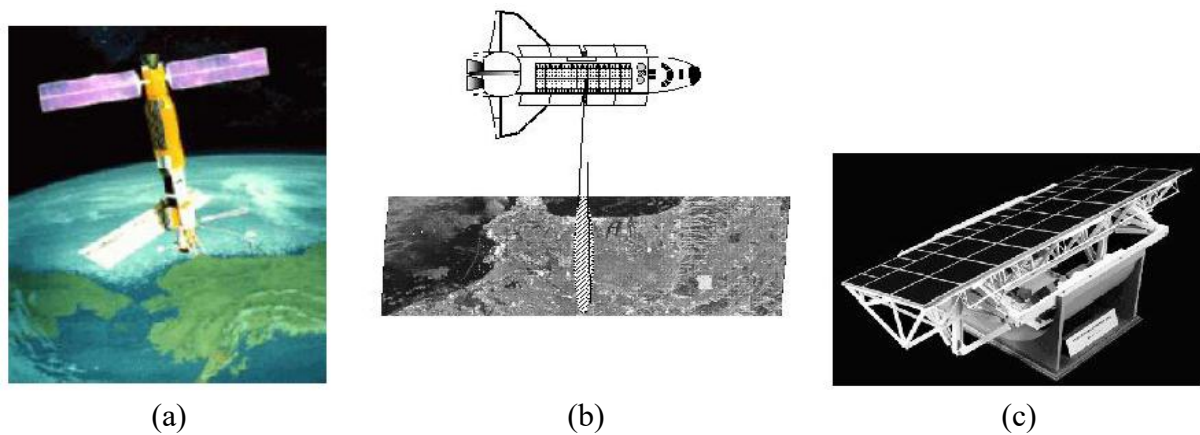


Figure I. 3. (a) Radar SEASAT RSO, (b) Le SIR-C/X-RSO illuminant une région de la terre, (c) Modèle de l'antenne SIR-C/X-RSO.

I.2.2.2 Définition du radar à synthèse d'ouverture

Le Radar à Synthèse d'Ouverture (RSO), appelé Synthetic Aperture Radar (SAR) en anglais, est un système radar permettant d'améliorer grandement la résolution en azimuth par rapport à un système radar imageur classique. Les premiers systèmes SAR ont été développés au début des années 50 et les premières images acquises à la fin de cette décennie. La Figure I.4 montre la première image SAR acquise par l'université du Michigan en 1957, grâce à des lampes à mercures. Ce n'est qu'avec l'avènement du laser et des avancées en informatique et en électronique que la qualité des images s'est améliorée. Depuis le lancement de SEASAR, le premier SAR numérique embarqué sur un satellite, en 1978, les systèmes SAR ont permis d'apporter une quantité importante d'information pour l'étude de la Terre [23].

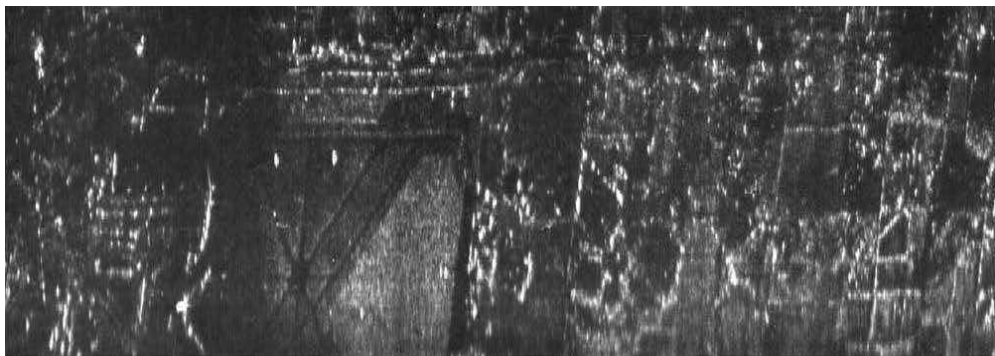


Figure I. 4. Première image SAR acquise en 1957 [23].

Un radar imageur fonctionne en émettant des ondes électromagnétiques dans une direction donnée, généralement vers le sol. La surface illuminée diffuse l'onde incidente dans toutes les directions, dont la direction d'émission. La partie qui est réémise vers le radar est enregistrée et traitée afin de former l'image radar. On appelle cette partie le signal rétrodiffusé. La Figure I.5 présente, de manière schématisée, le fonctionnement d'un système SAR. Un porteur avion ou satellite se déplace selon une trajectoire donnée et illumine une zone du sol [23].

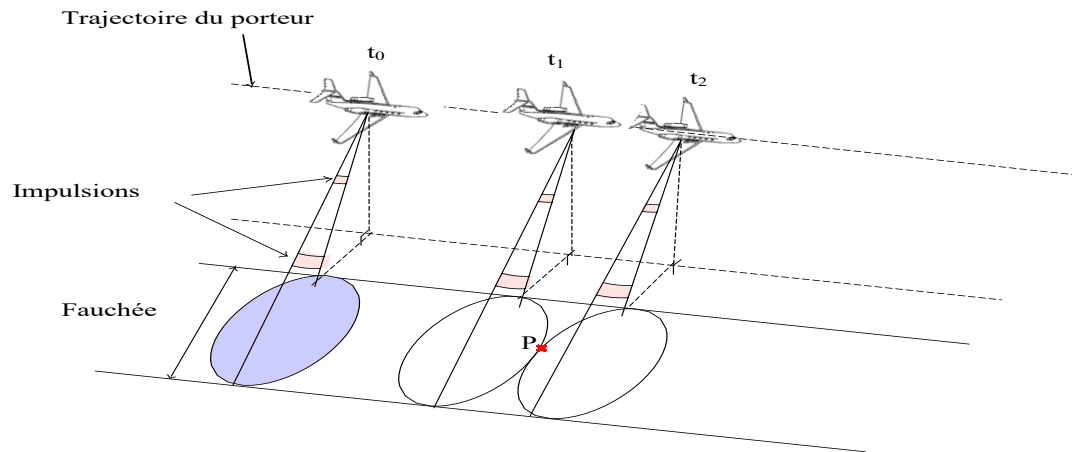


Figure I. 5. Schéma d'un système SAR [23].

La résolution spatiale est un point très important de tout système imageur. C'est la capacité d'un radar à séparer deux objets voisins selon une certaine distance. Si la résolution est suffisamment grande, les deux objets voisins seront placés dans deux cases de résolutions séparées. Sinon, ils seront situés dans la même case par combinaison de l'énergie réfléchie par les deux objets [23].

Cependant, la résolution est directement liée à la durée des impulsions comprimées émises par le radar : pour augmenter la résolution, il faut réduire la durée des impulsions ou augmenter la bande émise du signal si les impulsions sont codées. Il est aussi nécessaire que ces impulsions aient une énergie élevée afin de minimiser les effets de bruit induits par les rayonnements naturels mais surtout les bruits de mesure inhérents au système radar et de pouvoir détecter les signaux émis : on cherche à maximiser le rapport signal sur bruit. La génération d'impulsions courtes à grande énergie étant trop complexe à réaliser en pratique, les impulsions utilisées sont plus longues et codées mais ont une énergie suffisamment grande [23].

I.2.2.3 Concepts de base du Radar à Synthèse d'Ouverture

Il existe différents types de système SAR : spatial si le capteur est embarqué sur un satellite ou navette spatiale, ou aéroporté si le porteur est un avion. Un système SAR illumine une scène de la façon présentée par la Figure I.6. Le capteur se déplace le long d'un trajet rectiligne, appelé azimuth, à une altitude H , au-dessus d'un plan de référence (x, y) . Comme le montre la Figure I.5, la trace du faisceau au sol est nommée l'empreinte et la zone décrite par le faisceau en se déplaçant est nommée la fauchée. La direction de l'axe de visée du radar est appelée direction radiale ou distance [24].

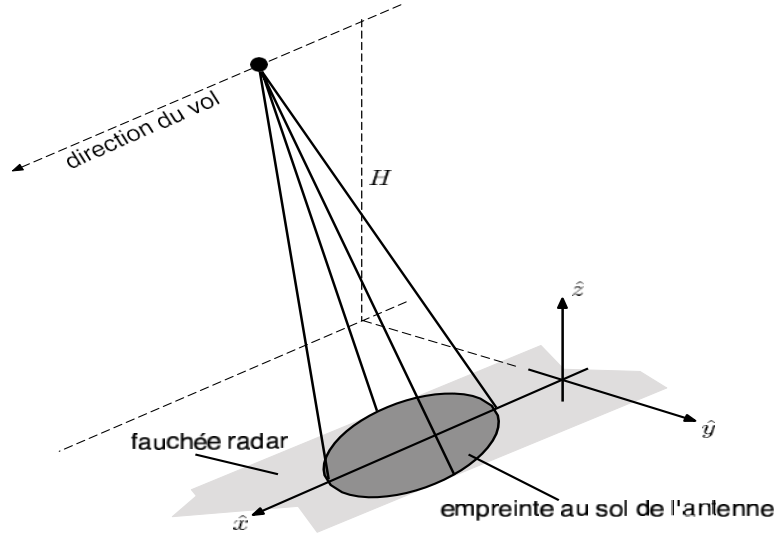


Figure I. 6. Illumination d'une scène par un SAR [24].

La résolution en distance est définie comme la distance minimale de séparation de deux points pouvant être distingués par le radar dans la direction distance. Ainsi, la résolution en distance δ_r , dépend de la durée du pulse d'émission τ_p , ou inversement de la largeur de bande du signal W :

$$\delta_r = \frac{c\tau_p}{2} = \frac{c}{2W} \quad (\text{I.1})$$

où c représente la célérité de la lumière dans le vide [24].

La résolution en azimuth dépend de la taille de l'antenne dans la direction azimuthale, D_a ainsi que de l'angle d'ouverture de l'antenne θ_{ap} reliés par la relation :

$$\theta_{ap} \propto \frac{\lambda}{D_a} \quad (\text{I.2})$$

où λ est la longueur d'onde. Ainsi, la résolution en azimuth δ_a , devient :

$$\delta_a = r_0 \frac{\lambda}{D_a} \quad (\text{I.3})$$

où r_0 est la distance radiale entre l'antenne et la cible.

La résolution dépend de la taille de l'antenne. Ainsi comme physiquement la taille d'une antenne ne peut être infinie, la résolution en azimuth est relativement médiocre. Elle peut être améliorée en employant le concept de la synthèse d'ouverture. Le principe SAR est basé sur l'utilisation d'une plateforme en mouvement afin de synthétiser une antenne effective plus longue et améliorer ainsi la résolution en azimuth. Ceci est réalisé en effectuant plusieurs mesures de la réflectivité d'une scène à différentes positions azimuthales, permettant ainsi,

grâce à une technique de compression des données, d'obtenir une résolution plus fine en azimut [24].

Comme pour un radar à ouverture réelle, le faisceau de l'antenne synthétique entraîne une ouverture θ_{sa} , avec une longueur d'antenne synthétique L_{sa} :

$$\theta_{sa} = \frac{\lambda}{2L_{sa}} \quad (\text{I.4})$$

Le facteur 2 prend en compte le trajet aller-retour entre le radar et la cible. La résolution en azimut devient donc, en tenant compte des nouveaux paramètres du SAR :

$$\delta_a = r_0 \frac{\lambda}{2L_{sa}} \quad (\text{I.5})$$

La longueur maximale pour l'antenne synthétique est limitée par le temps d'illumination de la cible, donnée par la taille de l'empreinte au sol, ainsi [24] :

$$L_{sa} \leq \frac{\lambda r_0}{D_a} \quad (\text{I.6})$$

En conséquence, la limite inférieure de la résolution qu'il est possible d'obtenir en utilisant un système SAR devient [24] :

$$\delta_a \geq \frac{D_a}{2} \quad (\text{I.7})$$

I.2.2.4 Géométrie d'une mesure SAR

La géométrie d'un système d'acquisition de données SAR est présentée par la Figure I.7.

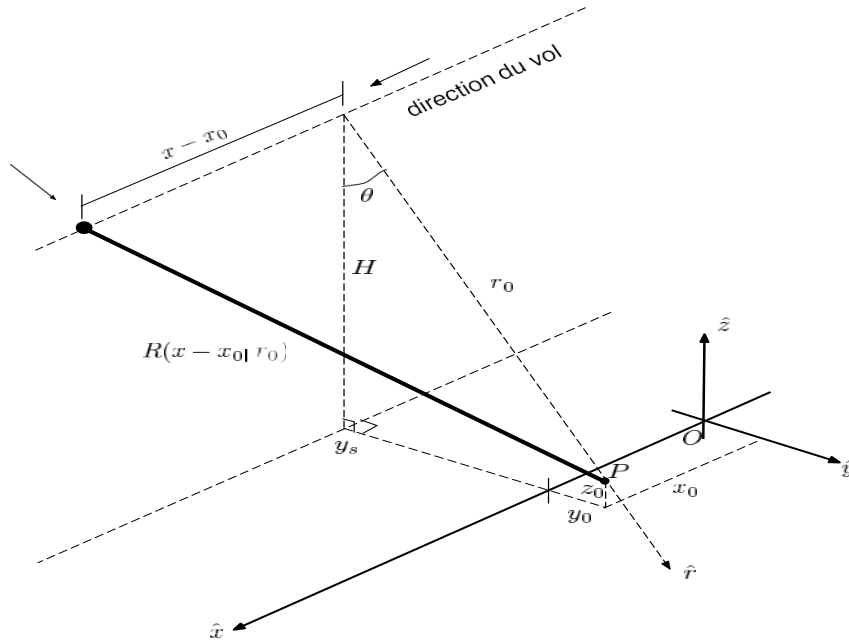


Figure I. 7. Géométrie d'une mesure SAR [24].

Le capteur se déplace suivant la direction $\hat{i}$. P représente une cible ponctuelle située à la position (x_0, y_0, z_0) par rapport à l'origine des coordonnées. La position du capteur est donnée par les coordonnées (x, y_s, H) , où $x = vt$ représente la position suivant la direction azimutale, v étant la vitesse du porteur et t le temps. y_s représente la position du porteur suivant $\hat{y}$ et H , l'altitude du porteur. θ est l'angle d'incidence du signal émis lorsque le radar se trouve à la même position azimutale que la cible [24].

Enfin $R(x - x_0; r_0)$ est la distance entre la cible et le radar et s'écrit:

$$R(x - x_0; r_0) = \sqrt{r_0^2 + (x - x_0)^2} \quad (\text{I.8})$$

Les données SAR sont représentées dans un espace à deux dimensions définies par la direction azimutale et la direction radiale (x, r) . En utilisant cette représentation, un diffuseur ponctuel est défini par son amplitude complexe $\sigma_s(x_0, r_0)$, à la position (x_0, r_0) :

$$\sigma_s(x_0, r_0) = e^{j\phi_{obj}} \sqrt{\sigma} \delta(x - x_0, r - r_0) \quad (\text{I.9})$$

où $\delta(x, r)$ est la distribution de Dirac bidimensionnelle, σ représente la section efficace radar (SER) et ϕ_{obj} le déphasage subi par l'onde lors de la réflexion sur l'objet [24].

I.2.2.5 Résolution d'un radar

La résolution d'un radar dépend principalement de deux paramètres : la résolution radiale et la résolution azimutale.

La résolution radiale d'un système radar à impulsions est limitée par la largeur de la bande de l'impulsion émise (plus la largeur de bande est ample, meilleure est la résolution radiale). Bien qu'une grande largeur de bande puisse être obtenue par une impulsion de courte durée, plus l'impulsion est courte, plus le rapport signal sur bruit est pauvre, de même pour la résolution radiométrique. Afin de préserver la résolution radiométrique, une impulsion de longue durée est souhaitable, et cela est obtenu par codage de phase de l'impulsion émise avec des fréquences d'impulsions linéaires (une modulation de fréquences linéaires).

La résolution radiale peut être déterminée à l'aide du temps de voyage de l'impulsion, c'est-à-dire, la résolution radiale correspondant à la distance minimale entre deux points de la surface de la Terre qui sont séparables [19].

Si deux points sont séparés par une distance X_r , leurs échos respectifs seront séparés par une différence de temps :

$$\Delta t = \frac{2X_r}{c} \sin \theta \quad (\text{I.10})$$

Où (c) est la vitesse de la lumière. En général, la largeur de bande de l'impulsion est définie comme $B = \frac{1}{T}$, où T est la longueur de l'impulsion. Ainsi, la résolution radiale peut être donnée par :

$$\tau = 2 \frac{X_r}{c} \sin \theta \quad (\text{I.11})$$

$$X_r = \frac{c\tau}{2 \sin \theta} = \frac{c}{2B \sin \theta} \quad (\text{I.12})$$

Ainsi, un signal de largeur de bande $B = 15.55 \text{ MHz}$ (ERS-1/2 RSO) fournira une résolution radiale égale à 24.7 m pour un angle d'incidence de balayage central θ égale à 23° [19].

Le deuxième paramètre est la résolution azimutale. Le radar à synthèse d'ouverture (SAR) exploite le mouvement du satellite afin de synthétiser une antenne virtuelle de grande longueur, ce qui permet d'améliorer considérablement la résolution azimutale. Cette résolution est obtenue en enregistrant la phase ainsi que l'amplitude des échos tout au long de la trajectoire de vol. Ainsi, un point situé à la surface de la Terre est illuminé pendant une durée finie T , comme illustré à la Figure I.8.

$$T = \frac{L}{v} \quad (\text{I.13})$$

où v est la vitesse de la plateforme et L est la dimension de l'empreinte du faisceau radar.

L'ouverture du faisceau de rayonnement azimutal est $\phi = \frac{\lambda}{D}$, alors nous pouvons écrire :

$$L = vT = R \frac{\lambda}{D} = R\phi \quad (\text{I.14})$$

Pour un faisceau de rayon étroit. Ici, R est la distance de l'antenne à la surface de la Terre (la distance oblique, "slant range"), D est la longueur de l'ouverture et λ la longueur d'onde du radar [19].

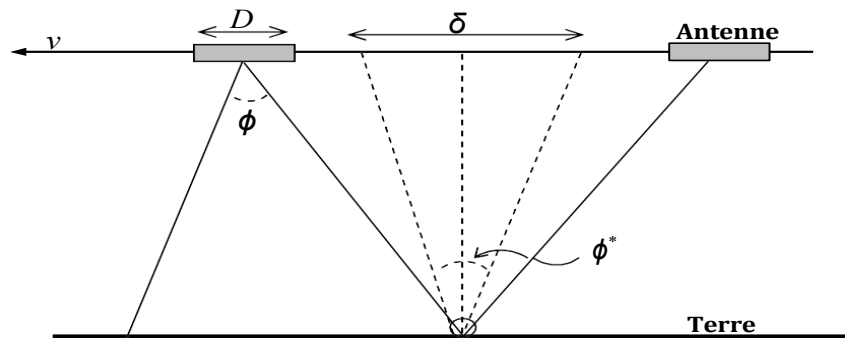


Figure I. 8. Idéalisation de la trajectoire de section linéaire du ERS-1/2 RSO [19].

Afin de calculer la résolution azimutale, on peut par simplification, considérer la cible comme une antenne équivalente. La cible reste illuminée pendant un temps T . À cet effet, nous introduit une antenne de longueur L , ce qui permet la résolution liée à l'ouverture du faisceau azimutal comme suit:

$$\phi^* = R\phi^* = R \frac{D}{R} \Rightarrow \delta = D \quad (\text{I.15})$$

Afin de garder un minimum de $3\text{dB}(50\%)$ comme la limite la plus petite dans des impulsions émises et reçues, la résolution azimutale efficace s'approche de $\delta \cong \frac{D}{2}$, comme illustré à la Figure I.9.

La résolution azimutale est inversement proportionnelle à la longueur de l'ouverture réelle, et cela est indépendant de la distance Terre-plateforme et de la longueur d'onde radar, ce qui est le grand avantage de la méthode SAR [19].

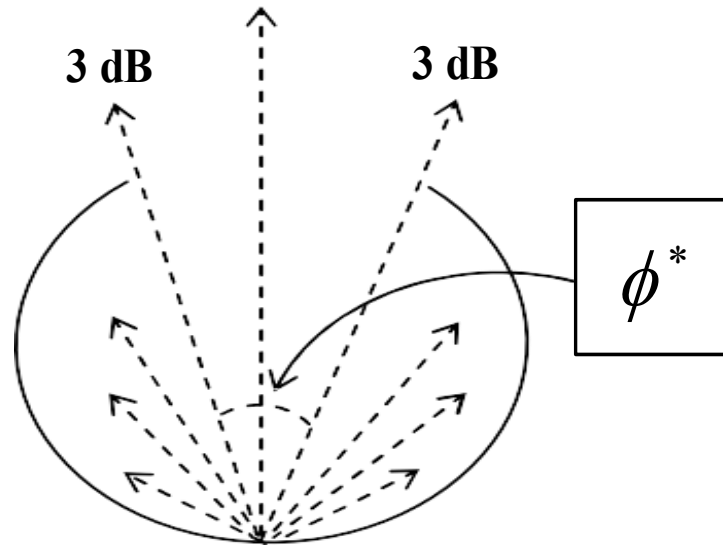


Figure I. 9. Caractéristique de la diffusion de l'onde RSO.

La synthèse d'ouverture permet d'obtenir de très bonnes résolutions, tant en distance qu'en azimut, tout en s'affranchissant des limites physiques qui s'appliquent au radar à ouverture (antenne) réelle. Les résolutions obtenues sont de l'ordre du mètre [25].

En général, un système radar imageur de télédétection est utilisé pour donner une image d'une portion du globe terrestre que l'on souhaite observer. L'angle défini par la direction de propagation de l'onde et la normale au sol au centre du lobe d'antenne (appelé angle d'incidence) conduit à des déformations géométriques. Ces effets sont illustrés dans les Figures I.10 et I.11.

- **Effet de repliement (basculement):** la pente locale du relief devient plus grande que l'angle d'incidence et le relief est imagé en sens inverse (le point f est situé entre les points d et e dans l'image radar).
- **Effet de compression ou dilatation :** selon que la pente locale est orientée vers le radar ou non, les pixels $\overline{cd}$ ou $\overline{gh}$ de l'image radar ont une dimension réduite ou dilatée par rapport à leurs projections au sol $\overline{cd}$ ou $\overline{gh}$
- **Effet d'ombre** la partie du sol $\overline{HI}$ n'est pas illuminée par le radar à cause des fortes variations locales d'altitude. Elle donne des pixels de très faible intensité (bruit seul) dans l'image radar $\overline{hi}$ [26].

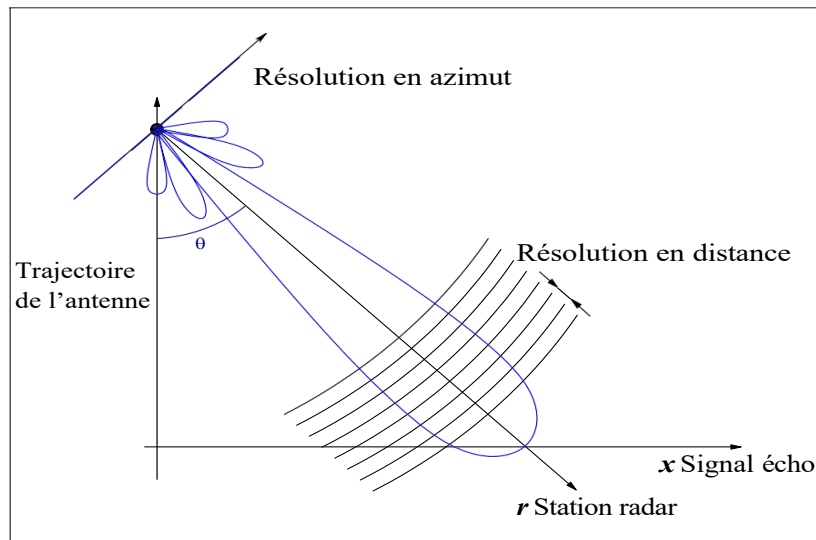


Figure I. 10. Echantillonnage d'une image SAR : résolution en distance et résolution en azimuth [26].

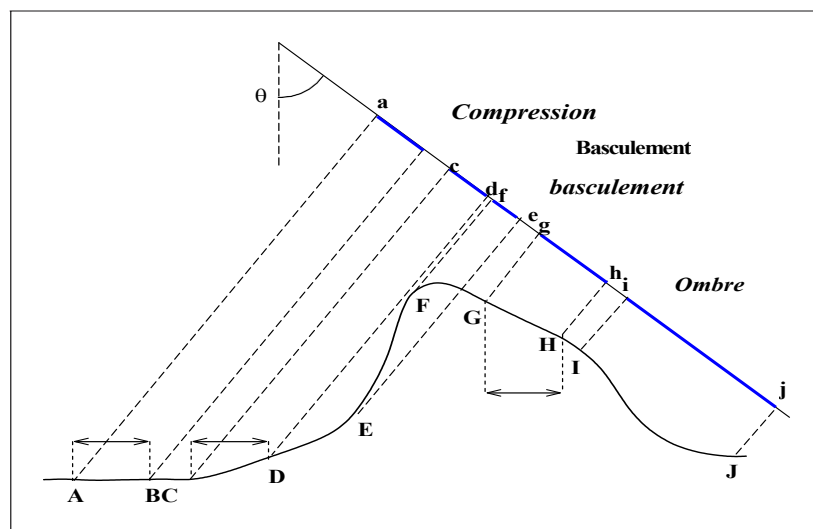


Figure I. 11. Déformations géométriques induites par le radar imageur à visée latérale [26].

I.3 Modes d'acquisition SAR

Cette partie présente les modes d'acquisition les plus courants. En fonction des caractéristiques de chacun d'entre eux, elle nous permettra de choisir le mieux adapté pour une utilisation en configuration bistatique [27].

I.3.1 Mode Strip-Map

Le mode Strip-Map, comme illustré à la Figure I.12, correspond aux acquisitions obtenues en configuration SLAR (Side Looking Airborne Radar). Dans ce mode, l'antenne reste fixe et est orientée perpendiculairement à la direction de déplacement du porteur. L'image radar est alors acquise de manière continue le long de la trajectoire de vol.

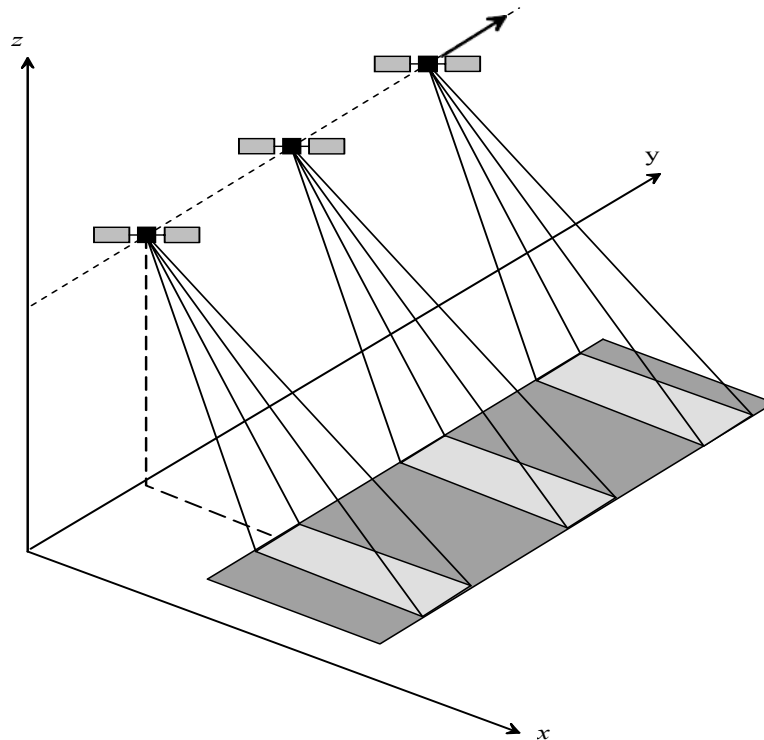


Figure I. 12. Acquisition en mode strip-map [27].

I.3.2 Mode Télescope

Le Télescope (en anglais spotbeam ou spotlight) est utilisé par les radars aéroportés et spatiaux. Son principe consiste à faire varier l'angle de visée du faisceau de l'antenne lors du déplacement du radar (Figure I.13). Ainsi, une même région est illuminée plus longtemps qu'avec une synthèse d'ouverture classique. Cette technique est envisageable sur les systèmes capables de dépointer leur antenne afin de suivre une zone au sol.

Le temps d'éclairement, et donc la taille de l'antenne synthétique, peut être augmenté jusqu'à ce que la résolution recherchée soit obtenue. L'image ne défile pas en continu, elle n'est disponible qu'à la fin de l'éclairement.

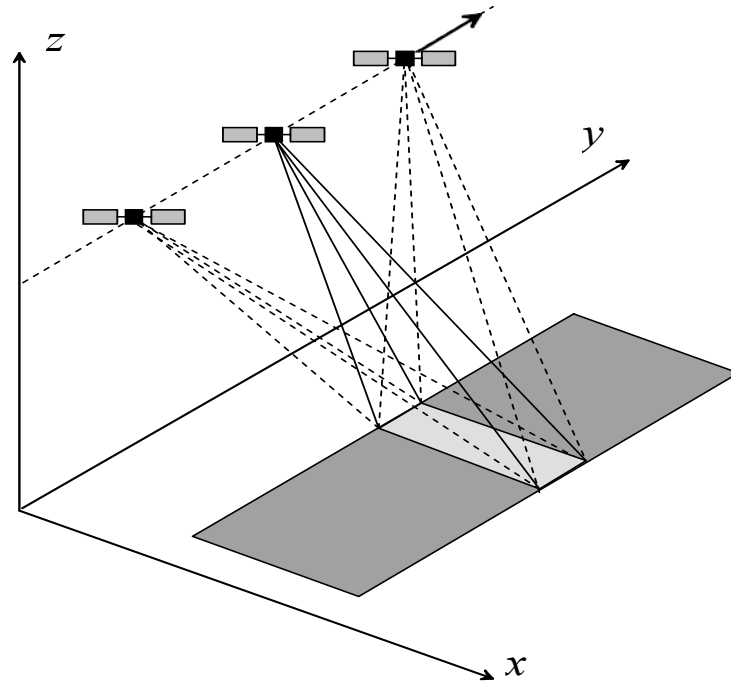


Figure I. 13. Acquisition en mode Télescope [27].

I.3.3 Affinage Doppler à antenne en rotation

L'affinage Doppler (DBS: Doppler Beam Sharpening) est une technique utilisée par les radars aéroportés pour acquérir des images du sol autour de l'avion, comme illustré à la Figure I.14. Dans ce mode, l'antenne effectue une rotation en gisement avec une vitesse angulaire w , ce qui permet d'améliorer la résolution angulaire et d'affiner l'image radar obtenue.

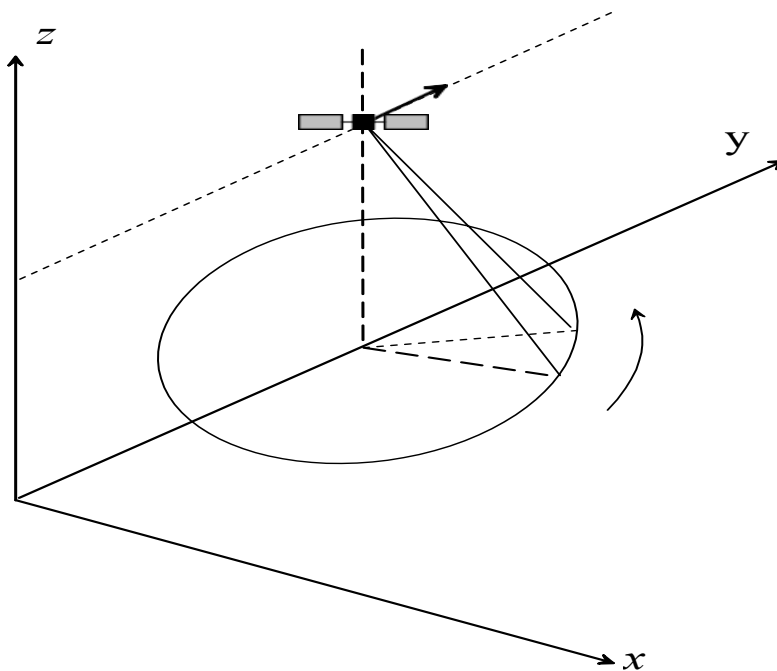


Figure I. 14. Acquisition par affinage Doppler [27].

I.3.4 Mode ScanSAR

Le mode ScanSAR est utilisé par les radars SAR afin d'obtenir une fauchée d'observation plus large que celle d'un mode SAR classique, au prix d'une dégradation de la résolution spatiale (voir Figure I.15). Cette technique consiste à utiliser les propriétés de dépointage latéral de l'antenne. Le temps d'éclairement naturel est partagé en n segments. Chaque segment est consacré à l'observation d'une fauchée différente.

Les fauchées sont choisies adjacentes. Le nombre de segments est ajusté pour atteindre la fauchée totale recherchée. La mise en œuvre de ce procédé nécessite une commutation rapide du faisceau d'antenne en élévation qui n'est réalisable qu'avec une antenne à balayage électronique.

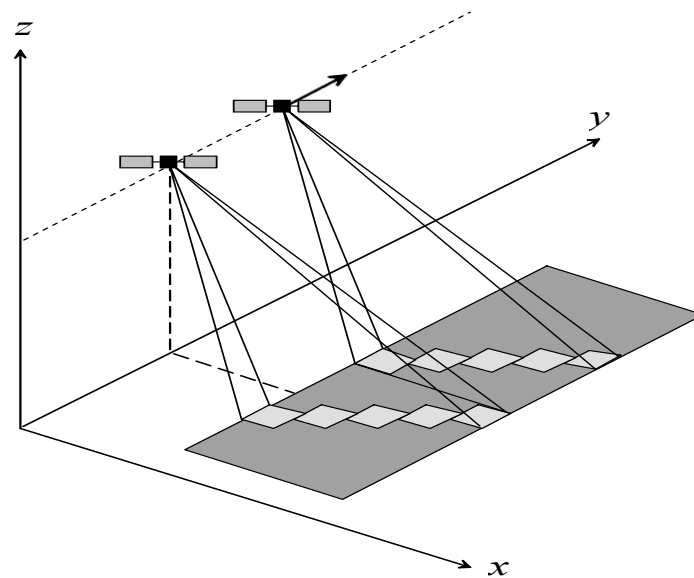


Figure I. 15. Acquisition en mode ScanSAR [27].

I.3.5 Mode Squint (dépointé)

Le mode Squint, également appelé mode dépointé, est similaire au mode Strip-Map. Toutefois, dans ce cas, l'antenne est maintenue fixe dans une direction non perpendiculaire au vecteur vitesse du porteur (Figure I.16). L'image radar est alors formée en défilement continu le long de la trajectoire de déplacement.

Les résolutions obtenues sont identiques au mode Strip-Map. L'antenne latérale reste toutefois une solution préférentielle car elle minimise le temps d'éclairement. Elle est également la plus simple du point de vue du traitement car c'est celle pour laquelle les phénomènes de migration sont les plus faibles c'est-à-dire que les variations de distances entre le radar et la cible sont faibles.

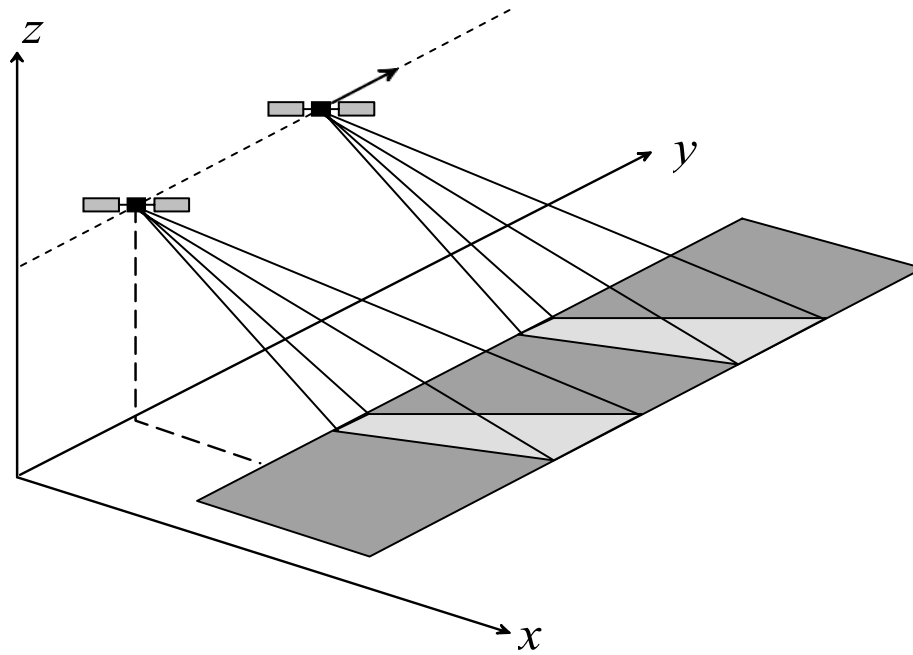


Figure I. 16. Acquisition en mode Squint [27].

I.3.6 Mode inverse

Le radar à synthèse d'ouverture inverse (ISAR- Inverse Synthetic Aperture Radar) permet de produire des images d'objets en mouvement, notamment ceux présentant une rotation relative par rapport au radar (Figure I.17). Son principe repose sur l'analyse du signal reçu en fonction du temps et de la fréquence Doppler, afin de reconstruire une image à deux dimensions de la cible.

L'analyse temporelle permet de déterminer la position des points brillants selon l'axe de la ligne de visée du radar, tandis que l'analyse de la fréquence Doppler fournit leur localisation suivant l'axe azimutal.

Dans le cas des radars aéroportés, cette technique est principalement utilisée pour la reconnaissance et l'identification d'objets mobiles, tels que les avions en vol ou de navires en mer. Etant donné que le radar et la cible étant simultanément en mouvement, le traitement du signal ne correspond pas exactement à un mode ISAR classique, mais plutôt une sorte de SAR généralisé. Le récepteur adapté au signal reçu est difficile à réaliser. En effet, les cibles n'étant pas coopératives, la géométrie du système n'est pas maîtrisée. De plus, les caractéristiques polarimétriques de la cible et du milieu sont difficilement prédictibles [27].

Lorsque le radar est fixe et que la cible est en mouvement rotatif, alors cette configuration est équivalente à celle pour laquelle la cible est fixe et le radar en mouvement suivant une trajectoire circulaire. Ainsi, on remarque que le mode inverse est un cas particulier du mode télescope pour lequel la trajectoire est circulaire [27].

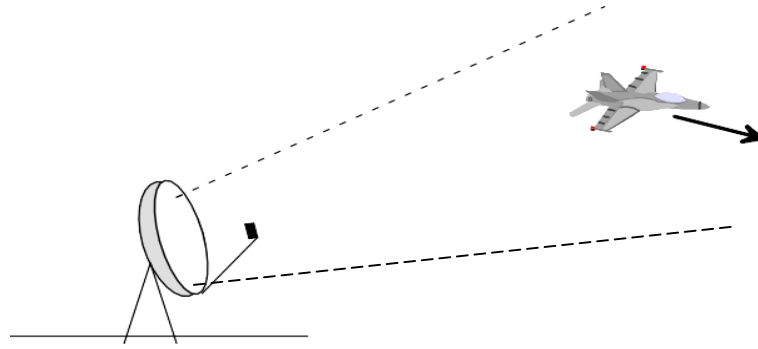


Figure I. 17. Acquisition en mode inverse [27].

I.3.7 Mode multivue

Le mode multivue est une variante applicable à tous les systèmes radar reposant sur le principe de l'antenne synthétique. Il consiste à former plusieurs images d'un même site à partir d'observations réalisées sous différents angles de vue.

Ces images, statistiquement indépendantes du fait de la fluctuation des cibles en fonction de l'angle de vue, sont ensuite sommées en module, produisant ainsi un effet de post-intégration non-cohérente qui réduit la granulation apparente (speckle). Ce mode ne diffère du mode de base dont il est dérivé que par le traitement. Les modes multi-vues sont largement utilisées pour améliorer la résolution radiométrique. Ils sont applicables aussi bien au cas de l'antenne synthétique focalisée qu'à celui de l'antenne synthétique non focalisée [27].

I.4 Radiométrie des images SAR

Les images SAR présentent un aspect granuleux, qualifié de bruit multiplicatif, dû à l'aspect cohérent de l'onde radar émise qu'on appelle communément le chatolement (speckle). Même pour des zones aux caractéristiques physiques stationnaires, c'est-à-dire homogènes, l'image radar apparaît comme étant composée de pixels dont les valeurs sont extrêmement dispersées, plus particulièrement pour les images à faible nombre de vues (une vue pour les images RADARSAT-1 en mode fin, deux vues pour ENVISAT en mode standard, une vue pour RADARSAT-2 en mode fin). La présence de chatolement dans une image radar masque la texture initiale des images, ce qui rend l'interprétation difficile. Ainsi, en utilisant une seule image, la détection d'objets de petites tailles ou faiblement contrastés par rapport à leur environnement, ainsi que la détection des contours et la discrimination des surfaces deviennent ardues. Pour remédier à ce problème, le filtrage représente un moyen très utilisé pour réduire le chatolement. Cependant, il ne faut pas oublier que le chatolement contient des informations sur le capteur et la surface observée, et que réduire ces informations peut nuire à l'analyse des valeurs numériques enregistrées dans chaque pixel de l'image radar [28].

I.5 Équation du radar

L'équation radar est une notion importante pour l'étude des différents systèmes radars. En effet, elle établit le bilan de liaison entre l'émetteur et le récepteur [29].

Elle est donnée par l'équation suivante :

$$P_{Rx} = \frac{P_{Tx} G_{Tx} G_{Rx} \sigma \lambda^2}{(4\pi)^3 R^4} \quad (\text{I.16})$$

où P_{Tx} et P_{Rx} sont les puissances émise et reçue, G_{Tx} et G_{Rx} sont les gains des antennes d'émission et de réception et σ est la surface équivalente radar. Cette dernière dépend principalement des caractéristiques de la cible (forme, taille, matériaux qui la constituent ...) et de la longueur d'onde λ et traduit la surface (fictive) vue par le radar. Enfin, la distance R séparant la cible du radar est directement liée au temps de propagation de l'onde électromagnétique.

L'équation radar peut renseigner sur la portée maximale $R_{\max}$ du radar, qui est inversement proportionnelle à la puissance minimale détectable par le récepteur $P_{Rx_{\min}}$ [30]:

$$R_{\max} = \left[\frac{P_{Tx} G_{Tx} G_{Rx} \sigma \lambda^2}{P_{Rx_{\min}} (4\pi)^3} \right]^{\frac{1}{4}} \quad (\text{I.17})$$

Ainsi, pour augmenter la portée d'un radar, il faut augmenter la puissance émise. A titre d'exemple, pour doubler la portée, la puissance émise doit être multipliée par 16. De plus, lors de la conception du système on en tient compte pour envisager l'utilisation d'antennes directives et de récepteur d'une grande sensibilité.

I.6 Principaux domaines d'application des images SAR

L'intérêt des images obtenues par le SAR réside dans le fait que les ondes émises ne sont pratiquement pas perturbées par les phénomènes atmosphériques. Cette propriété permet d'obtenir des images de la surface terrestre de jour comme de nuit et indépendamment des conditions météorologiques. Outre cet intérêt, l'utilisation des hyperfréquences fournit des informations nouvelles sur la nature de la scène illuminée par rapport aux capteurs visibles et infrarouges, permettant en cela une caractérisation plus complète de la surface terrestre. Nous distinguons principalement 3 champs d'application:

- L'étude de l'océan : les radars imageurs permettent de dresser une carte de l'océan mettant en évidence les phénomènes marins tels que les ondes, les courants, les remous et les ondes internes.

- L'étude des glaces : les radars imageurs permettent de suivre l'évolution de la fonte des glaces ainsi que le déplacement des icebergs.
- La surveillance des zones côtières: notamment la surveillance de pêches, du trafic maritime et la détection des films d'hydrocarbures.
- La restitution du relief: utilisation de la technique de l'interférométrie RSO ce qui permet de restituer le relief du terrain ou de mesurer les mouvements du sol qui ont eu éventuellement lieu entre les acquisitions des images RSO avec une précision dans l'échelle de la longueur d'onde radar [19].

I.7 Conclusion

Dans ce chapitre, nous avons présenté les bases fondamentales de l'imagerie SAR, en mettant en évidence le principe de fonctionnement du radar et les techniques de formation d'images. Nous avons également expliqué les notions clés de résolution radiale et azimutale, qui déterminent la qualité des images obtenues.

Grâce à ses caractéristiques uniques, le SAR constitue un outil puissant pour l'observation de la Terre dans diverses conditions. Ce chapitre sert ainsi de base théorique pour les chapitres suivants, où des méthodes avancées de traitement et d'amélioration des images SAR seront abordées.

Chapitre II

Caractérisation et méthodes de filtrage classiques

II.1 Introduction

Les images radar à synthèse d'ouverture (SAR) constituent une source d'information essentielle pour l'observation de la Terre grâce à leur capacité d'acquisition indépendante des conditions météorologiques et de l'éclairement solaire. Cependant, la qualité visuelle et l'interprétation de ces images sont fortement affectées par un bruit particulier appelé speckle. Ce phénomène résulte de l'interférence cohérente des ondes radar rétrodiffusées par les multiples diffuseurs présents à l'intérieur d'une cellule de résolution. Il se manifeste sous forme de fluctuations aléatoires de l'intensité des pixels, dégradant ainsi la qualité radiométrique de l'image et compliquant les tâches d'analyse, de segmentation, de classification et de détection d'objets.

Contrairement au bruit additif rencontré dans de nombreuses applications de traitement d'images optiques, le speckle est généralement modélisé comme un bruit multiplicatif, dont la variance dépend du signal observé. Cette caractéristique rend son élimination particulièrement complexe puisqu'il est nécessaire de réduire les fluctuations indésirables tout en préservant les informations structurales importantes telles que les contours, les textures et les détails fins de la scène observée.

Face à cette problématique, de nombreuses méthodes de filtrage ont été développées depuis les premières applications de l'imagerie SAR. Les approches classiques reposent principalement sur l'exploitation des statistiques locales de l'image afin d'estimer la contribution du bruit et d'adapter le filtrage aux caractéristiques de chaque région. Parmi les filtres les plus représentatifs figurent les filtres de Lee, Frost et Kuan, qui constituent encore aujourd'hui des références fondamentales dans le domaine du débruitage SAR. Ces méthodes cherchent à établir un compromis entre la réduction efficace du speckle et la conservation des détails significatifs de l'image.

Ce chapitre présente tout d'abord les caractéristiques statistiques du speckle et les modèles couramment utilisés pour sa représentation dans les images SAR. Ensuite, les principales méthodes classiques de filtrage adaptatif seront étudiées, en mettant en évidence leurs fondements théoriques, leurs formulations mathématiques ainsi que leurs limites. Cette analyse constitue une base indispensable pour comprendre les approches plus avancées qui seront abordées dans les chapitres suivants.

II.2 Bruit

On peut définir le bruit [31] comme étant une dégradation dans l'image, provoquée par une perturbation externe. Si une image est envoyée par voie électronique d'un endroit à un autre,

via le satellite ou la transmission sans fil, ou par le câble réseau, on peut s'attendre à des erreurs se produisant dans le signal de l'image. Ces erreurs apparaissent sur l'image de sortie de différentes manières en fonction du type de la perturbation dans le signal. Généralement, on peut savoir les types d'erreurs à attendre, et donc le type de bruit sur l'image, d'où nous pouvons choisir la méthode la plus adaptée pour réduire les effets.

Le filtrage d'une image corrompue par le bruit est donc un domaine important de la restauration d'image.

Soit $f(.)$ une image. L'image sans bruit est notée $g(.)$, et le composant de bruit est noté $q(.)$.

La décomposition la plus courante de l'image $f(.)$ est additive:

$$f(.) = g(.) + q(.) \quad (\text{II.1})$$

Par exemple, le bruit gaussien est habituellement considéré comme un composant additif. Le deuxième le plus courant est multiplicatif :

$$f(.) = g(.) \cdot q(.) \quad (\text{II.2})$$

La littérature scientifique distingue plusieurs types de bruit susceptibles d'affecter les images. Cependant, cette étude se focalise exclusivement sur le bruit de speckle, en raison de son importance dans les applications considérées.

II.2.1 Bruit de speckle

Le speckle est un phénomène inhérent aux systèmes d'imagerie radar cohérente tels que le radar à synthèse d'ouverture (SAR). Il apparaît sous la forme d'un motif granuleux aléatoire qui dégrade la qualité visuelle des images et complique leur interprétation. Contrairement aux bruits électroniques classiques, le speckle ne résulte pas d'un défaut du capteur mais constitue une conséquence directe du processus physique de formation de l'image radar [32].

L'origine du speckle est liée à la nature cohérente de l'onde radar. En effet, chaque cellule de résolution contient généralement un grand nombre de diffuseurs élémentaires qui rétrodiffusent l'énergie électromagnétique incidente. Le signal reçu par l'antenne radar correspond à la somme complexe de toutes ces contributions. Les différences aléatoires de phase entre les ondes rétrodiffusées provoquent des phénomènes d'interférences constructives et destructives, engendrant ainsi d'importantes fluctuations de l'intensité observée d'un pixel à l'autre, même dans les zones physiquement homogènes [33].

Le speckle est généralement modélisé comme un bruit multiplicatif, ce qui le distingue des modèles additifs couramment utilisés en traitement d'images optiques. Dans ce cadre, l'image

observée $I(x, y)$ est exprimée comme le produit de la réflectivité réelle de la scène $R(x, y)$ et d'une composante aléatoire $S(x, y)$ représentant le speckle :

$$I(x, y) = R(x, y) \cdot S(x, y) \quad (\text{II.3})$$

où $R(x, y)$ représente l'information utile de la scène et $S(x, y)$ la variable aléatoire décrivant le bruit de speckle. Cette modélisation implique que la variance du bruit dépend directement de l'amplitude du signal observé, rendant son élimination plus complexe que celle d'un bruit additif classique.

La présence du speckle entraîne une diminution du contraste radiométrique, une dégradation des contours et une réduction des performances des algorithmes de segmentation, de classification et de détection. Pour cette raison, le développement de techniques de filtrage adaptées représente une étape essentielle dans la chaîne de traitement des images SAR [34].

II.2.2 Restauration et suppression des bruits

Le filtrage d'une image est une tâche essentielle dans le traitement d'image. Il existe une multitude de méthodes pour la suppression de bruit dans l'image ou dans un ensemble de données. La propriété importante de ces méthodes est qu'elle devrait complètement enlever le bruit aussi loin que possible dans l'image ainsi que la conservation de ses contours. Traditionnellement, il y a deux types de modèles c.-à-d. les modèles linéaires et les modèles non-linéaires.

Généralement, les modèles linéaires sont les plus employés. Les avantages des modèles linéaires est la vitesse et leurs limitations. Ils ne peuvent pas préserver les contours des images d'une façon efficace c.-à-d. les contours, qui sont identifiés comme discontinuités dans l'image sont masqués (présentés comme une région floue). D'autre part, les modèles non linéaires peuvent présenter les contours d'une manière bien meilleure que les modèles linéaires [35].

Afin de concevoir des méthodes de filtrage adaptées au bruit de speckle, il est nécessaire de comprendre au préalable son comportement statistique. La modélisation probabiliste du speckle permet en effet de caractériser ses fluctuations et de développer des algorithmes capables de distinguer le bruit de l'information utile. Les principales distributions utilisées pour représenter le speckle dans les images SAR sont présentées dans la section suivante.

II.3 Modélisation statistique du Speckle

II.3.1 Distribution Gamma

La distribution Gamma est une loi à deux paramètres. Sa densité de probabilité est donnée par [36]:

$$G_{|\mu,L|}(x) = \frac{1}{\Gamma(L)} \frac{L}{\mu} \left(\frac{L \cdot x}{\mu} \right)^{L-1} \cdot \exp\left(-\frac{L \cdot x}{\mu}\right) \quad (\text{II.4})$$

Où $\Gamma(L)$ est la fonction Gamma définie par :

$$\Gamma(L) = \int_0^{\infty} x^{L-1} \cdot \exp(-x) \cdot dx \quad (\text{II.5})$$

Le paramètre μ est appelé paramètre d'échelle et L paramètre de forme. Elle sert souvent à modéliser les phénomènes résultants de la somme de L (nombre de looks) variables aléatoires indépendantes qui suivent une loi exponentielle de moyenne μ .

En effet, on remarque que pour $L = 1$ on obtient bien la loi exponentielle :

$$E_{\mu}(x) = \frac{1}{\mu} \cdot \exp\left(-\frac{x}{\mu}\right) \quad (\text{II.6})$$

La Figure II.1 illustre la densité de cette loi avec $\mu = 1$ et $L = 2, 3, 4$.

On note que L se comporte effectivement comme un paramètre de forme : $G_{|\mu,L|}$ est d'autant plus centrée autour de μ que L est grand.

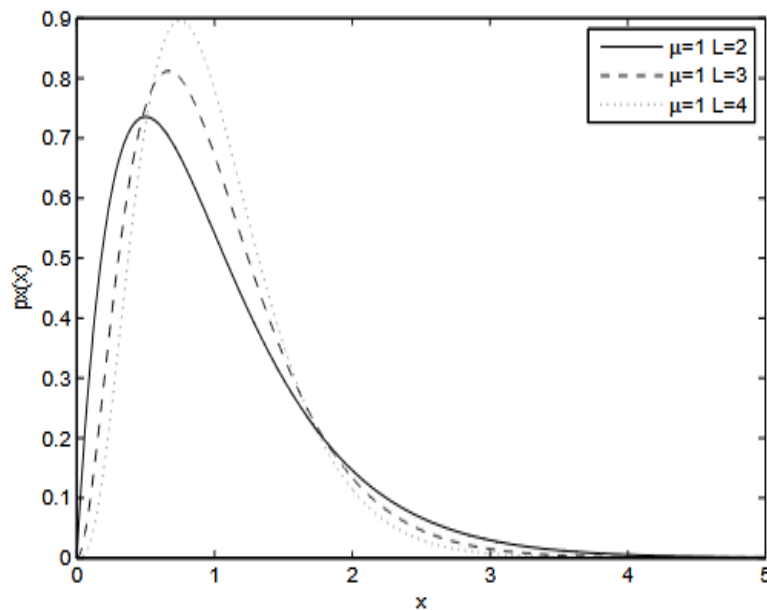


Figure II. 1. Fonction de densité de probabilité de la distribution de Gamma avec : $\mu = 1$ et $L = 2, 3, 4$.

II.3.2 Nombre de looks (L)

Le nombre de looks (L) constitue un paramètre fondamental en imagerie SAR. Il correspond au nombre d'observations indépendantes utilisées pour produire une image finale à partir du signal radar brut.

L'augmentation du nombre de looks permet de réduire les fluctuations du speckle grâce à un effet de moyennage statistique. La variance de l'intensité est liée au nombre de looks par la relation :

$$\sigma^2 = \frac{\mu^2}{L} \quad (\text{II.7})$$

où :

- σ^2 est la variance de l'intensité ;
- μ est la moyenne de l'intensité ;
- L représente le nombre de looks.

Cette équation montre que la variance décroît inversement avec L . Ainsi, une image multilook présente un niveau de speckle plus faible qu'une image single-look. Toutefois, cette amélioration radiométrique s'accompagne généralement d'une diminution de la résolution spatiale, ce qui impose un compromis entre réduction du bruit et conservation des détails [32].

II.3.3 Impact sur la qualité d'image

Le speckle affecte significativement la qualité visuelle et radiométrique des images SAR. Il provoque une augmentation artificielle des variations locales d'intensité, réduisant ainsi le contraste entre les régions homogènes et compliquant l'identification des structures présentes dans la scène observée.

Les principaux effets du speckle sont :

- diminution du contraste radiométrique ;
- dégradation des contours et des détails fins ;
- réduction des performances des méthodes de segmentation ;
- augmentation des erreurs de classification ;
- difficulté d'interprétation visuelle des images.

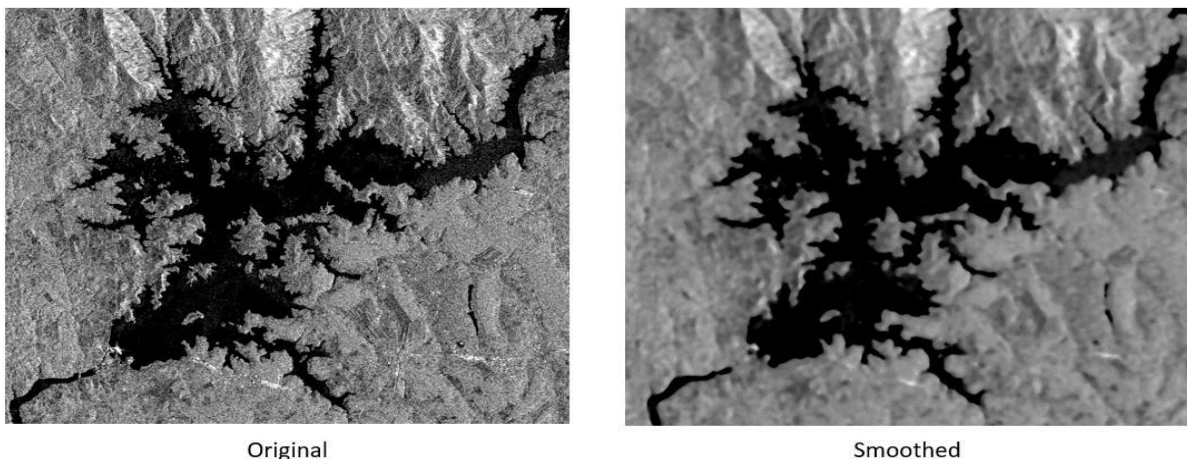


Figure II. 2. Impact du bruit de speckle sur la qualité des images SAR.

Afin d'évaluer quantitativement l'influence du bruit de speckle ainsi que les performances des méthodes de filtrage, plusieurs indicateurs sont couramment utilisés, notamment le PSNR (Peak Signal-to-Noise Ratio), le SSIM (Structural Similarity Index Measure) et l'UQI (Universal Quality Index). Une réduction efficace du bruit de speckle doit permettre d'améliorer l'homogénéité des régions de l'image tout en préservant les informations structurales et les détails essentiels qu'elle contient [34], comme l'illustre la figure suivante.

Les dégradations induites par le bruit de speckle justifient le recours à des techniques de traitement spécifiques visant à améliorer la qualité des images SAR. De nombreuses approches ont ainsi été développées afin d'atténuer les fluctuations du speckle tout en préservant les caractéristiques essentielles de la scène observée. Les principales méthodes classiques de débruitage sont présentées dans la section suivante.

II.4 Méthodes Classiques de Débruitage

La réduction du bruit de speckle constitue une étape essentielle dans la chaîne de traitement des images radar à synthèse d'ouverture (SAR). L'objectif principal des méthodes de débruitage est d'atténuer les fluctuations aléatoires induites par le speckle tout en préservant les informations radiométriques et géométriques importantes de l'image, telles que les contours, les textures et les détails structuraux. En raison de la nature multiplicative du speckle, les techniques classiques de filtrage développées pour les images optiques ne sont pas directement applicables aux données SAR. Ainsi, plusieurs méthodes spécifiques ont été proposées afin d'exploiter les propriétés statistiques du signal radar et d'adapter localement le processus de filtrage [34].

Les premières approches reposent sur l'analyse des statistiques locales calculées dans une fenêtre de voisinage autour de chaque pixel. Ces méthodes supposent généralement que les caractéristiques statistiques du signal restent relativement homogènes à l'intérieur de la fenêtre d'observation. À partir de la moyenne et de la variance locale, elles estiment la contribution du bruit de speckle puis déterminent un coefficient de filtrage permettant de lisser les régions homogènes tout en limitant la dégradation des contours [37].

Parmi les filtres adaptatifs les plus connus figurent le filtre de Lee, le filtre de Frost et le filtre de Kuan. Ces méthodes constituent des références majeures dans le domaine du traitement des images SAR en raison de leur simplicité de mise en œuvre et de leur efficacité dans les zones relativement homogènes. Le filtre de Lee exploite les statistiques locales pour réaliser une estimation linéaire minimale de l'image originale, tandis que le filtre de Frost applique une pondération exponentielle dépendant des caractéristiques locales du signal. Le filtre de Kuan,

quant à lui, améliore le modèle de Lee en prenant explicitement en compte la nature multiplicative du speckle dans l'estimation du coefficient de filtrage [38].

Les sous-sections suivantes présentent en détail les principaux filtres classiques de débruitage appliqués aux images SAR, notamment les filtres de Lee, Frost et Kuan, en décrivant leurs principes théoriques, et leurs formulations mathématiques.

II.4.1 Filtres adaptatifs

Les filtres adaptatifs sont très intéressants car ils utilisent des statistiques locales de l'image afin de modifier leur comportement. Les grandes trois familles des filtres sont présentées ici : la première, historiquement la plus ancienne, repose sur une minimisation de l'erreur quadratique moyenne EQM, la seconde est une variante adaptative du filtre moyenne et enfin, la dernière, plus récente, utilise une évaluation locale des statistiques de l'image [39].

Parmi les différentes catégories de filtres adaptatifs, les méthodes fondées sur la minimisation de l'erreur quadratique moyenne (EQM) occupent une place importante. Elles exploitent les statistiques locales de l'image afin d'estimer la contribution du bruit et d'adapter le filtrage aux caractéristiques de chaque région.

II.4.1.1 Méthodes basées sur EQM minimale

Les différents filtres détaillés ci-après reposent sur une minimisation de l'erreur quadratique moyenne (EQM) en utilisant des statistiques du second ordre. Ils se fondent sur l'évaluation du coefficient de variation (CV) dans l'image. On distingue deux types de coefficients de variation : un coefficient de variation global et un coefficient de variation local. Le premier donne une caractérisation globale du speckle présent dans l'image alors que le second en donne une estimation locale. Cet estimateur local peut, de plus, être considéré comme un détecteur de contours dans les images bruitées par le speckle, comme c'est le cas dans les travaux de Tauber et al [40]. Afin d'expliquer le principe de tous les filtres de la section suivante, nous avons repris les conventions de et modes de description utilisés par Gagnon et al [41]. Nous considérons dans la suite une image I , corrompue par un bruit stationnaire multiplicatif η , tel que l'image finale I_f , soit égale à $I_f = \eta \cdot I$.

La variable L représente une mesure du ratio signal sur bruit :

$$L = \frac{\overline{I_f}}{\sigma_f} \quad (\text{II.8})$$

Tel que : $\overline{I_f}$ la valeur moyenne de I_f et σ_f , son écart-type.

Une fois cette notation explicitée, nous allons à présent détailler quelques filtres classiques ayant rencontré une grande adhésion en termes de filtrage d'images ultrasonores.

II.4.1.2 Filtre de Kuan

Dans l'approche proposée par Kuan [38] le modèle de bruit multiplicatif est d'abord transformé en un modèle additif dépendant du signal:

$$f = u + u(\eta - 1) \quad (\text{II.9})$$

Le filtre de Kuan est semblable au filtre de Lee, mais avec un gain qui se calcule comme suit :

$$k = \frac{1 - \frac{\gamma_g^2}{\gamma_l^2}}{1 + \gamma_g^2} \quad (\text{II.10})$$

Selon la relation précédente nous pouvons distinguer deux cas :

- Si $\gamma_g \approx \gamma_l \Rightarrow \frac{\gamma_g}{\gamma_l} \approx 1 \Rightarrow k \approx 0$: le gain est faible dans les zones homogènes.
- Si $\gamma_g^2 \ll \gamma_l^2 \Rightarrow k \approx 1$: le gain est proche de l'unité sur les contours.

La Figure II.3 illustre les résultats du filtrage de Kuan sur l'image « Clock.tiff » :

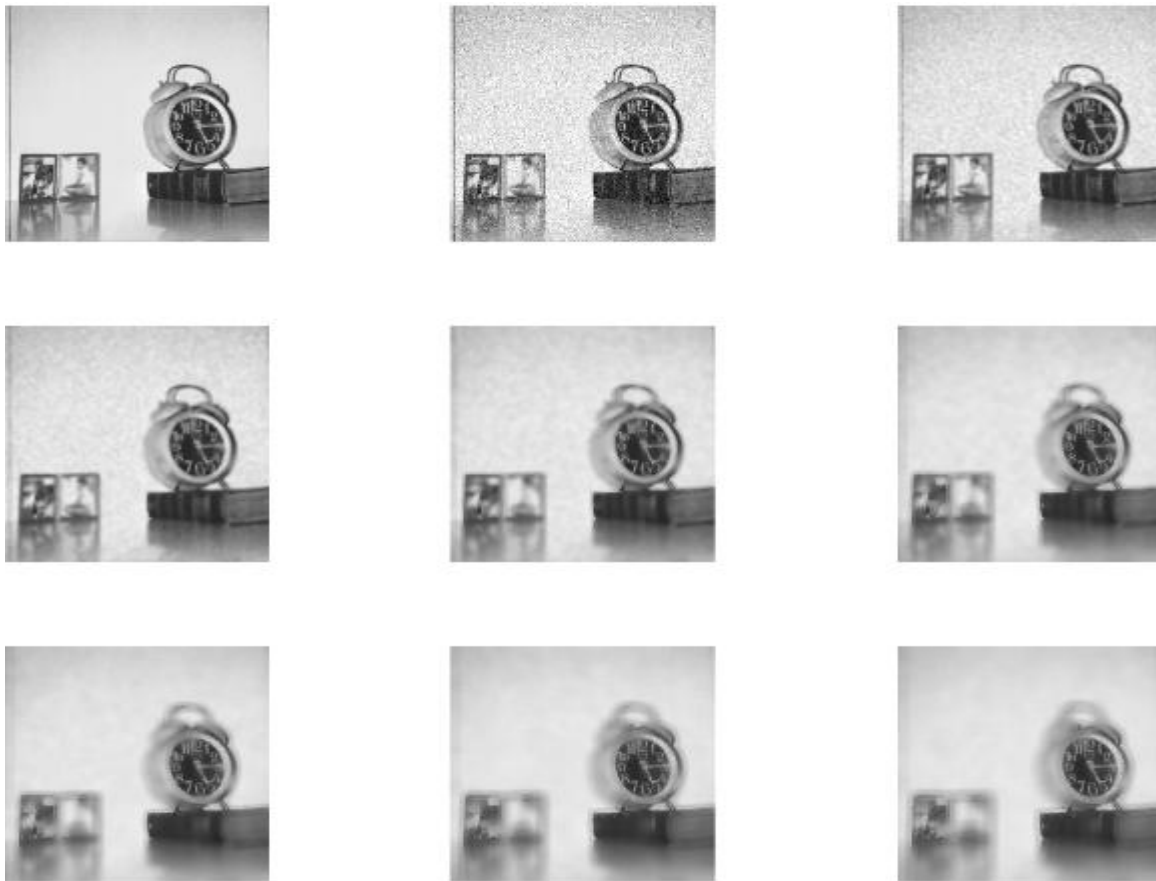


Figure II. 3. Restauration par le filtre de Kuan de l'image Clock.tiff: de gauche à droite : originale, corrompue par un bruit multiplicatif, restaurée pour les fenêtres:

$$W = 3, W = 5, W = 7, W = 9, W = 11, W = 13, W = 15$$

Le filtre proposé par Kuan et al [42] est un estimateur linéaire de l'erreur quadratique moyenne minimale (EQMM) et s'appuie sur une mesure locale d'intensité et de variance des pixels. Dans un premier temps, un critère de l'EQMM est calculé pour un modèle de bruit additif tel que :

$$I_f = I + \eta \quad (\text{II.11})$$

Avec I_f l'image résultat, I l'image sans bruit et η le bruit. Une fois ce calcul effectué, il est dérivé afin de considérer un bruit multiplicatif de la forme :

$$I_f = I + (\eta - 1) \cdot I \quad (\text{II.12})$$

La valeur du pixel débruité $\tilde{I}$ est ainsi estimée par la formule [41] :

$$\tilde{I} = \overline{I_f} + \frac{\sigma_f^2 (I_f - \overline{I_f})}{\sigma_I^2 + \left(\overline{I_f}^2 + \sigma_I^2 \right) / L} \quad (\text{II.13})$$

avec :

$$\sigma_I^2 = \frac{L \sigma_{I_f}^2 - \overline{I_f}^2}{L + 1} \quad (\text{II.14})$$

on a donc :

$$\tilde{I} = \overline{I_f} + k (I_f - \overline{I_f}) \quad (\text{II.15})$$

Tel que k le gain du filtre [40], il est défini comme suit :

$$k = \frac{\sigma_I^2}{\sigma_I^2 + \left(\overline{I_f}^2 + \sigma_I^2 \right) / L} \quad (\text{II.16})$$

Sur une zone homogène, la valeur du gain est très faible et le filtre peut donc être approximé par un filtre moyennneur [40]. A l'inverse, sur les zones de fort contour, le gain est proche de 1 et le filtrage est donc arrêté. Ce même principe peut être retrouvé dans le filtre présenté ci-après, qui peut être considéré comme un cas particulier du filtre de Kuan, même s'il lui est antérieur.

II.4.1.3 Filtre de Lee

Le filtre de Lee [34] est donné par la formulation suivante :

$$\hat{u} = \bar{f} + k (f - \bar{f}) \quad (\text{II.17})$$

Où f est l'image observée, $\bar{f}$ la moyenne de f et k est appelé gain du filtre, il est donné par l'expression:

$$k = 1 - \left(\frac{\gamma_g}{\gamma_l} \right)^2 \quad (\text{II.18})$$

Ce paramètre k est évalué en comparant le CV local au CV global :

- Si $\gamma_g \approx \gamma_l \Rightarrow \frac{\gamma_g}{\gamma_l} \approx 1 \Rightarrow k \approx 0$: sur une zone homogène le CV local et le CV global sont proches; la valeur de k est donc très faible et le filtre se comporte comme un filtre moyennneur.
- Si $\gamma_l \gg \gamma_g \Rightarrow \frac{\gamma_g}{\gamma_l} \approx 0 \Rightarrow k \approx 1$: sur un fort contour, le CV local est très supérieur au CV global. La valeur de k s'approche donc de l'unité et le filtrage est arrêté.
- Si $0 < k < 1 \Rightarrow 0 < 1 - \left(\frac{\gamma_g}{\gamma_l} \right)^2 < 1 \Rightarrow 0 < \left(\frac{\gamma_g}{\gamma_l} \right)^2 < 1$, le filtre lisse l'image à la manière d'un filtre passe-bas.

La Figure II.4 montre le résultat du filtrage de Lee avec des fenêtres de tailles différentes pour l'image « clock.tiff ».

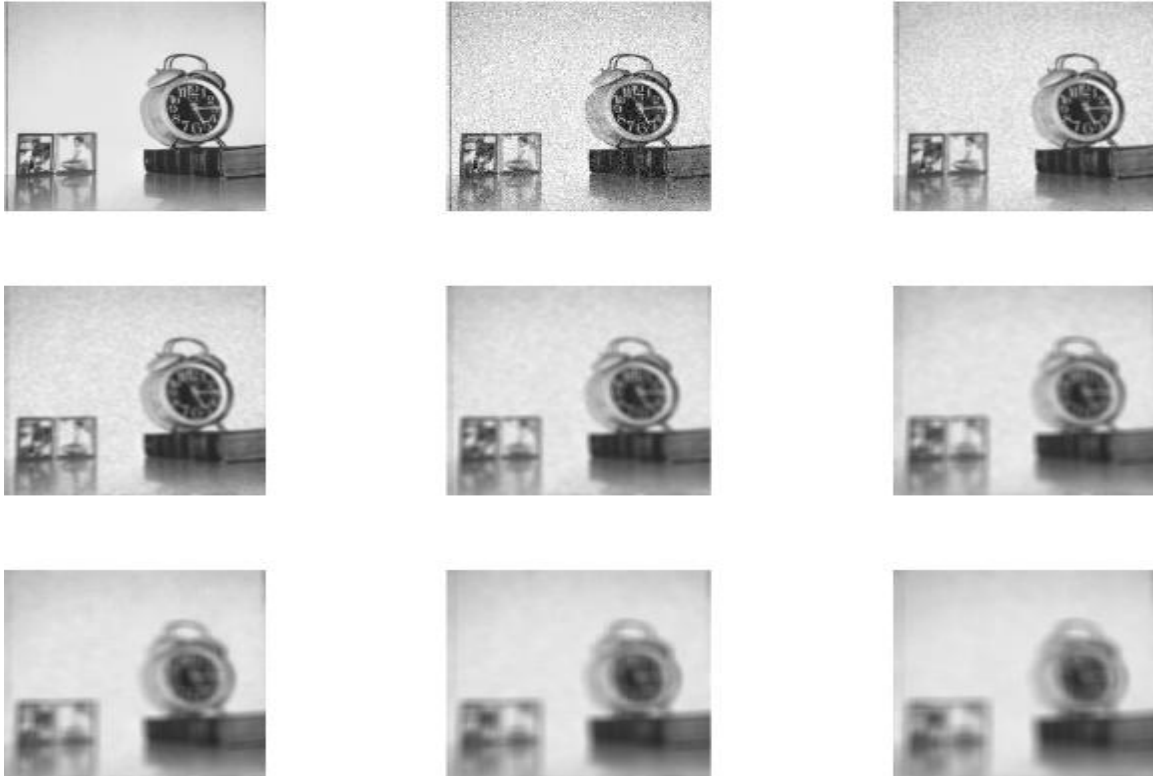


Figure II. 4. Restauration par le filtre de Lee de l'image Clock.tiff: de gauche à droite : originale, corrompue par un bruit multiplicatif, restaurée pour les fenêtres:

$$W = 3, W = 5, W = 7, W = 9, W = 11, W = 13, W = 15$$

Faisant l'approximation d'une distribution gaussienne, Lee et al [43] développement en 1980 une méthode de filtrage qui effectue une approximation linéaire du bruit et qui diffère principalement du précédent par le terme $\frac{\sigma_I^2}{L}$ qui n'existe pas dans cette version du filtre. Un pixel débruité par le filtre de Lee, sera donc selon la formule :

$$\tilde{I} = \overline{I_f} + \frac{\sigma_I^2 (I_f - \overline{I_f})}{\sigma_I^2 + \overline{I_f}^2 / L} \quad (\text{II.19})$$

Avec :

$$\sigma_I^2 = \frac{L\sigma_{I_f}^2 - \overline{I_f}^2}{L+1} \quad (\text{II.20})$$

II.4.1.4 Filtre de Frost

Le filtre de Frost [44] peut être considéré comme une variante du filtre de Wiener [45] adapté au bruit multiplicatif. L'image filtrée est obtenue en convoluant l'image par la réponse impulsionnelle h du filtre de Frost :

$$f = u \times \eta - h \quad (\text{II.21})$$

La réponse impulsionnelle du filtre de Frost se calcule comme suit :

$$m(t_0) = K_1 \cdot \exp(-K_2 \cdot \gamma_I^2(t_0) \cdot |t|) \quad (\text{II.22})$$

où $t_0 = (i_0, j_0)$ est le centre de la fenêtre de calcul, $t = (i - i_0, j - j_0)$ les coordonnées du pixel courant exprimées dans le repère centré sur t_0 , $|t|$ la distance au pixel t_0 , K_1 est une constante de normalisation introduite pour ne pas biaiser la moyenne et K_2 le paramètre du filtre.

Le principe de ce filtre est que tous les pixels de la fenêtre participent au filtrage mais avec un poids qui diminue en fonction de :

- La distance $|t|$ au pixel central t_0
- L'hétérogénéité mesurée par le CV local γ_I

K_2 Permet de régler la quantité de filtrage:

- Lorsque K_2 est grand, le filtrage est plus faible,
- Lorsque K_2 est faible, la fonction exponentielle décroît plus lentement: le filtrage est plus fort.

K_2 est choisi de telle sorte que le produit $K_2 \gamma_l^2$ tend vers zéro sur les zones homogènes et devient suffisamment grand aux abords des contours pour que le filtrage s'arrête. La figure II.5 montre les résultats du filtrage de Frost sur l'image « Clock.tiff ».

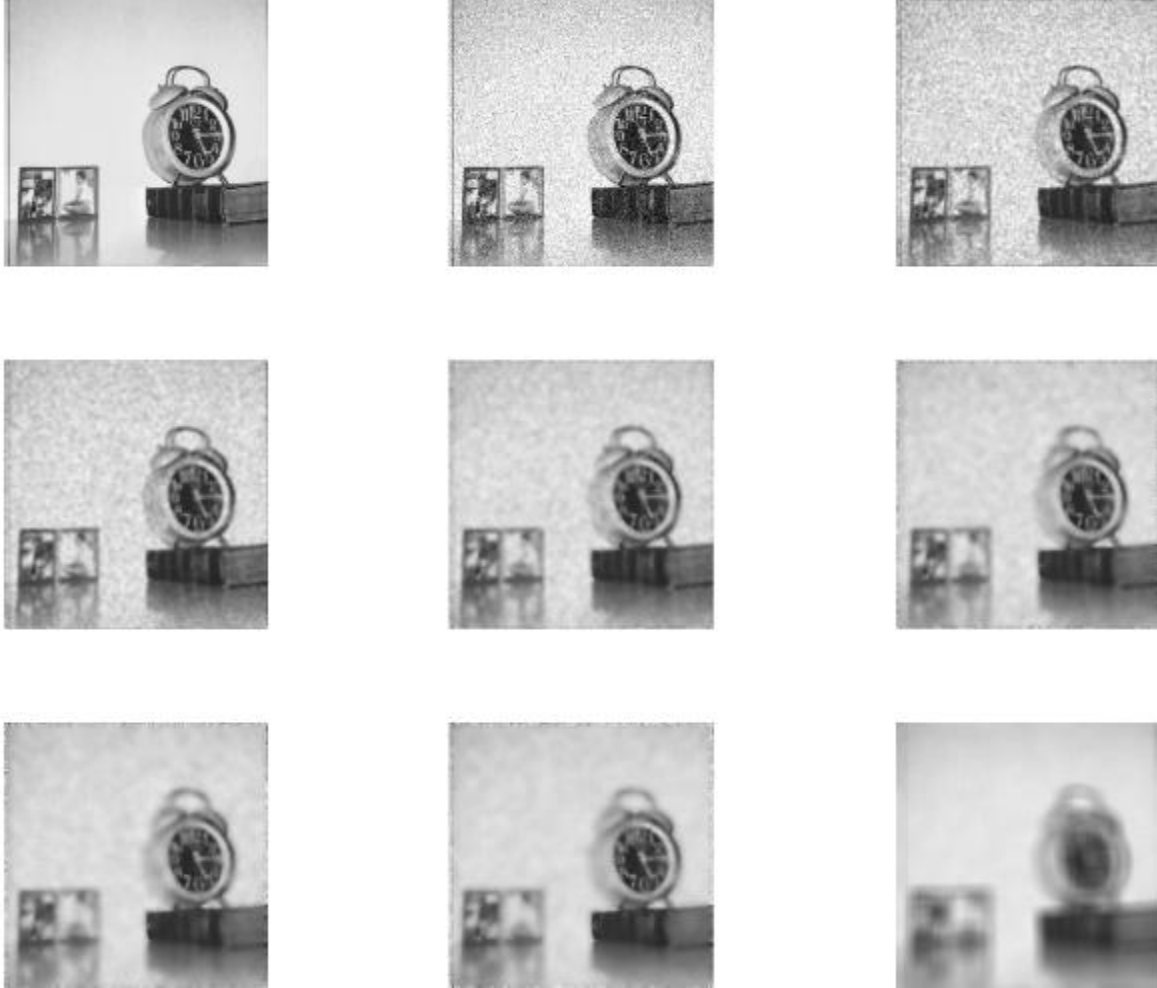


Figure II. 5. Restauration par le filtre de Frost de l'image Clock.tiff de gauche à droite:

originale, corrompue par un bruit multiplicatif, restaurée pour les fenêtres :

$$W = 3, W = 5, W = 7, W = 9, W = 11, W = 13, W = 15$$

A l'inverse des deux algorithmes précédents, le filtre de Frost [46] propose une modélisation de la réflectivité de la scène. Il peut, de plus, être considéré comme une variante du filtre de Wiener [47], adaptée au bruit multiplicatif. Cette modélisation est réalisée par le biais d'une fonction d'autocorrélation supposée exponentielle. Une convolution de la valeur des pixels avec la réponse impulsionnelle exponentielle du filtre est réalisée dans une sous-fenêtre de traitement donnée. La réponse impulsionnelle m , a pour expression :

$$m = \exp\left(-KC_{I_f}(t_0)|t|\right), C_{I_f} = \frac{\sigma_{I_f}}{I_f} \quad (\text{II.23})$$

Où K est le paramètre du filtre, t_0 le centre de la fenêtre de calcul, t les coordonnées du pixel courant exprimées dans le repère t_0 et $|t|$ la distance au pixel t_0 .

Tous les pixels d'une fenêtre de traitement participent donc au processus de filtrage modulo une pondération donnée. Cette pondération dépend de la distance au pixel central t_0 et du caractère hétérogène de la scène, exprimé ici par la valeur de C_{I_f} .

C_{I_f} Représente l'écart-type des intensités de l'image divisée par leur moyenne.

Bien que les filtres adaptatifs tels que ceux de Lee, Kuan et Frost offrent des performances satisfaisantes dans les régions homogènes, leur efficacité demeure limitée dans les zones présentant des textures complexes ou des détails fins. Afin d'améliorer la préservation des structures de l'image, de nouvelles approches fondées sur l'exploitation de similarités à plus grande échelle ont été proposées. Ces méthodes, dites non locales, sont présentées dans la section suivante.

II.4.2 Méthodes non locales

Les méthodes non locales représentent une évolution importante des techniques de réduction du speckle. Contrairement aux approches fondées exclusivement sur le voisinage immédiat du pixel, elles exploitent les similarités existantes entre différentes régions de l'image, même lorsqu'elles sont spatialement éloignées.

L'approche Probabilistic Patch-Based (PPB) repose sur la comparaison statistique de blocs d'image (patches) afin d'identifier les zones présentant des caractéristiques similaires. Les informations provenant de ces régions sont ensuite combinées pour produire une estimation plus robuste du signal utile. Cette stratégie améliore significativement la conservation des textures et des détails fins par rapport aux filtres locaux traditionnels.

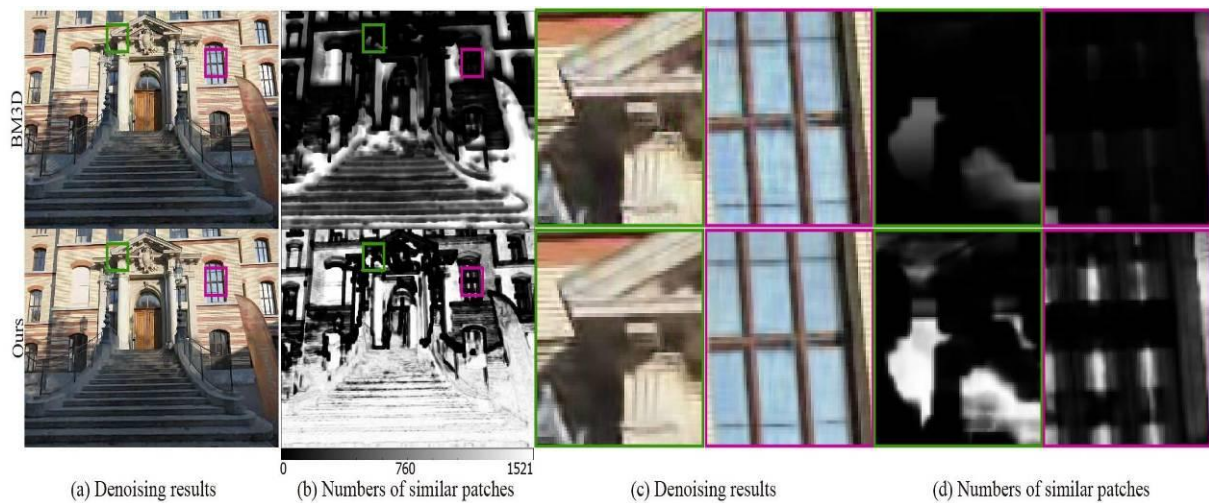


Figure II. 6. Principe des méthodes non locales basées sur la recherche de patches similaires.

L'algorithme SAR-BM3D applique quant à lui un regroupement collaboratif de blocs similaires dans un espace transformé, suivi d'une étape de seuillage et de reconstruction. Cette méthode est aujourd'hui considérée comme l'une des références les plus performantes pour la réduction du speckle dans les images SAR mono-canal [48].

Donc, les méthodes présentées dans la section du filtrage adaptatif reposent sur une comparaison locale des valeurs des pixels. Une approche basée sur une comparaison globale de patches est proposée par Buades et al [49] pour les images naturelles, puis ré-exploitée par Coupé et al [50] en 2009.

La description de cette approche et de son adaptation sont présentées ci- dessous.

II.4.2.1 Méthode des moyennes non locales

Les auteurs dans [49] partent du constat que la plupart des méthodes de filtrage ont pour objectif la réduction du bruit et la reconstruction des principales structures géométriques mais qui ne préservent ni les détails, ni la texture de ces structures.

Considérant le degré de redondance existant dans les images naturelles, les auteurs redéfinissent la notion de voisinage. Le voisinage d'un pixel i devient l'ensemble des pixels j de l'image tels qu'une fenêtre centrée autour du pixel j ressemble à une fenêtre centrée autour du pixel i . La similarité est calculée à partir des valeurs de niveau de gris des pixels des régions considérées. La nouvelle valeur d'un pixel sera ainsi calculée en fonction de toutes les valeurs des pixels de l'image pondérées par leur degré de similarité. Une des principales limitations de l'utilisation de cette méthode est le temps de calcul conséquent engendré par la quantité de comparaisons à effectuer [50].

II.4.2.2 Filtre des moyennes non locales bayésiennes par blocs

L'adaptation de la version Bayésienne de l'algorithme précédent, proposée par Kervrann et al [51], a été reprise par Coupé et al [50] et testée sur des images d'échographie 2D intra opératoires cérébrales et 3D hépatiques. La formulation Bayésienne a été utilisée afin de prendre en compte un modèle a priori du bruit dans l'échographie (introduit par Loupas et al [52] en 1989). La distance de Pearson a été utilisée afin de comparer les patches entre eux. Une adaptation GPU de cette méthode a été proposée en 2010 par De Fontes et al [53] afin d'accélérer les calculs. Si les résultats décrits par les auteurs semblent impressionnants, on constate toutefois une nette dégradation des textures après filtrage, ce qui peut s'avérer problématique dans les cas où seule la texture peut permettre une différenciation des structures.

Les filtres présentés dans cette section reposent sur des propriétés directement visibles de l'image. Afin de clore cette première section sur les algorithmes de filtrage classiquement rencontrés dans la littérature, nous allons effectuer une brève revue de techniques moins classiques, mais qu'il nous semble incontournable de citer.

II.4.3 Approches variationnelles (Total Variation)

II.4.3.1 Filtres basés sur le phénomène de diffusion

Partant de l'équation de diffusion de chaleur telle que définie en physique, Per on a et Malik [54] proposé en 1990 une équation aux dérivées partielles de diffusion anisotrope, afin de construire un filtre itératif sur les pixels d'une image. Leur méthode tente d'adapter le procédé de diffusion aux caractéristiques locales de l'image en utilisant un coefficient de diffusion dépendant à la fois de l'espace et de l'itération considérée. Afin de favoriser un lissage intra-région et non inter-régions, un gradient d'intensité est calculé à toutes les échelles et la diffusion est arrêtée à la rencontre des frontières dans l'image. Une amélioration de cette méthode, intitulée Total Variation minimization, a été proposée en 1992 par Rudin et al [55]. Une comparaison des résultats de filtrage de données échographiques de ces deux méthodes a été effectuée par Krissian et al [41]. Elle semble montrer que les résultats obtenus par les deux méthodes sont équivalents et restent insuffisants en termes de débruitage.

La principale limitation mise en cause est l'hypothèse du modèle de bruit gaussien effectuée par ces approches. Yu et al [42] ont proposé une méthode combinant diffusion anisotrope et coefficient de variation, intitulée SRAD, pour Speckle Reducing Anisotropic Diffusion. Une version optimisée a, par ailleurs, été proposée par Krissian et al [41].

Cette méthode effectue un lissage des zones homogènes tout en renforçant les contours. Les résultats de filtrage, obtenus par le biais de cette méthode, sur des données échographiques, surpassent ceux obtenus par les deux méthodes précédentes ainsi que par les méthodes de Lee, Kuan et Frost. Ils sont comparables en termes de qualité à ceux obtenus avec la méthode classique des moyennes non-locales [43].

Toutefois, il semble que l'utilisation de la technique engendre un certain nombre d'artefacts au niveau des contours de l'image. Un autre exemple de méthode combinant coefficient de variation local et diffusion anisotrope est l'algorithme de filtrage proposé par Tauber [31].

Les résultats ont essentiellement été validés sur des images d'échocardiographie et, si les résultats semblent prometteurs, le nombre d'itérations à mettre en œuvre et par conséquent le temps de calcul reste prohibitif. Comme on l'a vu précédemment le phénomène de diffusion a

été largement utilisé dans le domaine du filtrage du speckle et a rencontré un certain succès. Une de ses principales limitations reste cependant un temps de calcul souvent prohibitif.

Afin de clore cette première section sur les algorithmes de filtrage rencontrés dans la littérature, nous allons effectuer une brève revue de techniques moins classiques, mais qu'il nous semble incontournable de citer.

II.4.3.2 Débruitage par minimisation de la variation totale

Outre les algorithmes décrits précédemment, un certain nombre de méthodes basées sur les méthodes variationnelles ont été développées et mises en œuvre à des fins de filtrage du speckle, tant sur des données radar qu'échographiques. Un certain nombre d'approches permettent de restaurer avec succès ce type d'images ont dernièrement vu le jour [46], [47], [56], [57].

Ces filtres réduisent le bruit de speckle. Ils diminuent également la résolution de l'image, ce qui rend les contours de l'image. Certaines informations sur les contours sont bien gardées, mais le bruit de speckle n'est pas entièrement lissé. Récemment, les problèmes inverses tels que la restauration d'image sont apparus dans de nombreuses applications comme la télédétection, l'imagerie médicale, l'astronomie et la photographie numérique [58]. La plupart des problèmes inverses sont non linéaires et très mal posés. Afin de résoudre ce problème, un grand nombre de techniques ont été développées. Une des techniques les plus connues est la minimisation et la régularisation de la Variation totale (TV) celle-ci forme un concept puissant pour une estimation robuste [59].

II.4.4 Synthèse et limites des approches traditionnelles

L'analyse des méthodes classiques montre qu'aucune approche ne permet de supprimer totalement le speckle tout en préservant parfaitement l'ensemble des informations présentes dans l'image. Les filtres adaptatifs offrent une faible complexité de calcul mais présentent souvent une perte de détails dans les zones complexes. Les approches variationnelles améliorent la conservation des contours grâce aux techniques de régularisation, mais nécessitent un réglage précis des paramètres et peuvent engendrer certains artefacts. Les méthodes non locales fournissent généralement les meilleurs résultats en matière de qualité visuelle et de préservation des textures, au prix d'une complexité computationnelle plus importante.

Ces limitations ont conduit au développement de nouvelles générations d'algorithmes combinant modélisation statistique, optimisation variationnelle et apprentissage automatique.

Toutefois, les méthodes traditionnelles demeurent des références incontournables pour l'évaluation des performances des techniques modernes de débruitage SAR.

II.5 Conclusion

Dans ce chapitre, les principales caractéristiques du bruit de speckle ainsi que les méthodes classiques de réduction appliquées aux images radar à synthèse d'ouverture (SAR) ont été étudiées. L'analyse de l'origine physique du speckle a montré que ce phénomène est une conséquence directe de la nature cohérente de l'imagerie radar et qu'il peut être modélisé comme un bruit multiplicatif affectant la qualité radiométrique des images.

Plusieurs familles de méthodes de débruitage ont été présentées dans ce chapitre. Les filtres adaptatifs, tels que les filtres de Lee, Kuan et Frost, exploitent les statistiques locales afin de réduire les fluctuations du speckle tout en préservant partiellement les contours. Les approches variationnelles, fondées sur la régularisation et l'optimisation, offrent une meilleure conservation des structures géométriques grâce à l'utilisation de modèles tels que la variation totale (*Total Variation*).

Malgré les progrès réalisés, aucune méthode traditionnelle ne permet d'éliminer complètement le speckle sans compromettre certains détails de l'image. Les filtres locaux souffrent souvent d'un effet de lissage excessif, tandis que les approches variationnelles et non locales impliquent généralement une complexité computationnelle plus importante. Ces limitations ont conduit au développement de nouvelles approches fondées sur l'apprentissage profond, capables d'exploiter efficacement les caractéristiques complexes des images SAR. Ces techniques, qui feront l'objet du chapitre suivant, offrent des performances de débruitage supérieures tout en assurant une meilleure préservation des structures et des informations pertinentes de la scène observée.

Chapitre III

Apprentissage profond pour la restauration d'images

III.1. Introduction

Ce chapitre présente les fondements de l'intelligence artificielle appliqués au traitement d'images, en mettant l'accent sur les réseaux de neurones artificiels et l'apprentissage profond. Après une introduction aux concepts fondamentaux, tels que le perceptron, les réseaux de neurones multicouches ainsi que les modes d'apprentissage supervisé et non supervisé, une attention particulière est accordée aux réseaux de neurones convolutionnels (CNN). Grâce à leur architecture hiérarchique et à leur capacité à extraire automatiquement des caractéristiques pertinentes, les CNN constituent une approche particulièrement adaptée aux tâches de débruitage d'images.

III.2. Fondements des réseaux de neurones artificiels

Le premier neurone formel (voir les Figures III.1 et III.2) a été proposé par les neurologues Warren McCulloch et Walter Pitts (What the frog's eye tells the frog's brain) [60]. Le neurone formel est construit comme un système composé d'une fonction de transfert qui module ses sorties en fonction de ses entrées selon des règles précises, ce qui lui procure l'aptitude à résoudre des problèmes simples. À chaque entrée est associé un poids synaptique. Il calcule tout d'abord la somme pondérée de ses entrées. Le résultat est ensuite transformé par une fonction d'activation non linéaire, la fonction de Heaviside ou fonction marche [61].

L'équation du neurone formel est donnée par :

$$z = \sum_{j=1}^n w_j \cdot x_j \quad (\text{III.1})$$

avec $x = (x_1, x_2, \dots, x_n)$ vecteur d'entrée, $w = (w_1, w_2, \dots, w_n)$ vecteur des poids synaptiques, et w_0 une constante représentant un biais. Le biais procurant une certaine malléabilité de la frontière de décision du neurone [61].

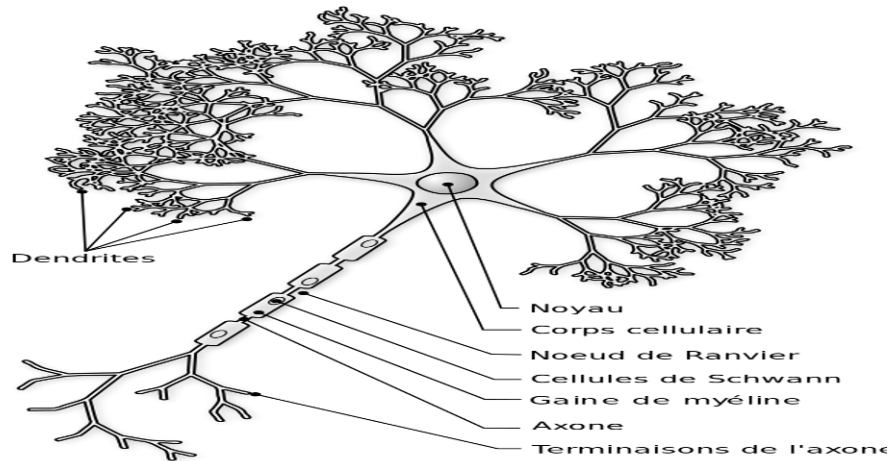


Figure III. 1. Neurone biologique [61].

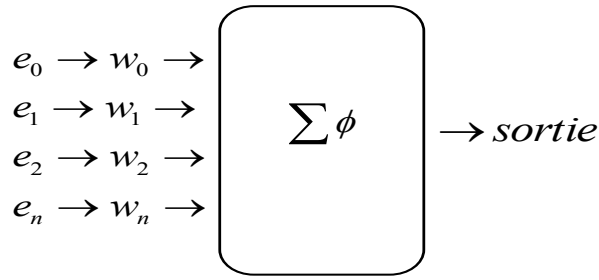


Figure III. 2. Neurone formel.

L'expression de la sortie du neurone formel est donnée par la relation suivante :

$$sortie = \phi(z) = \phi\left(\sum_{j=1}^n w_j \cdot x_j + w_0\right) \quad (III.2)$$

Le physiologiste canadien Donald Hebb fournit une méthode pour modifier les poids synaptiques appelée règle de Hebb [62]. Il a imaginé un mécanisme qui permet de modifier la valeur des coefficients synaptiques en fonction de l'activité des entités connectées ensembles. Cette méthode de modification des poids du neurone est encore aujourd'hui utilisée couramment [61].

III.2.1. Perceptron

Le neurone formel est associé à d'autres neurones afin de constituer un réseau de neurones et ainsi augmenter la complexité de résolution. Le perceptron, représenté sur la Figure III.3 a été introduit par Frank Rosenblatt en 1957 [63].

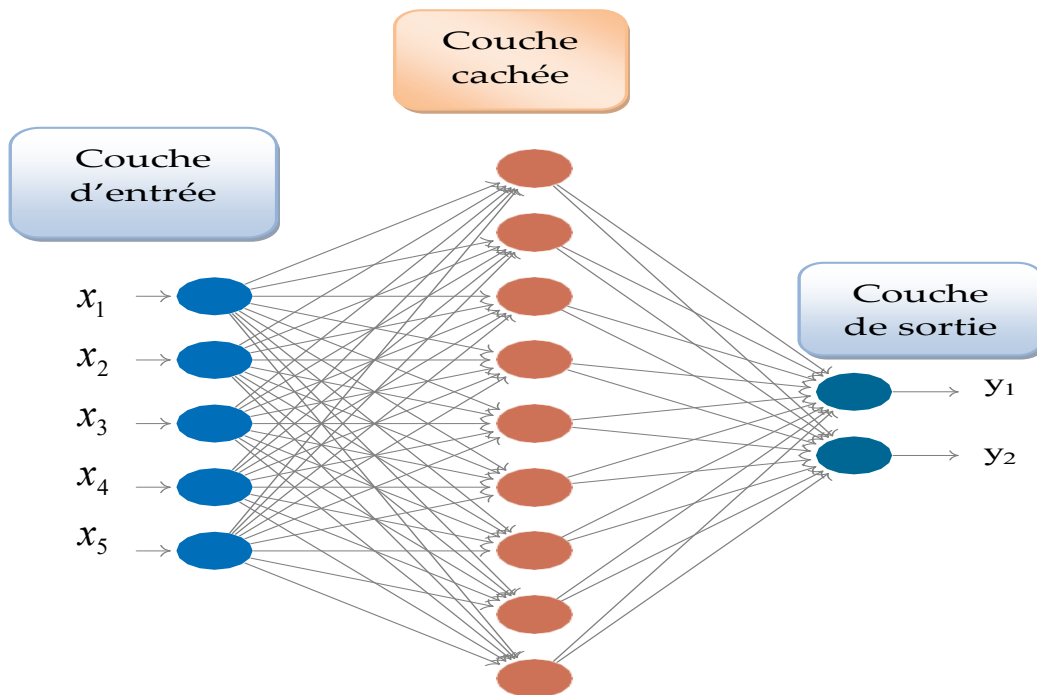


Figure III. 3. Perceptron [61].

Il s'agit du premier système automatique capable d'apprendre de ces expériences grâce à une implémentation du concept d'induction. Le perceptron est un réseau de type feedforward, c'est-à-dire que l'information ne se propage que dans un sens, de la couche d'entrée vers la couche de sortie. Malheureusement, le perceptron ne permet pas de traiter les problèmes non linéaires, ce qui constitue une limitation importante [61].

Le physiologiste canadien Donald Hebb fournit une méthode pour modifier les poids synaptiques appelée règle de Hebb. Il a imaginé un mécanisme qui permet de modifier la valeur des coefficients synaptiques en fonction de l'activité des entités connectées ensemble. Cette méthode de modification des poids du neurone est encore aujourd'hui utilisée couramment [61].

III.2.2. Réseaux multicouches

Dans les années 80, l'apparition de la rétro-propagation du gradient de l'erreur permet la création des perceptrons multicouches. Les réseaux de neurones multicouches ont une topologie en trois parties : une première couche connectée à l'extérieur du réseau, une ou plusieurs couches cachées connectées séquentiellement à partir de la couche d'entrée et ensuite une couche de sortie. On parle de réseau profond lorsque les couches cachées sont nombreuses.

Soit $x = (x_1, x_2, \dots, x_n)$ le vecteur d'entrée d'un perceptron à L couches, exprimons h_1 le vecteur de sortie de la première couche [61].

L'équation des perceptrons multicouches: expression de la sortie de la couche d'entrée :

$$h_1 = \phi(w_{0,1} \cdot x + b_1) \quad (\text{III.3})$$

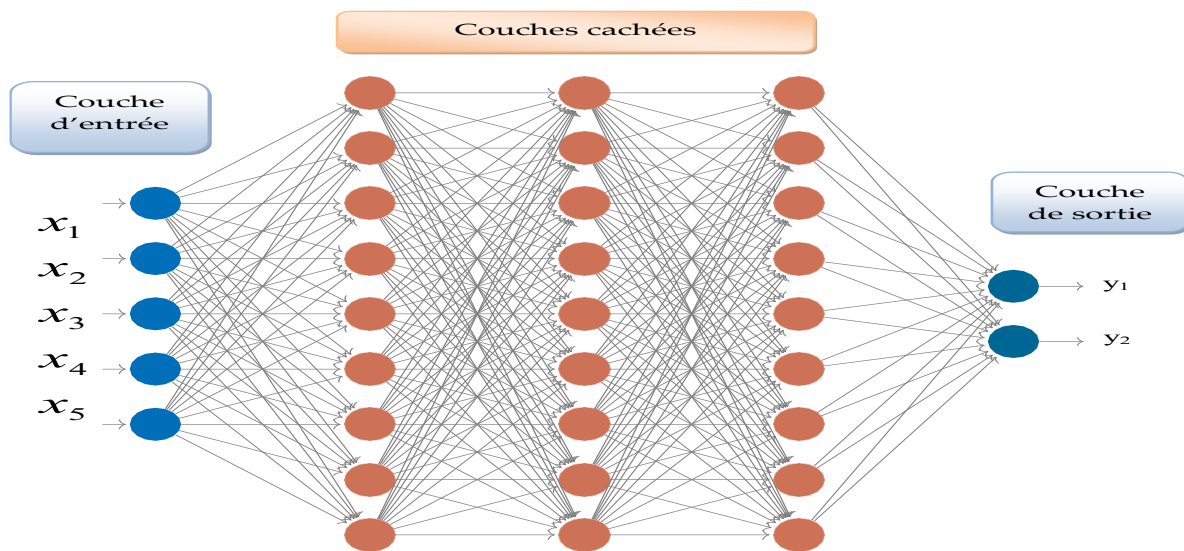


Figure III. 4. Perceptron multicouches.

avec $w_{0,1}$ la matrice des poids de la couche d'entrée en regard aux vecteurs d'entrée, b_1 le biais et ϕ la fonction d'activation. Nous pouvons généraliser cette écriture aux couches suivantes [61].

Équation: Perceptron expression de la sortie de la couche cachée

$$h_{l+1} = \phi(w_{l,l+1} \cdot h_l + b_{l+1}) \quad (\text{III.4})$$

La sortie du réseau sera alors calculable comme suit.

Équation: Perceptron expression de la sortie du perceptron

$$y = h_L = \sigma(w_{L-1,L} \cdot h_{L-1} + b_L) \quad (\text{III.5})$$

Avec σ la fonction d'activation de la couche de sortie, et $y = (y_1, y_2, \dots, y_m)$ le vecteur de sortie du réseau de neurones pour une tâche de classification mono-étiquette de m classes $C = (c_1, c_2, \dots, c_m)$.

La fonction d'activation de la couche de sortie est différente de celle des couches cachées. En effet, le rôle de chaque couche est différent ainsi que son implémentations. La dernière couche pour une tâche de classification permettra la production des probabilités des classes pour les échantillons mis en entrée.

Généralement, pour une classification mono-étiquette, la fonction d'activation de la couche de sortie du réseau est de type *Softmax* [61].

La fonction de coût est définie par l'expression suivante:

$$\text{Softmax} = \frac{e^{z_i}}{\sum_{j=1}^m e^{z_j}} \quad (\text{III.6})$$

avec $z_i = w \cdot h_{i-1} + b$, et y le vecteur de sortie du réseau est alors une variable aléatoire représentant la probabilité du vecteur x d'appartenir à la classe c ; que l'on note $P(y=i|x)$.

La fonction *Softmax*, par sa construction ainsi que la normalisation qu'elle conduit, permet d'obtenir une somme des termes égale à un [61].

Les neurones expriment une distribution de probabilités donnée par:

$$\sum_{i \in 1, 2, \dots, m} P(y=i|x) = 1 \quad (\text{III.7})$$

Avec m le nombre de neurones de la couche de sortie L .

Pratiquement, la prédiction de la classe pour l'échantillon x est obtenue par la valeur maximale parmi les éléments du vecteur y [61].

L'équation d'estimation de l'étiquette est donnée par la relation suivante :

$$\arg \max (Softmax(z_L))_{i \in 1,2,\dots,m} \quad (III.8)$$

III.3. Apprentissage des réseaux de neurones

La propriété la plus importante pour un réseau de neurones, est sûrement sa capacité d'apprendre de son environnement, d'améliorer sa performance à travers un processus d'apprentissage [64].

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle les valeurs des poids synaptiques du réseau sont ajustées pour remplir au mieux la tâche à laquelle le réseau est destiné.

Les méthodes d'apprentissage peuvent être classées en deux catégories principales [64] : l'apprentissage supervisé et l'apprentissage non supervisé. Les principes fondamentaux de chacune de ces approches sont détaillés dans les sous-sections suivantes.

III.3.1. Apprentissage supervisé

L'apprentissage supervisé est utilisé lorsque les couples entrée–sortie désirés sont connus à l'avance. Dans ce type d'apprentissage, le réseau de neurones est entraîné à partir d'un ensemble de données étiquetées, pour lesquelles la sortie attendue est disponible. À chaque itération, la sortie produite par le réseau est comparée à la sortie désirée, et l'erreur correspondante est calculée, comme illustre la Figure III.5. Cette erreur est ensuite utilisée pour ajuster les poids synaptiques du réseau afin d'améliorer progressivement ses performances [64].

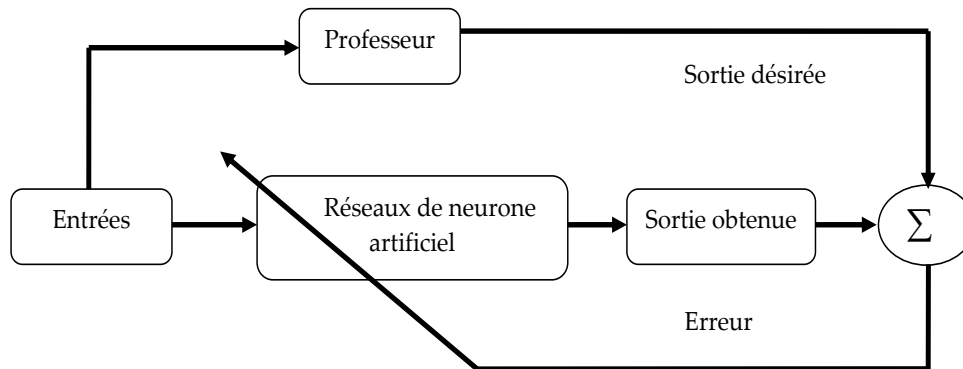


Figure III. 5. Schéma bloc de l'apprentissage supervisé d'un réseau de neurone.

III.3.2. Apprentissage non-supervisé

Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé ne nécessite aucune connaissance a priori des sorties associées aux données d'entrée. Dans ce mode d'apprentissage, seules les données d'entrée sont fournies au réseau, sans étiquettes ni classes de référence. L'algorithme explore alors les données afin d'identifier automatiquement des

structures, des similitudes ou des regroupements pertinents, en ajustant les poids synaptiques de manière à maximiser la qualité de la classification ou du regroupement des données [64].

Comme l'illustre la Figure III.6, seules les entrées sont disponibles durant la phase d'apprentissage, tandis que les sorties sont déterminées automatiquement par le modèle. Une fois cette phase terminée et le réseau entraîné, le modèle peut être utilisé pour prédire ou classer de nouvelles données en fonction des connaissances acquises.

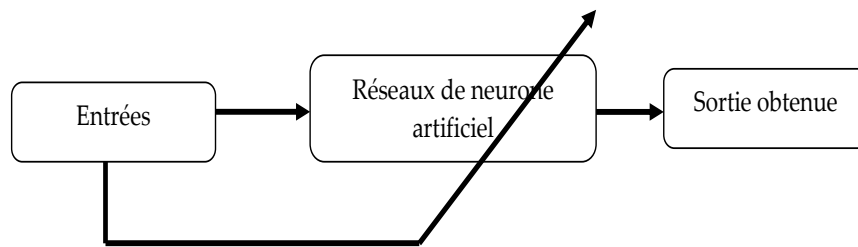


Figure III. 6. Schéma bloc de l'apprentissage non supervisé d'un réseau de neurone.

III.4. Apprentissage profond

L'apprentissage profond «deep learning» est un ensemble de techniques d'apprentissage automatique qui a permis des avancées importantes en intelligence artificielle dans les dernières années.

Dans l'apprentissage automatique, un programme analyse un ensemble de données afin de tirer des règles qui permettront de tirer des conclusions sur de nouvelles données. L'apprentissage profond est basé sur ce qui a été appelé, par analogie, des « réseaux de neurones artificiels », composés de milliers d'unités (les « neurones ») qui effectuent chacune de petites opérations simples. Les résultats d'une première couche de « neurones » servent d'entrée aux calculs d'une deuxième couche et ainsi de suite.

Par exemple, pour la reconnaissance visuelle, des premières couches d'unités identifient des lignes, des courbes, des angles... des couches supérieures identifient des formes, des combinaisons de formes, des objets, des contextes...

Les progrès de l'apprentissage profond ont été possibles notamment grâce à l'augmentation de la puissance des ordinateurs et au développement de grandes bases de données «big data» [65].

III.4.1. Architectures de réseaux de neurones profonds

Il existe un grand nombre d'architectures de réseaux de neurones profonds, dont la plupart sont dérivées ou inspirées de quelques architectures fondamentales. Dans ce mémoire, nous nous intéressons particulièrement aux réseaux de neurones convolutifs (Convolutional Neural

Networks, CNN), qui constituent l'une des architectures les plus performantes pour les tâches de traitement et d'analyse d'images.

Les réseaux de neurones convolutifs sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a deux dimensions pour une image aux niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu]. La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers d'une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions (Figure III.7). Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. En fin, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN [65].

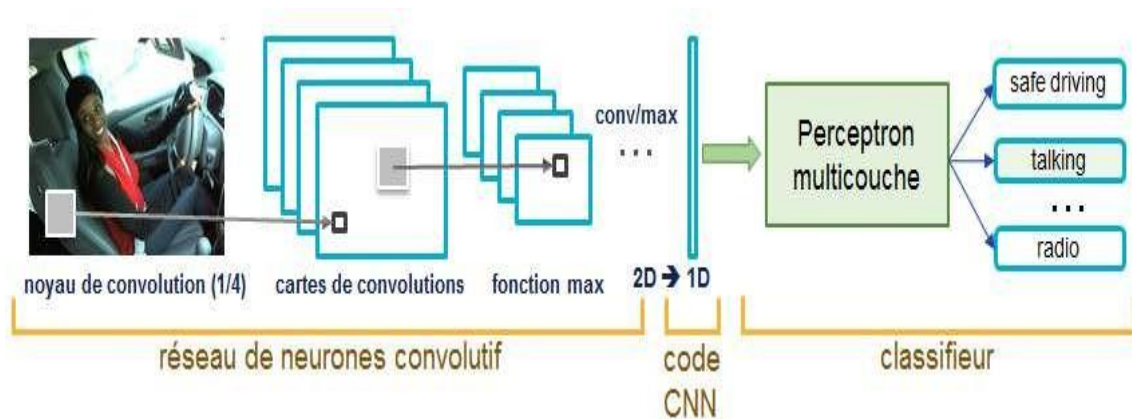


Figure III. 7. Réseaux de neurones convolutifs [65].

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées. Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image. La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories [65].

Ce type d'architecture, les réseaux de neurones convolutifs (CNN), est constitué d'un empilement de couches de traitement successives, chacune jouant un rôle spécifique dans l'extraction et la transformation des caractéristiques. Les sous-sections suivantes présentent ces différentes couches en détail.

III.4.2. Couche de convolution (CONV)

Trois hyper paramètres permettent de dimensionner le volume de la couche de convolution (aussi appelé volume de sortie) : la ‘profondeur’, le ‘pas’ et la ‘marge’.

- a) ‘Profondeur’ de la couche: nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
- b) ‘Le pas’ contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.
- c) ‘La marge (à 0)’ ou ‘zero padding’: parfois, il est commode de mettre des zéros à la frontière du volume d’entrée. La taille de ce ‘zero-padding’ est le troisième hyperparamètres. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d’entrée [66].

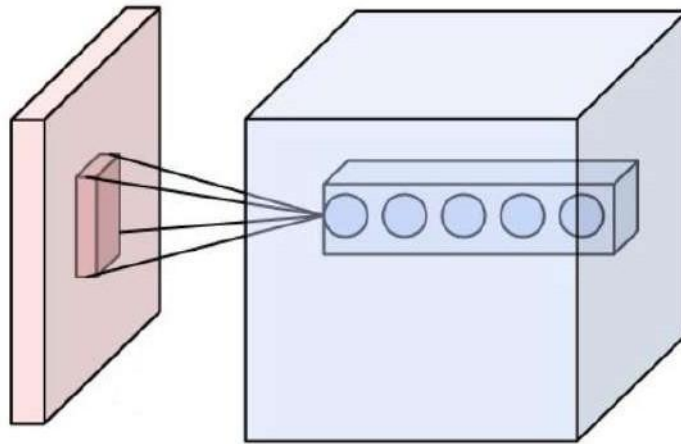


Figure III. 8. Ensemble de neurones (cercles) créant la profondeur d’une couche de convolution (bleu). Ils sont liés à un même champ récepteur (rouge).

III.4.3. Couche de pooling

Un autre concept important des CNNs est le pooling « mise en commun », ce qui est une forme de sous-échantillonnage de l’image. L’image d’entrée est découpée en une série de rectangles de n pixels de côté ne se chevauchant pas (pooling). Chaque rectangle peut être vu comme une tuile. Le signal en sortie de tuile est défini en fonction des valeurs prises par les différents pixels de la tuile.

Le pooling réduit la taille spatiale d’une image intermédiaire, réduisant ainsi la quantité de paramètres et de calcul dans le réseau. Il est donc fréquent d’insérer périodiquement une couche de pooling entre deux couches convolutives successives d’une architecture CNN pour contrôler l’overfitting (sur-apprentissage). L’opération de pooling créait aussi une forme d’invariance par translation [65].

La couche de pooling fonctionne indépendamment sur chaque tranche de profondeur de l'entrée et la redimensionne uniquement au niveau de la surface. La forme la plus courante est une couche de mise en commun avec des tuiles de taille 2x2 (largeur/hauteur) et comme valeur de sortie la valeur maximale en entrée. On parle dans ce cas de « Max- Pool 2x2 » (compression d'un facteur 4) [65].

Il est possible d'utiliser d'autres fonctions de pooling que le maximum. On peut utiliser un «average pooling» (la sortie est la moyenne des valeurs du patch d'entrée), du « L2-norm pooling ». Dans les faits, même si initialement l'average pooling était souvent utilisé il s'est avéré que le max-pooling était plus efficace car celui-ci augmente plus significativement l'importance des activations fortes. En d'autres circonstances, on pourra utiliser un pooling stochastique.

Le pooling permet de gros gains en puissance de calcul. Cependant, en raison de la réduction agressive de la taille de la représentation (et donc de la perte d'information associée), la tendance actuelle est d'utiliser de petits filtres (type 2x2) (Figure III.9). Il est aussi possible d'éviter la couche de pooling mais cela implique un risque sur-apprentissage plus important [67].

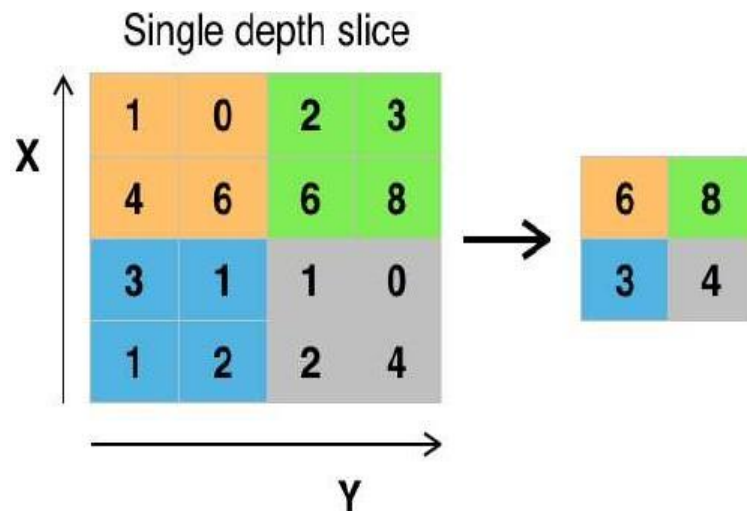


Figure III. 9. Pooling avec un filtre 2x2 et un pas de 2 [65].

III.4.4. Couche entièrement connectée(FC)

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente (comme on le voit régulièrement dans les réseaux réguliers de neurones). Leurs fonctions d'activations peuvent donc être calculées avec une multiplication matricielle suivie d'un décalage de polarisation [65].

III.4.5. Couche de Couche de perte (LOSS)

La fonction de perte est la métrique qui aide un réseau à comprendre s'il apprend dans la bonne direction. Pour encadrer la fonction de perte en termes simples, considérez-la comme le score que vous obtenez lors d'un examen. Supposons qu'un étudiant veut participer à plusieurs tests sur le même sujet : quelle métrique utilisera l'étudiant pour comprendre ses performances à chaque test? De toute évidence, le score du test. Supposons qu'il a obtenu 11, 12, 15, 17 et 19 sur 20 dans cinq tests de langue consécutifs.

Il est clairement que l'amélioration des résultats des tests est une indication de ses performances. Si les résultats des tests avaient diminué, le verdict serait que ses performances diminuent et qu'il aura besoin de changer ses méthodes d'étude ou son matériel pour qu'il s'améliore [68].

En fonction du type de résultat de données, nous avons plusieurs pertes standard fonctions définies dans l'apprentissage par ordinateur. Pour les cas d'utilisation de régression (c'est-à-dire, où la prédiction finale serait un nombre continu comme les notes obtenues par un étudiant, le nombre d'unités de produits vendues par un magasin, le nombre d'appels reçus de clients dans un centre de contact, etc.) [69].

III.4.6. Fonction d'activation

Les fonctions peuvent être de différents types en fonction du problème ou des données. Il existe deux types de fonctions d'activations (linéaire et non-linéaire), dans le mode d'activation linéaire un neurone linéaire prend une combinaison linéaire des entrées pondérées et la sortie peut prendre n'importe quelle valeur entre infini et infini. Dans le mode d'activation non- linéaire est le mode d'activation le plus utilisé, où la sortie se limite entre une certaine gamme. Parmi les fonctions d'activation les plus couramment utilisées, on peut citer :

- Les fonctions de type ReLU (*Rectified Linear Unit*) et ses variantes : Leaky ReLU, PReLU et Swish ;
- La fonction sigmoïde (*Sigmoid*) et la tangente hyperbolique (*Tanh*) ;
- La fonction Softmax, principalement utilisée dans la couche de sortie pour les problèmes de classification multiclasse.

Parmi ces fonctions, la ReLU est souvent privilégiée dans les architectures profondes en raison de sa simplicité de calcul et de sa capacité à accélérer la convergence de l'apprentissage. Elle permet généralement d'obtenir des performances comparables, voire

supérieures, à celles d'autres fonctions d'activation, tout en réduisant le temps d'entraînement du réseau [65].

La fonction de l'unité linéaire rectifiée (ReLU) [70] est l'une des fonctions d'activation non saturées les plus remarquables. Elle est définie comme suit : $f(x) = \max(x, 0)$.

Comme le montre la Figure III.10, la fonction ReLU remplace toutes les valeurs négatives par zéro et conserve les valeurs positives. Par conséquent, les sorties de cette fonction sont comprises entre 0 à l'infini.

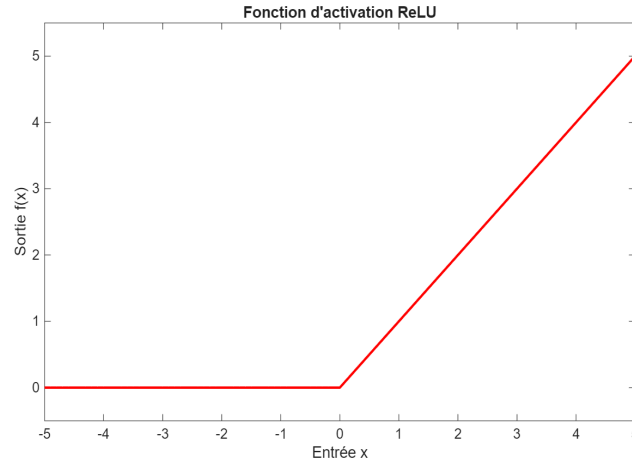


Figure III. 10. Schéma représente la fonction d'activation ReLU.

Un inconvénient potentiel de l'unité ReLU est qu'elle a un gradient nul chaque fois que le neurone n'est pas activé. Cela peut faire en sorte que les neurones qui ne sont pas activés initialement ne soient jamais actifs car l'optimisation basée sur le gradient n'ajustera pas leurs poids. En outre, cela peut ralentir le processus d'apprentissage en raison des gradients zéro constants. Pour pallier ce problème Mass et al. [71], introduire LeakyReLU qui est définie par l'expression suivante et illustrée à la Figure III.11 :

$$a_{i,j,k} = \max(z_{i,j,k}, 0) + \lambda \min(z_{i,j,k}, 0) \quad (\text{III.9})$$

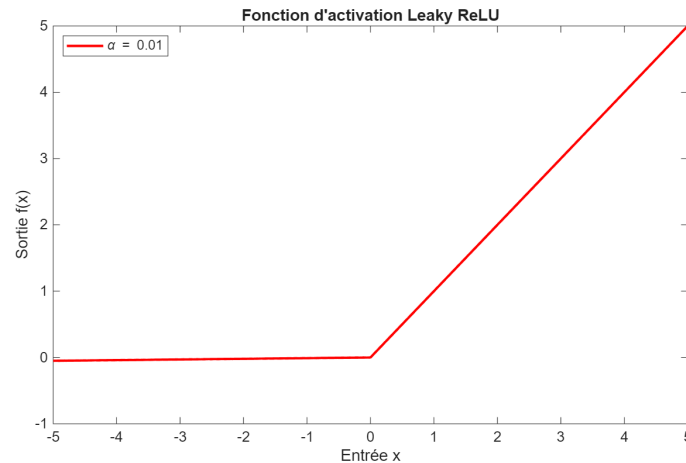


Figure III. 11. Schéma représente la fonction d'activation LeakyRelu

où λ est un paramètre prédéfini dans la plage (0,1). Par rapport à ReLU, Leaky ReLU comprime la partie négative plutôt que de la mapper sur un zéro constant, ce qui permet un petit gradient non nul lorsque le neurone n'est pas activé.

Plutôt que d'utiliser un paramètre prédéfini λ dans LeakyReLU, par exemple. La fonction d'activation de l'unité linéaire rectifiée paramétrique (PReLU) apprend de manière adaptative les paramètres des redresseurs afin d'améliorer la précision [72], et son comportement est illustré à la Figure III.12.

Mathématiquement, la fonction PReLU est définie comme :

$$a_{i,j,k} = \max(z_{i,j,k}, 0) + \lambda_k \min(z_{i,j,k}, 0) \quad (\text{III.10})$$

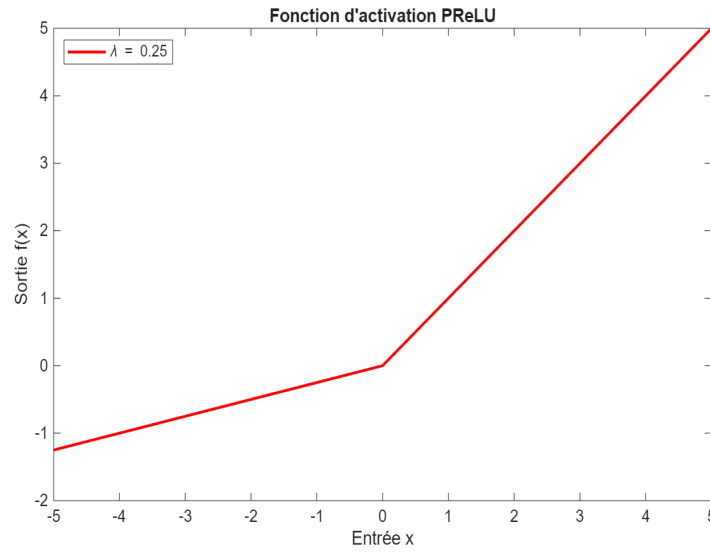


Figure III. 12. Schéma représente la fonction d'activation PRelu.

Où λ_k est le paramètre appris pour le k -ième canal. Comme PReLU n'introduit qu'un très petit nombre de paramètres supplémentaires, par exemple, le nombre de paramètres supplémentaires est le même que le nombre de canaux du réseau, il n'y a pas de risque supplémentaire de sur-apprentissage et le coût de calcul supplémentaire est négligeable. Il peut également être formé simultanément avec d'autres paramètres par rétro-propagation.

La fonction d'activation automatique (SWISH) [73] est en fait une version de la fonction sigmoïde, ça fonction et est définie comme suit:

$$f(x) = x * \sigma(x) \quad (\text{III.11})$$

Où $\sigma(x)$ est le sigmoïde de x donné par : $\sigma(x) = \frac{1}{1 + e^{-x}}$

La fonction d'activation SWISH est une fonction non monotone (Figure III.13).

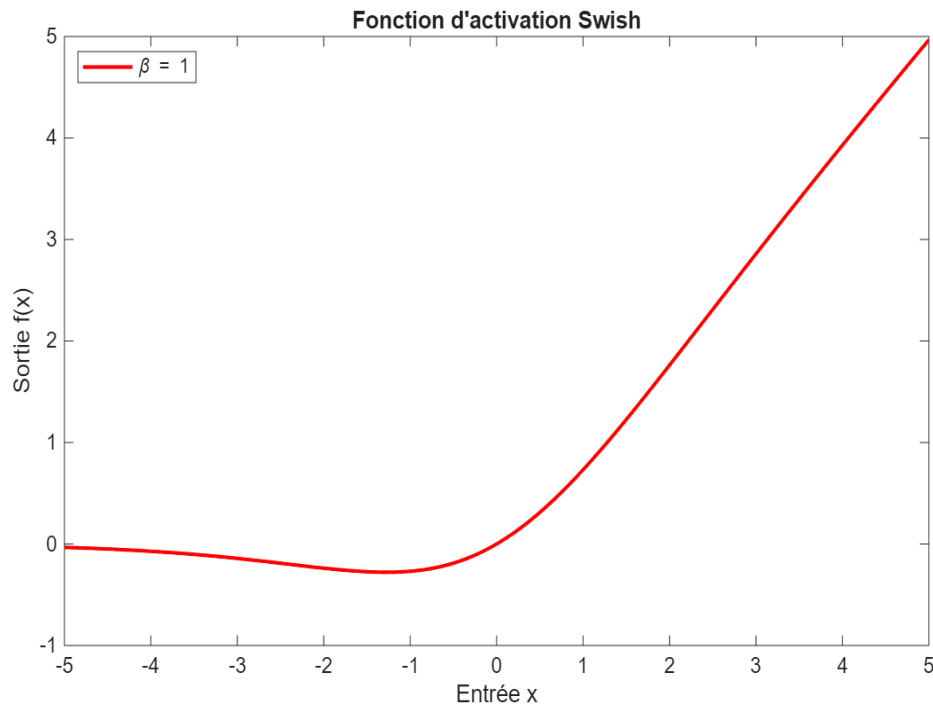


Figure III. 13. Schéma représente la fonction d'activation SWICH.

La fonction d'activation sigmoïde permet de réduire la sortie entre 0 et 1 en appliquant la fonction log, qui rend les problèmes de classification plus facile [74].

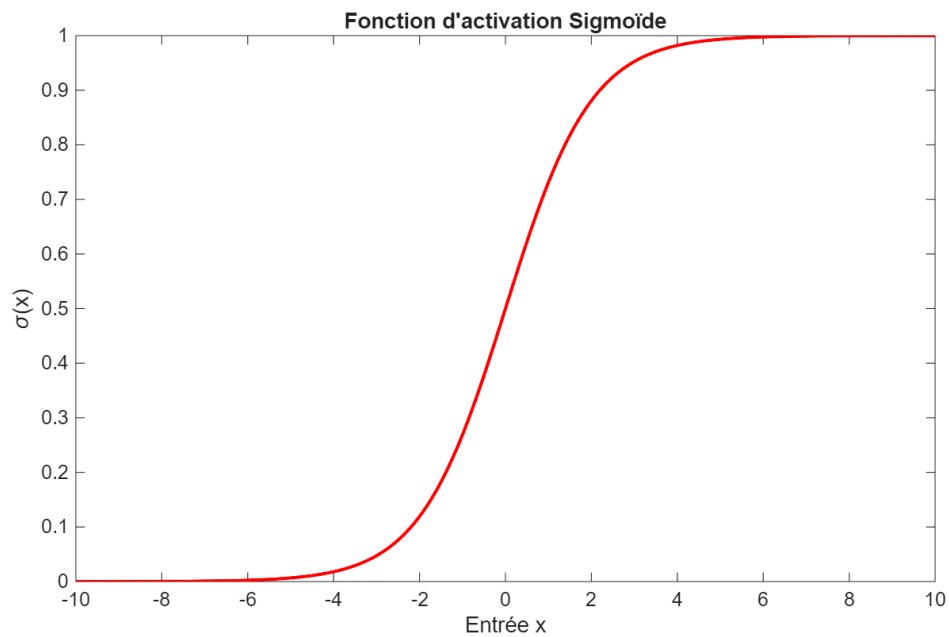


Figure III. 14. Schéma représente la fonction d'activation Sigmoid.

La plage de la fonction d'activation Tanh est de (-1 à 1) [74] comme le montre la Figure III.15.

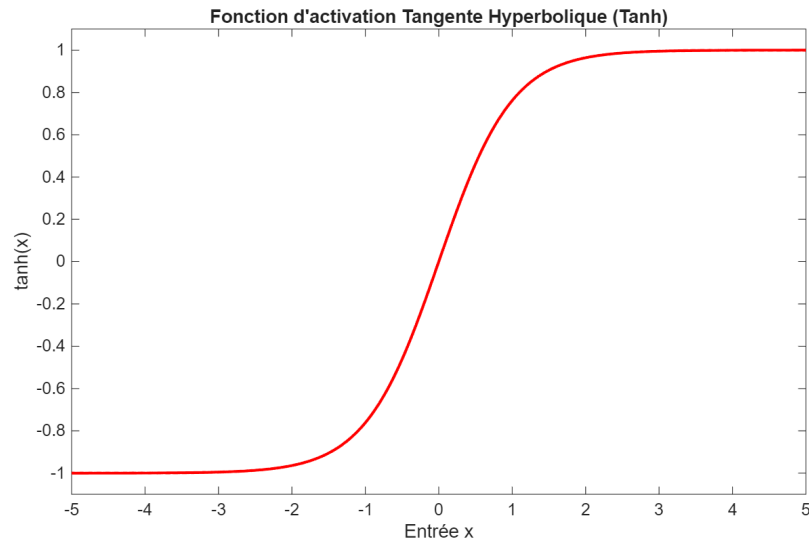


Figure III. 15. Schéma représente la fonction d'activation Tanh.

La fonction Softmax est presque similaire à sigmoïde, mais il calcule les probabilités des événements sur n classes différentes, ce qui sera utile pour déterminer la cible dans la classification multi-classes.

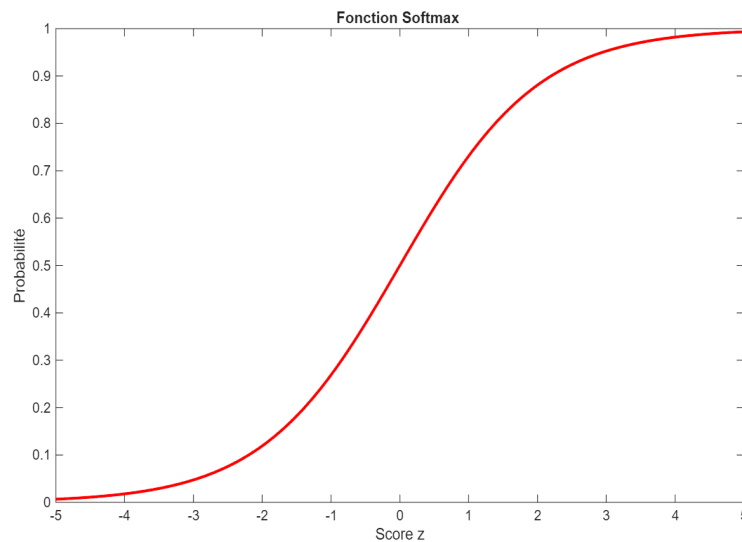


Figure III. 16. Schéma représente la fonction d'activation Softmax.

III.5. Optimisation

La recherche d'un minimum local d'une fonction de coût nécessite généralement l'utilisation d'un algorithme d'optimisation itératif. Le principe consiste à mettre à jour les paramètres du modèle par des déplacements successifs dans la direction opposée au gradient de la fonction de coût (ou à une approximation de celui-ci), cette direction correspondant à la plus forte diminution de l'erreur. Cette méthode, connue sous le nom de descente de gradient (Gradient Descent), permet de converger progressivement vers un minimum local de la fonction objectif [75].

À l'inverse, un déplacement dans le sens du gradient entraîne une augmentation de la valeur de la fonction. Cette approche, appelée ascension de gradient (Gradient Ascent), est utilisée lorsque l'objectif est de maximiser une fonction plutôt que de la minimiser [76].

III.5.1. Descente de gradient (GD)

Il existe trois variantes de descente de gradient, qui varient dans la quantité de données que nous utilisons pour évaluer le gradient de la fonction objectif [77]. En fonction de la quantité de données [78], nous avons un compromis entre l'exactitude du paramètre de mise à jour et le temps d'exécution d'une mise à jour.

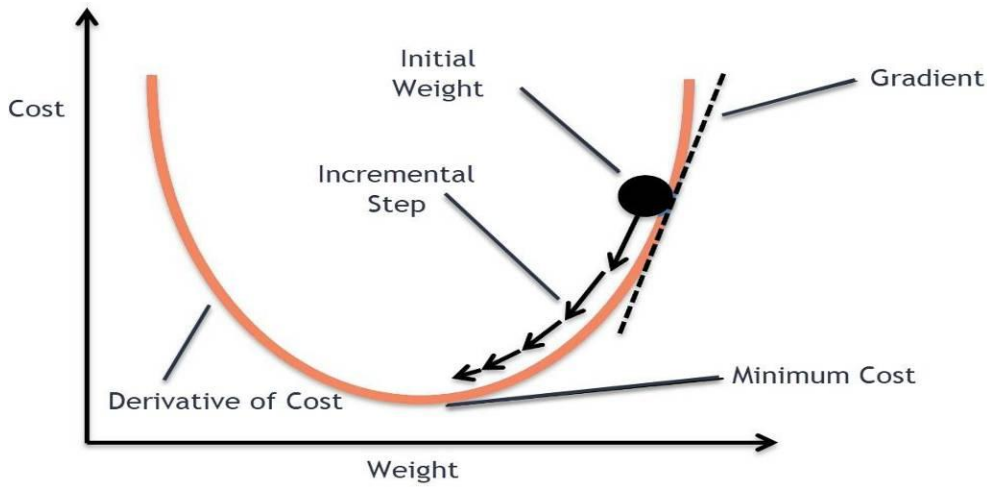


Figure III. 17. Algorithme de descente de gradient [79].

III.5.2. Descente de gradient par lots (BGD)

La descente de gradient par lots, parfois nommé vanilla GD, détecte l'erreur pour chaque individu de la base d'apprentissage. Mais les poids du modèle sont révisés après que tous les individus de cette base ont été évalués. Tout ce processus est comme une boucle, et c'est ce qu'on appelle une époque d'entraînement [80]. Certains des avantages de l'optimisation avec descente de gradient par lots sont son calcul efficace, un gradient d'erreur équilibré et une convergence stable [81]. Certaines faiblesses sont que le gradient d'erreur constant peut souvent conduire à une condition de convergence qui n'est pas la meilleure que le modèle puisse accomplir. Cela exige également que les éléments de la base d'apprentissage soit en mémoire et disponible pour l'algorithme[82]. BGD offre les performances les plus précises mais nécessite de nombreuses analyses complètes coûteuses des données réelles [83].

L'optimisation par la descente de gradient par lots calcule le gradient de la fonction d'erreur [84] concernant les paramètres θ pour la base d'apprentissage, comme le montre l'équation suivante :

$$\theta_{\text{nouveau}} = \theta_{\text{ancien}} - \eta \cdot \Delta_{\theta} J(\theta) \quad (\text{III.12})$$

où:

θ_{nouveau} = position suivante

θ_{ancien} = position actuelle

η = taux d'apprentissage (taille du pas)

$\Delta_{\theta} J(\theta)$ = Direction de l'augmentation la plus rapide

III.5.3. Descente de gradient stochastique (SGD)

Les algorithmes SGD se sont avérés efficaces dans l'optimisation des modèles d'apprentissage à grande échelle [85]. Le terme stochastique désigne un mécanisme ou une méthode liée à une possibilité aléatoire ; par conséquent, au lieu de passer toute la base d'apprentissage à chaque itération pour mettre à jour les poids du modèle, seulement quelques échantillons sont choisis [73]. SGD vise à trouver le minimum global en modifiant la structure du réseau après chaque étape d'apprentissage [86]. Cette approche réduit simplement l'erreur en se rapprochant du gradient pour un lot choisi au hasard au lieu de trouver le gradient de la base d'apprentissage. En réalité, l'échantillonnage aléatoire est effectué en mélangeant au hasard la base d'apprentissage et en se déplaçant pas à pas dans les lots de la fonction objective[87], comme le montre la Figure III.18.

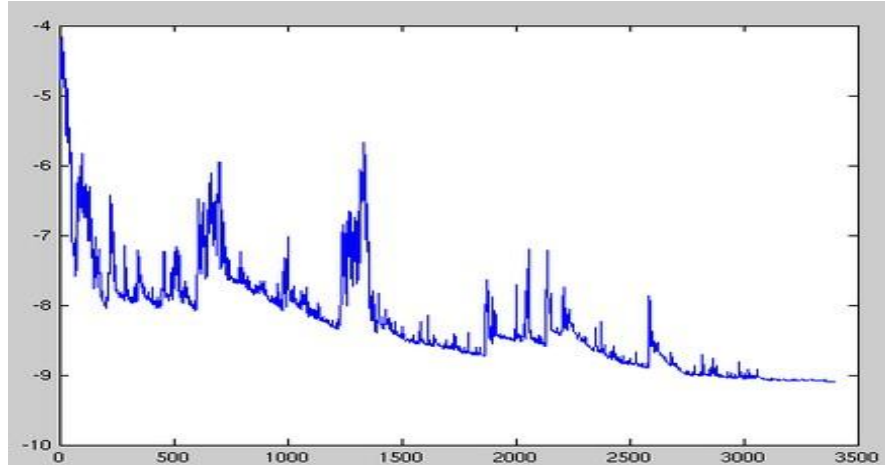


Figure III. 18. Descente de gradient stochastique [78].

III.5.3. Descente de gradient stochastique (SGD)

Une autre variante des principes de SGD et BGD est le mini-batch GD. Il divise la base d'apprentissage en lots plus petits et effectue une mise à jour pour chacun de ces lots. Cela fournit un équilibre entre la robustesse du SGD et l'efficacité du GD par lots [88].

Pour plusieurs problèmes d'optimisation, ces trois algorithmes GD (Batch GD, SGD et Mini-batch GD) fonctionnent bien et peuvent tous converger vers un optimum prometteur (local ou

global), mais ils souffrent toujours de nombreux problèmes tels que ceux mentionnés ci-dessous :

Le taux d'apprentissage η peut avoir un effet majeur sur la convergence des algorithmes GD [89]. Il existe un compromis entre le taux de convergence et le dépassement lors de la définition d'un taux d'apprentissage. Si le taux d'apprentissage est trop élevé, nous pourrions dépasser le minimum et commencer à rebondir sans atteindre le minimum. En revanche, si le taux d'apprentissage est trop faible, la formation peut être trop longue [87].

Dans la plupart des cas, pour les algorithmes GD, un taux d'apprentissage fixe ne fonctionne pas bien dans le processus de mise à jour des poids [90]. L'algorithme nécessitera un taux d'apprentissage plus élevé dans les étapes initiales pour obtenir un optimum rapide réussi (local ou global). Dans les dernières étapes cependant, l'algorithme devra modifier le taux d'apprentissage [87].

Pour différentes variables, au lieu du processus de mise à jour, leur mise à niveau peut simplement impliquer un taux d'apprentissage différent. Par conséquent, il est nécessaire et approprié d'utiliser un taux d'apprentissage individuel pour diverses variables [87].

Un autre défi majeur est d'éviter d'être coincé dans leurs multiples minima locaux sous-optimaux en minimisant les fonctions d'erreur extrêmement non convexes communes aux réseaux de neurones [78]. Les minima locaux sont le plus grand défi pour analyser la convergence [91] comme la Figure III.19.

Les algorithmes d'optimisation constituent la base sur laquelle une machine peut bénéficier de sa pratique [92]. Ils mesurent des gradients et tentent de minimiser la fonction des coûts.

L'apprentissage peut être mis en œuvre de plusieurs manières avec différents types de méthodes d'optimisation [93].

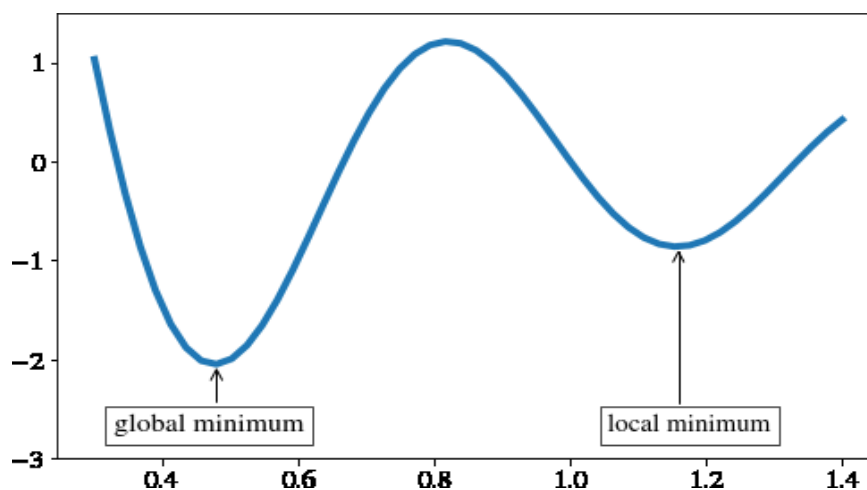


Figure III. 19. Minima globaux Vs. Minima locaux [79].

III.5.3.1. Momentum

Une technique d'optimisation courante est SGD, mais lors de l'apprentissage du modèle, le temps d'exécution est relativement élevé. Momentum est conçu pour apprendre rapidement, en particulier face à de larges courbures, des gradients petits mais bruyants ou des gradients stables. L'utilisation d'un terme de quantité de mouvement est une autre approche qui peut aider le réseau à se débarrasser des minima locaux [94]. C'est peut-être l'extension la plus courante de l'algorithme de rétropropagation. D'autres cas dans lesquels cette approche n'est pas utilisée sont difficiles à trouver. Le gradient ne pointe pas vers le minimum à des points spécifiques de la surface, et les pas successifs de GD oscilleront d'un côté à l'autre, n'avançant que très lentement vers le minimum [95].

III.5.3.2. Nesterov Momentum

Sur la base de paramètres qui estiment des positions potentielles plutôt que des paramètres actuels, le gradient est calculé. Nesterov Momentum est une amélioration par rapport à l'élan et ne décide pas de l'emplacement futur du paramètre [96]. Le Nesterov est une version de l'algorithme de Momentum inspiré de la méthode du gradient accéléré de Nesterov. La distinction entre cette approche et la méthode de l'impulsion est que la vitesse est déjà ajoutée aux paramètres lors du calcul du gradient dans la méthode Nesterov. Cela peut être considéré comme une tentative d'incorporer un facteur correctif à la forme de quantité de mouvement conventionnelle [97].

III.5.4. Descente de gradient adaptative (AdaGrad)

C'est une méthode qui sélectionne le rythme d'apprentissage en fonction de la situation [98]. Comme le taux réel est calculé à partir de paramètres, les taux d'apprentissage ont tendance à s'adapter. Un gradient de paramètre plus élevé aura un taux d'apprentissage réduit et vice versa [99]. La théorie d'AdaGrad est similaire à l'algorithme AdaDelta en ce sens qu'elle mesure différents taux d'apprentissage pour d'autres éléments de paramètres. Pourtant, il utilise l'agrégation des carrés de gradient : contrairement à AdaDelta, il utilise la moyenne mobile des carrés de gradient [100].

III.5.5. Delta adaptatif (AdaDelta)

AdaDelta est l'extension AdaGrad. AdaDelta fonctionne en utilisant plusieurs fenêtres de taille fixe au lieu d'accumuler les dégradés. Il ne surveillera que les gradients disponibles à l'intérieur de la fenêtre [101]. Étant donné que l'algorithme SGD nécessite une sélection manuelle du taux d'apprentissage, le taux d'apprentissage inapproprié choisi conduirait à une faible précision de prédiction. Cependant, en tant qu'optimisation de l'algorithme SGD,

l'Adadelta, connu sous le nom d'algorithme de taux d'apprentissage adaptatif (LR adaptatif), peut ajuster automatiquement le taux d'apprentissage et augmenter la précision de la prédiction [102].

III.5.6. Propagation racine carrée moyenne (RMSProp)

L'un des algorithmes stochastiques adaptatifs les plus courants pour l'apprentissage profond (Deep Neural Network ou DNN) est RMSProp [103]. RMSProp modifie Adagrad de manière à accumuler le gradient. Les dégradés s'agrègent en une moyenne pondérée de manière exponentielle. RMSProp rejette le passé et ne préserve que les connaissances actuelles sur le gradient [104].

III.5.7. Estimation adaptative du moment (Adam)

Adam est une méthode d'optimisation SGD qui mesure des taux d'apprentissage adaptables pour chaque paramètre. Adam est l'une des stratégies d'optimisation la plus courante dans le domaine des réseaux de neurones. Le nom a été tiré d'Adaptive Moments. C'est un mélange de RMSProp et de Momentum. L'opération de mise à niveau considère la variante de gradient lisse et fournit un mécanisme de correction de biais. Adam réduit les coûts de calcul, nécessite moins de mémoire d'exécution et est invariant au redimensionnement diagonal du gradient [105].

RMSprop est un optimiseur basé sur le gradient qui utilise un taux d'apprentissage adaptatif (LR adaptatif) qui varie dans le temps au lieu de traiter le taux d'apprentissage comme un hyperparamètre [106].

III.5.8. Estimation du moment adaptatif maximal (AdaMax)

AdaMax est une forme de SGD avec taux d'apprentissage adaptatif, et une version Adam basée sur la norme de l'infini. AdaMax présente l'avantage majeur d'être beaucoup moins sensible à l'option des hyperparamètres par rapport au SGD [107]. L'équation AdaMax utilise la valeur complète de la deuxième composante de quantité de mouvement de la méthode d'estimation ADAM. Cela offre une solution plus stable [108].

III.5.9. Estimation du moment adaptatif accélérée par Nesterov (Nadam)

Cette technique est une combinaison d'approches d'optimisation Adam et RMSProp. Elle a été conçue de manière similaire à l'approche d'optimisation d'Adam. Cependant, elle a été remplacée par l'élan de Nesterov. Cela se traduit par une augmentation marquée de la surperformance du Momentum [109].

III.6. Choix des hyperparamètres

Les CNNs utilisent plus d'hyperparamètres qu'un réseau de neurones multi couches MLP standard. Même si les règles habituelles pour les taux d'apprentissage et des constantes de régularisation s'appliquent toujours, il faut prendre en considération les notions de nombre de filtres, et leur forme.

III.6.1. Nombre de filtres

Comme la taille des images intermédiaires diminue avec la profondeur du traitement, les couches proches de l'entrée ont tendance à avoir moins de filtres tandis que les couches plus proches de la sortie peuvent en avoir davantage. Pour égaliser le calcul à chaque couche, le produit du nombre de caractéristiques et le nombre de pixels traités est généralement choisi pour être à peu près constant à travers les couches. Pour préserver l'information en entrée, il faudrait maintenir le nombre de sorties intermédiaires (nombre d'images intermédiaire multiplié par le nombre de positions de pixel) pour être croissante (au sens large) d'une couche à l'autre.

Le nombre d'images intermédiaires contrôle directement la puissance du système, dépend du nombre d'exemples disponibles et la complexité du traitement.

III.6.2. Forme de filtres

Les formes de filtre varient grandement dans la littérature. Ils sont généralement choisis en fonction de l'ensemble de données. Les meilleurs résultats sur les images de MNIST (28x28) sont habituellement dans la gamme de 5x5 sur la première couche, tandis que les ensembles de données d'images naturelles (souvent avec des centaines de pixels dans chaque dimension) ont tendance à utiliser de plus grands filtres de première couche de 12x12, voire 15x15. Le défi est donc de trouver le bon niveau de granularité de manière à créer des abstractions à l'échelle appropriée et adaptée à chaque cas [65].

III.7. Avantages et les inconvénients Les réseaux de neurones CNNs

Les avantages des réseaux à convolution par rapport aux méthodes de classifications traditionnelles peuvent être résumés comme suit :

- Représentation hiérarchique des caractéristiques, qui correspond aux représentations multi niveaux du pixel aux caractéristiques sémantiques de haut niveau apprises par une structure hiérarchique à plusieurs étapes [110].
- L'architecture des réseaux CNN offre la possibilité d'optimiser conjointement plusieurs tâches connexes (par exemple, Fast RCNN combine la classification et la régression de la boîte englobant dans un mode d'apprentissage multitâches).

- Bénéficiant de la grande capacité d'apprentissage des CNN, certains défis classiques de la vision par ordinateur peuvent être reformulé en tant que problèmes de transformation de données de grande dimension et résolu d'un point de vue différent.
- En raison de ces avantages, les réseaux à convolution ont été largement appliqués dans de nombreux domaines de recherche, tels que la super-résolution d'images, reconstruction, classification d'images, récupération d'images, reconnaissance faciale, détection et analyse vidéo [111].

Un avantage majeur des réseaux convolutifs est l'utilisation d'un poids unique associé aux signaux entrant dans tous les neurones d'un même noyau de convolution. Cette méthode réduit l'empreinte mémoire, améliore les performances et permet une invariance du traitement par translation.

Comparés à d'autres algorithmes de classification de l'image, les réseaux de neurones convolutifs utilisent relativement peu de prétraitement. Cela signifie que le réseau est responsable de faire évoluer tout seul ses propres filtres (apprentissage sans supervision), ce qui n'est pas le cas d'autres algorithmes plus traditionnels. L'absence de paramétrage initial et d'intervention humaine est un atout majeur des CNN [65].

Parmi les inconvénients, la difficulté d'interpréter le comportement d'un réseau de neurones est un inconvénient pour la mise au point d'une application. Il est souvent impossible d'utiliser les résultats obtenus pour améliorer ce comportement. Il est également hasardeux de généraliser à partir d'expériences antérieures et de conclure ou de créer des règles sur le fonctionnement et le comportement des réseaux de neurones [112].

III.8. Conclusion

L'apprentissage profond, et plus particulièrement les réseaux de neurones convolutifs (CNN), offre des perspectives prometteuses pour le débruitage des images SAR. Grâce à leur capacité à extraire automatiquement des caractéristiques pertinentes et à modéliser des relations complexes au sein des données, ces méthodes surpassent généralement les approches classiques en termes de performances et de robustesse. Les concepts théoriques présentés dans ce chapitre constituent ainsi une base solide pour le développement de solutions avancées de réduction du bruit speckle. Leur mise en œuvre ainsi que leur évaluation seront détaillées dans le chapitre suivant consacré aux résultats expérimentaux.

Chapitre IV

Résultats et Discussion

IV.1. Introduction

Ce chapitre est consacré à l'étude et à l'analyse des techniques avancées de traitement des images SAR (Synthetic Aperture Radar), à travers l'application des méthodes d'apprentissage profond (deep learning) basées sur les réseaux de neurones convolutifs, dans le but de réduire le bruit speckle (Speckle Noise). Les images SAR jouent un rôle essentiel dans plusieurs domaines tels que la télédétection, la surveillance environnementale et la gestion des catastrophes. Cependant, elles sont fortement affectées par le bruit speckle, qui est un bruit multiplicatif dégradant la qualité visuelle des images et compliquant leur interprétation.

Dans ce contexte, nous proposons une méthodologie complète couvrant les différentes étapes du processus de débruitage, depuis la préparation et le prétraitement des données, jusqu'à l'application de modèles avancés pour la suppression du bruit. Par ailleurs, des méthodes classiques de filtrage, telles que les filtres de Lee, Kuan et Frost, sont utilisées afin d'établir une comparaison objective avec la méthode proposée.

Nous nous intéressons particulièrement aux réseaux de neurones convolutifs (CNN), qui constituent l'une des approches les plus performantes en traitement d'images, grâce à leur capacité à extraire efficacement les caractéristiques spatiales tout en préservant les détails importants.

L'ensemble de cette étude a été réalisé à l'aide d'outils open-source, notamment le langage Python et des bibliothèques telles que TensorFlow et Keras. De plus, la plateforme Google Colab a été utilisée afin de bénéficier de ressources matérielles performantes (GPU), nécessaires à l'entraînement des modèles de deep learning.

Ce chapitre est structuré comme suit : nous commençons par la description des bases de données utilisées, puis nous présentons les métriques d'évaluation adoptées, telles que le PSNR, le SSIM et le UQI. Ensuite, nous détaillons les méthodes classiques de filtrage, suivies de la présentation de la méthode proposée basée sur les CNN. Enfin, une étude comparative est réalisée, accompagnée d'une discussion des résultats obtenus, avant de conclure le chapitre.

IV.2. Constitution des bases de données

La constitution des bases de données représente une étape essentielle dans ce travail, puisqu'elle vise à fournir des données adaptées à l'entraînement, à la validation et à l'évaluation du modèle proposé pour le débruitage des images SAR et la réduction du bruit speckle. En effet, les images SAR sont intrinsèquement affectées par le bruit speckle, un bruit multiplicatif qui dégrade la qualité visuelle de l'image et complique les tâches d'analyse,

d'interprétation et de traitement. Par conséquent, l'utilisation directe de ces images comme données de référence n'est pas appropriée dans le cadre de l'apprentissage supervisé. Un exemple illustrant l'impact de ce bruit sur la qualité des images SAR est présenté dans la Figure IV.1.

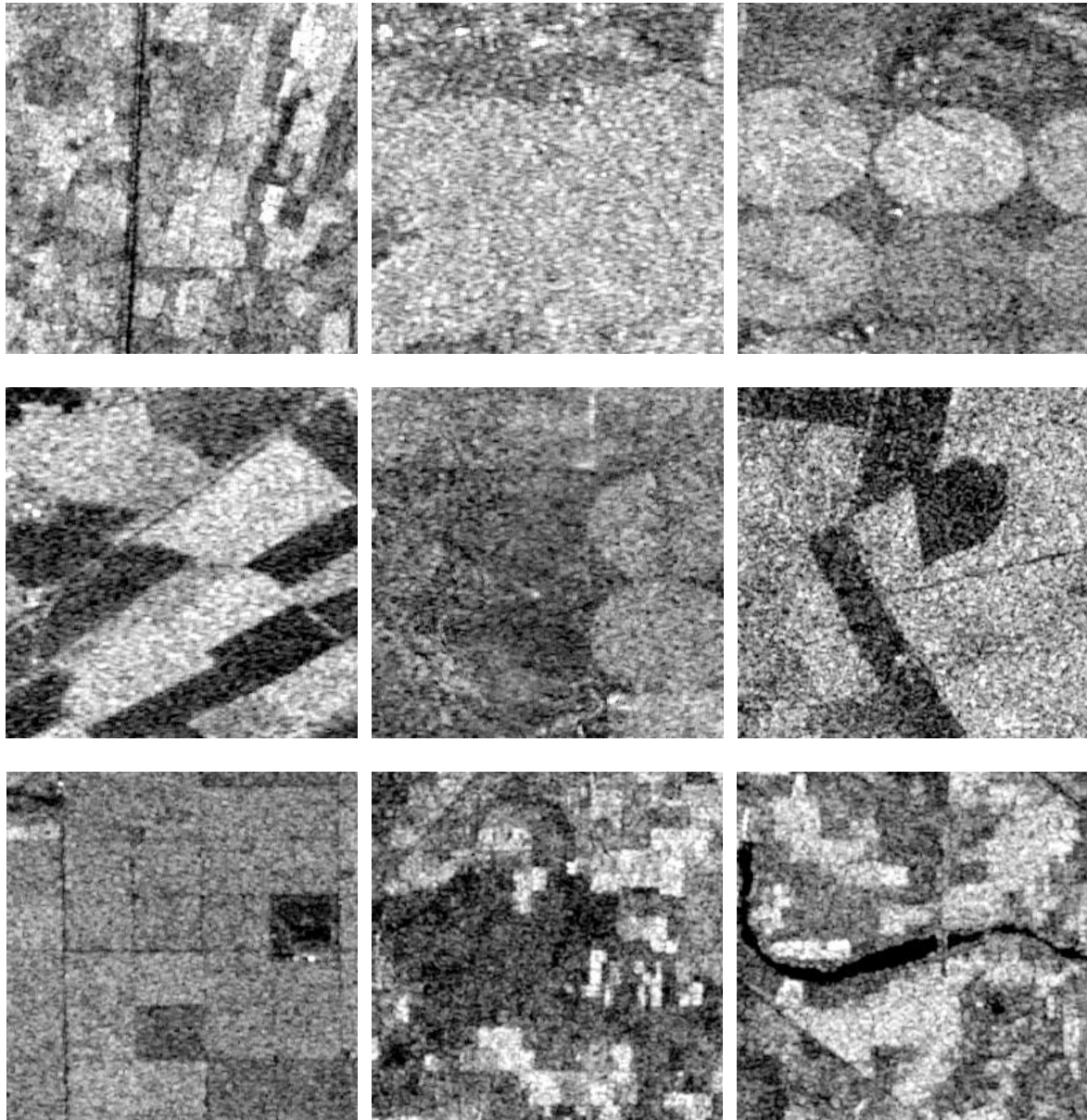


Figure IV. 1. Échantillons d'images SAR affectées par le bruit Speckle.

Dans ce contexte, plusieurs sources de données ont été exploitées suivant une méthodologie structurée afin de garantir à la fois la diversité, la représentativité et la qualité des données utilisées. Les données géospatiales ont ainsi été acquises via la plateforme Google Earth Engine (GEE), un environnement cloud puissant offrant un accès à une vaste archive d'images satellitaires et de données géographiques.

Parmi ces différentes sources, les images Sentinel-2 ont été particulièrement retenues en raison de leur haute résolution spatiale, de leur large couverture spectrale et de leur libre accessibilité. Ces caractéristiques les rendent particulièrement adaptées aux tâches d'analyse, de traitement d'images et de constitution de jeux de données pour les applications de télédétection. Leur richesse spectrale et leur diversité de scènes permettent également de générer des données représentatives de différents environnements géographiques, renforçant ainsi la robustesse et la capacité de généralisation du modèle proposé. Par la suite, des échantillons de bruit speckle ont été générés en considérant différents niveaux de bruit, afin de simuler diverses conditions d'acquisition des images SAR et d'évaluer la performance du modèle dans plusieurs scénarios de dégradation.

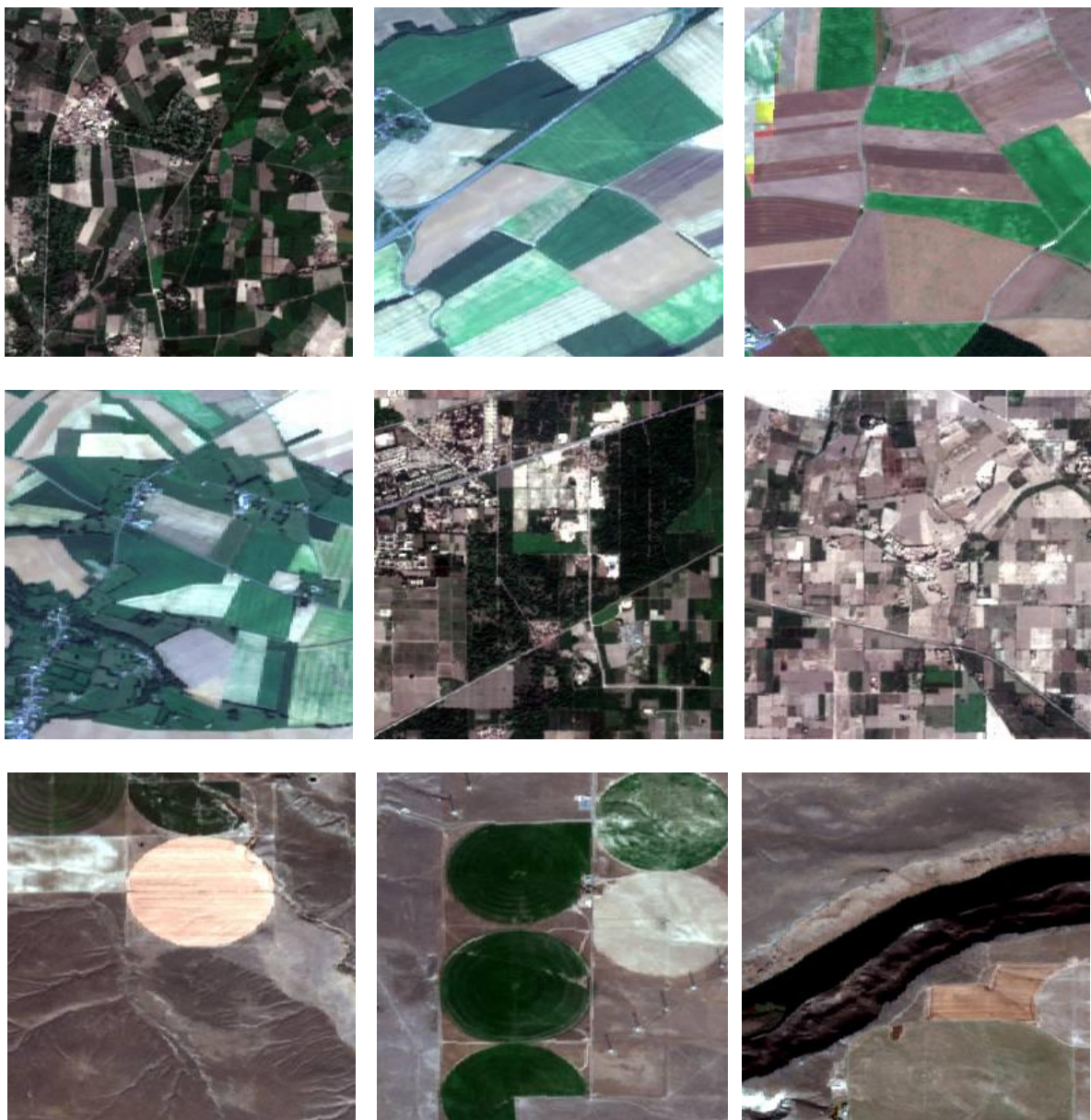


Figure IV. 2. Échantillons d'images de couverture terrestre issues de Sentinel-2.

En outre, la zone d'étude a été sélectionnée dans la région de Saguenay, en raison de sa diversité géographique et de la présence de plusieurs types de couverture terrestre tels que les forêts, les plans d'eau et les terres agricoles. Cette diversité permet d'améliorer la capacité de généralisation du modèle.

Ainsi, ces données ont permis de constituer une base de données complète contenant des images représentant différents types de couverture terrestre, lesquelles seront utilisées ultérieurement dans les phases d'entraînement et de test du modèle proposé [113]. La Figure IV.2 illustre quelques exemples des images utilisées dans ce travail.

IV.2.1. Images naturelles pour l'apprentissage

Les images naturelles constituent la base fondamentale de la phase d'apprentissage du modèle proposé. Elles représentent les données de référence (*images propres* ou *clean images*), permettant au réseau de neurones convolutif (CNN) d'apprendre les caractéristiques structurelles et texturales des scènes avant l'introduction du bruit.

Les images sont organisées en trois ensembles distincts : apprentissage, validation, et test. Elles sont chargées à l'aide de l'outil ImageDataGenerator de TensorFlow qui permet un chargement dynamique des données sans saturer la mémoire.

Un prétraitement est appliqué aux images avant leur utilisation. Celui-ci comprend :

- ✚ Le redimensionnement à une taille fixe de (256×256) pixels, afin d'assurer la compatibilité avec l'entrée du réseau.
- ✚ La conversion en niveaux de gris, permettant de réduire la complexité computationnelle tout en conservant les informations structurelles essentielles.
- ✚ La normalisation des images SAR, permettant d'harmoniser les valeurs des pixels et d'améliorer la stabilité ainsi que les performances du modèle lors de l'apprentissage.

Pour la sélection des échantillons des images destinés aux phases d'apprentissage, de validation et de test, une fonction a été implémentée en Python afin d'automatiser le processus de partitionnement du jeu de données.

Dans cette étude, le nombre total d'images a été limité à 1000 images, afin de maîtriser le temps d'entraînement et les ressources de calcul. Ces images sont réparties selon les proportions: 70 % pour l'apprentissage (700 images), 15 % pour la validation (150 images), 15 % pour le test (150 images).

La fonction développée permet de mélanger aléatoirement les images afin d'éviter tout biais dans la sélection, puis de les diviser en trois sous-ensembles distincts. Le code correspondant est présenté ci-dessous :

```
def split_data(DATA_GEN_INPUT, Ntrain=0.7, Nval=0.15, Nsample=1000):
    dst = os.path.join(DATA_GEN_INPUT, "DATA")
    all_files = [f for f in os.listdir(dst) if f.lower().endswith(".png")] # Liste toutes les images
    random.shuffle(all_files) # Mélange
    if Nsample is not None: # Sélection de NSample images max
        all_files = all_files[:min(Nsample, len(all_files))]
    n_total = len(all_files)
    n_train = int(Ntrain * n_total)
    n_val = int(Nval * n_total)
    train_files = all_files[:n_train]
    val_files = all_files[n_train:n_train + n_val]
    test_files = all_files[n_train + n_val:]
```

Cette approche garantit une répartition équilibrée et aléatoire des données, contribuant ainsi à une évaluation fiable des performances du modèle.

IV.2.2. Génération synthétique du bruit speckle

Les images SAR sont souvent contaminées par un bruit multiplicatif connu sous le nom de speckle. Ce bruit est causé par les interférences constructives et destructives des ondes cohérentes rétrodiffusées par de petits réflecteurs à l'intérieur de chaque cellule de résolution. Le bruit speckle est un bruit multiplicatif intrinsèque aux images SAR, qui dégrade la qualité visuelle et rend l'interprétation des scènes plus difficile.

Dans cette étude, ce bruit est généré de manière artificielle afin de simuler les conditions réelles des images radar. Le bruit est modélisé à l'aide d'une distribution de Gamma, comme indiqué dans le code suivant :

```
noise = np.random.gamma(shape=L, scale=1.0/L, size=img.shape)
noisy = img * noise
```

Plus précisément, le bruit généré suit une distribution Gamma de moyenne unitaire $E[n] = 1$ et de variance $\text{var}[n] = 1/L$, où L représente le nombre de *looks*. La fonction de densité de probabilité associée est donnée par :

$$p(F) = \frac{1}{\Gamma(L)} L^L F^{L-1} e^{-LF} \quad (\text{IV.1})$$

Le bruit est ensuite appliqué à l'image propre selon un modèle multiplicatif défini par :

$$L_{\text{bruitée}}(x, y) = I_{\text{propre}}(x, y) \times n(x, y) \quad (\text{IV.2})$$

où $I_{\text{propre}}(x, y)$ désigne l'image originale, $n(x, y)$ le bruit speckle généré suivant une loi Gamma, et $I_{\text{bruitée}}(x, y)$ l'image bruitée obtenue. Ce processus est illustré dans la Figure IV.3.

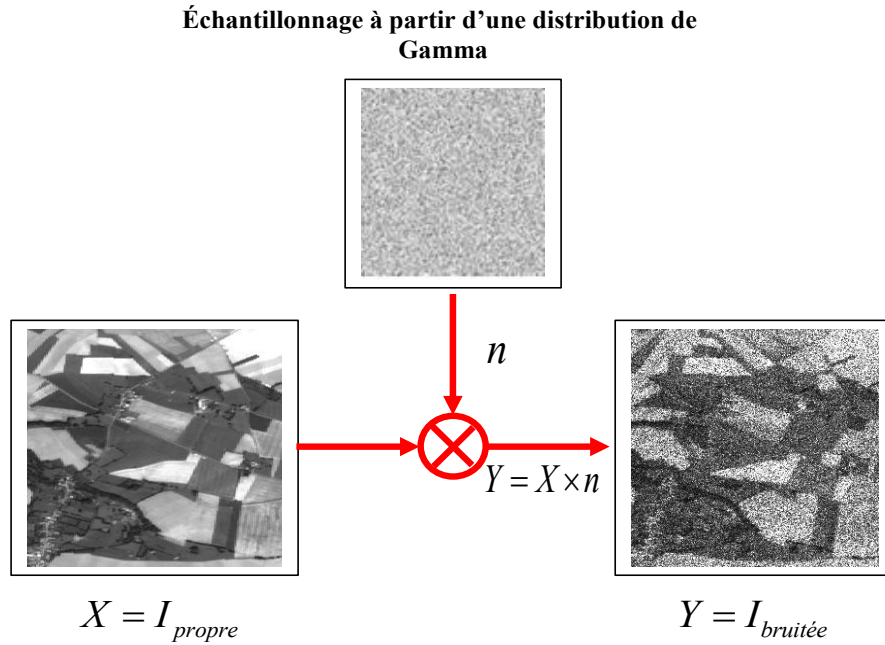


Figure IV. 3. Génération synthétique du bruit speckle.

Cette approche permet de générer des données réalistes et variées, nécessaires pour l'entraînement et l'évaluation des méthodes de débruitage, notamment celles basées sur les réseaux de neurones convolutifs.

Afin d'analyser l'influence du paramètre L sur le bruit speckle, plusieurs cas représentatifs sont étudiés.

- **Cas $L=1$:** Ce cas correspond à une situation où le bruit speckle est très fort, avec une variance maximale $Var[n]=1$. L'image est fortement dégradée et les détails sont peu visibles.
- **Cas $L=4$:** Dans ce cas, le niveau de bruit diminue progressivement, avec une variance $Var[n]=0.25$. L'image présente une dégradation modérée, tout en conservant une partie des structures importantes.
- **Cas $L=10$:** Ce cas représente une situation de faible bruit, où la variance devient faible $Var[n]=0.1$. L'image est beaucoup plus propre et les détails sont mieux préservés.

Ce processus permet de constituer des paires d'apprentissage supervisé : image propre (clean), et image bruitée (noisy). Les Figures IV.4 et IV.5 illustrent deux exemples de ces images pour différentes valeurs du paramètre L , permettant d'observer l'effet du bruit speckle sur la qualité visuelle.

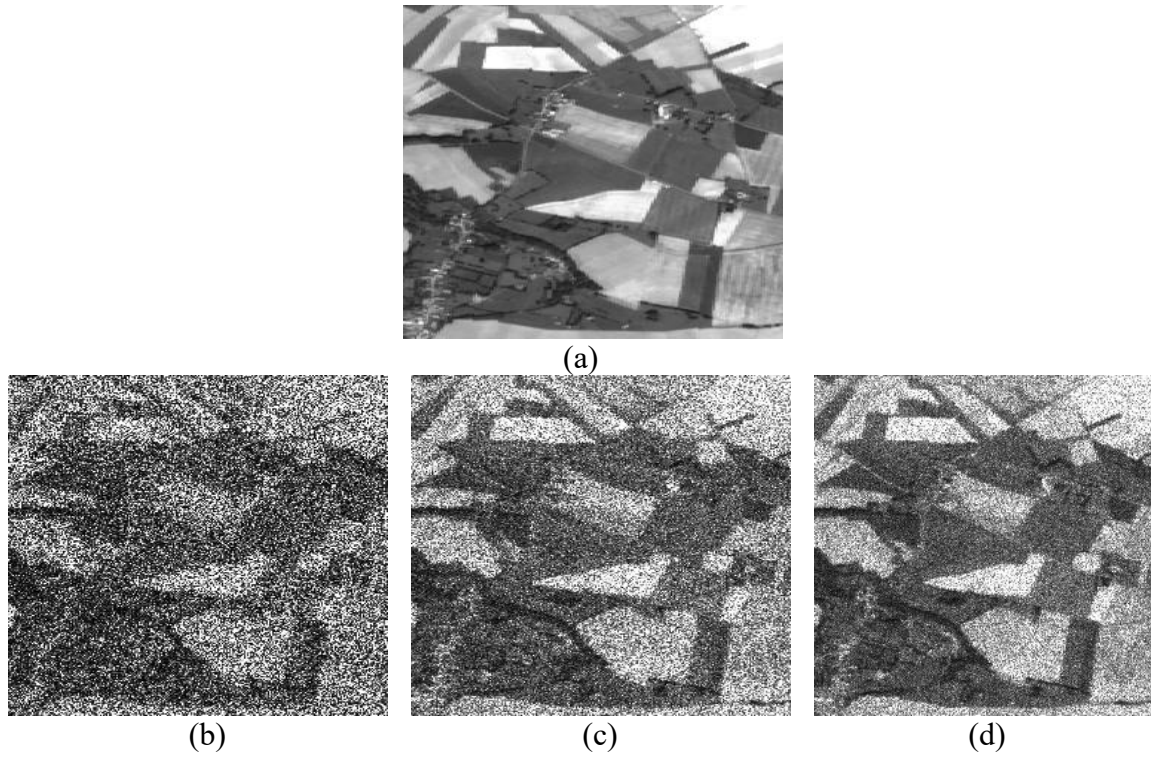


Figure IV. 4. Exemple 1, (a) image propre (clean), (b) image bruitée (noisy) avec $L=1$, (c) image bruitée (noisy) avec $L=4$, (d) image bruitée (noisy) avec $L=10$.

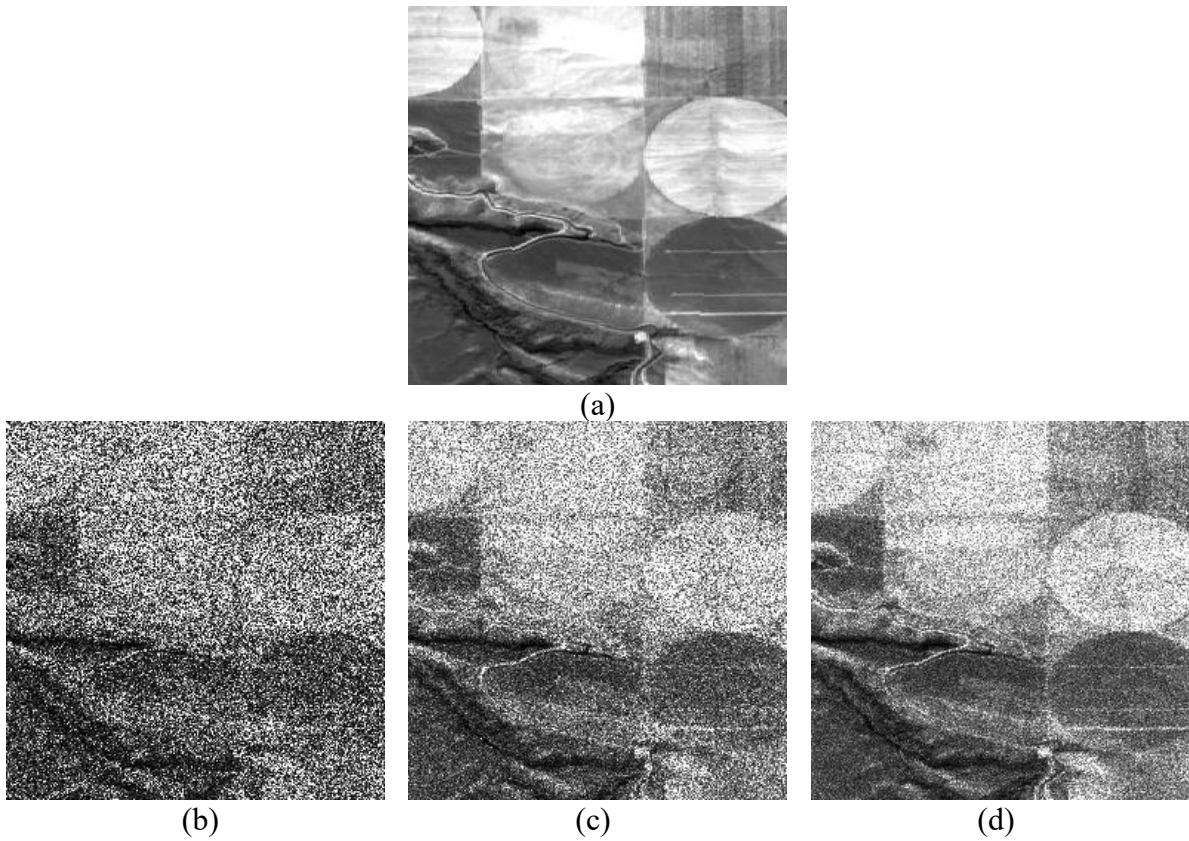


Figure IV. 5. Exemple 2, (a) image propre (clean), (b) image bruitée (noisy) avec $L=1$, (c) image bruitée (noisy) avec $L=4$, (d) image bruitée (noisy) avec $L=10$.

IV.3. Critères d'évaluation

L'évaluation des performances du modèle proposé repose sur l'utilisation de plusieurs métriques permettant de quantifier la qualité des images débruitées par rapport aux images de référence (ground truth). Dans ce travail, trois indicateurs principaux sont utilisés : le PSNR (Peak Signal-to-Noise Ratio), le SSIM (Structural Similarity Index) et le UQI (Universal Quality Index). Ces mesures permettent d'évaluer à la fois la fidélité de reconstruction, la similarité structurelle et la qualité globale des images reconstruites.

PSNR : Le premier critère utilisé est le PSNR. Contrairement à une simple mesure de distorsion, il permet d'évaluer la fidélité de reconstruction de l'image restaurée. Cette métrique est directement liée à la qualité de l'image reconstruite et dérive de l'erreur quadratique moyenne (Mean Squared Error, MSE), largement utilisé en traitement du signal.

$$PSNR = 10 \log_{10} \left(\frac{I_{\max}^2}{MSE} \right) \quad (IV.3)$$

où $I_{\max}$ représente la valeur maximale des niveaux de gris de l'image. Une valeur infinie du PSNR correspond à une reconstruction parfaite, sans aucune dégradation. En pratique, plus la valeur du PSNR est élevée, meilleure est la qualité de l'image restaurée.

Ainsi, le PSNR établit un lien direct entre l'erreur de reconstruction et l'intensité maximale de l'image, ce qui explique son utilisation fréquente dans l'évaluation des méthodes de débruitage et de restauration d'images [114].

SSIM : Le deuxième critère est le SSIM, une métrique utilisée pour mesurer la similarité structurelle entre deux images. Contrairement au PSNR, qui se base uniquement sur l'erreur pixel par pixel, le SSIM prend en compte la perception visuelle humaine en intégrant trois composantes principales : la luminance, le contraste et la structure [115].

Le SSIM est défini à partir de trois fonctions :

Comparaison de la luminance :

$$I(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (IV.4)$$

Comparaison du contraste :

$$C(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (IV.5)$$

Comparaison de la structure :

$$S(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (IV.6)$$

où :

- μ_x , et μ_y sont les moyennes des images originale et reconstruite ;
- σ_x , et σ_y sont les écarts types ;
- σ_{xy} est la covariance entre les deux images ;
- C_1, C_2, C_3 sont des constantes permettant d'éviter les instabilités numériques.

Le SSIM global est alors défini par :

$$SSIM(x, y) = I(x, y) \cdot C(x, y) \cdot S(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x\sigma_y + C_2)(\sigma_{xy} + C_3)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)(\sigma_x\sigma_y + C_3)} \quad (IV.7)$$

UQI : Le dernier critère d'évaluation est l'index de qualité universel (UQI). Il repose principalement sur une analyse locale des moments statistiques et est obtenu en combinant différentes informations structurelles locales de l'image [116].

$$UQI = \frac{1}{N} \sum_{i=1}^N Q(i) \quad (IV.8)$$

où :

$$Q(i) = \frac{4\sigma_{xy}(i) \cdot \overline{x(i)} \cdot \overline{y(i)}}{(\sigma_x^2(i) + \sigma_y^2(i))(\overline{x(i)}^2 + \overline{y(i)}^2)} \quad (IV.9)$$

avec $\sigma_x(i)$ et $\sigma_y(i)$ sont les écarts types du bloc i , $\overline{y(i)}$ et $\overline{x(i)}$ sont les moyennes du bloc i . N représente le nombre total de blocs dans l'image. $\sigma_{xy}(i)$ est la covariance du bloc i .

IV.4. Méthode proposée

Ce travail est organisé en deux grandes catégories d'approches. La première catégorie repose sur l'utilisation de méthodes classiques de filtrage, telles que les filtres de Lee, de Kuan, et de Frost, appliquées au débruitage des images SAR. Ces techniques constituent des approches de référence largement utilisées dans la littérature scientifique.

La seconde catégorie est basée sur des approches d'apprentissage profond, notamment les réseaux de neurones convolutifs (CNN), qui permettent d'extraire efficacement les caractéristiques spatiales des images SAR et d'améliorer la qualité du débruitage.

L'objectif de la première catégorie est de fournir une base de comparaison avec les méthodes existantes dans la littérature scientifique, afin d'évaluer et de valider les performances de la méthode proposée, basée sur les CNN.

IV.4.1. Méthodes classiques

Dans cette section, nous présentons les résultats expérimentaux obtenus après l'application des filtres classiques de débruitage sur les images SAR. L'objectif principal est d'évaluer leur

efficacité en termes de réduction du bruit speckle, tout en préservant les détails structurels des images.

Trois types de filtres (Lee, Kuan, Frost) sont appliqués directement aux images de test. Leurs performances sont ensuite évaluées à l'aide de différentes métriques, notamment le PSNR, le SSIM, et le UQI. Contrairement aux approches basées sur l'apprentissage profond, ces filtres ne nécessitent pas de phase d'entraînement ni de validation. Ils sont appliqués directement sur les images, ce qui permet une évaluation immédiate de leur efficacité.

IV.4.1.1. Filtre de Lee

Le filtre de Lee a été appliqué sur les images SAR bruitées afin de réduire le bruit speckle tout en préservant au mieux les structures et les détails de l'image. Ce filtre repose sur une approche statistique locale, basée sur le calcul de la moyenne et de la variance dans une fenêtre glissante autour de chaque pixel.

Dans notre expérimentation, trois types d'images ont été considérés pour l'évaluation de ce filtre : l'image originale (de référence), l'image bruitée par le speckle, et l'image restaurée après l'application du filtre de Lee. Cette approche permet d'analyser visuellement et quantitativement les performances du filtre en termes de réduction du bruit et de préservation des détails.

En Python, l'utilisation du filtre de Lee est relativement simple. Le code ci-dessous montre comment implémenter et appliquer ce filtre pour le débruitage des images SAR :

```
def lee_filter(img, mul, size=WINDOW_SIZE):
    img = img.astype(np.float32)
    img_mean = uniform_filter(img, (size, size))
    clean_mean = img_mean / mul
    img_sqr_mean = uniform_filter(img**2, (size, size))
    img_variance = img_sqr_mean - img_mean**2
    img_mean = np.where(img_mean == 0, EPS, img_mean)
    sigma = np.sqrt(img_variance) / img_mean
    var = (img_variance + img_mean**2) / (sigma**2 + mul**2) - clean_mean**2
    T = clean_mean**2 * sigma**2 + mul**2 * var
    T = np.where(T == 0, EPS, T)
    weights = mul * var / T
    output = clean_mean + weights * (img - img_mean)
    return np.clip(output, 0, 1)
```

La même procédure a été appliquée aux 150 images de test pour différents nombres de *looks* ($L=1, 4$ et 10). Les performances moyennes (PSNR, SSIM et UQI) ont ensuite calculées afin d'effectuer une comparaison globale entre les différentes méthodes. Les résultats obtenus sont présentés dans le Tableau IV.1.

Tableau IV. 1. Evaluation des performances moyennes (PSNR, SSIM, UQI) du filtre de Lee sur 150 images de test pour différents nombres de *looks* ($L=1, 4$ et 10).

<i>looks</i>	Critères	Image bruitée	Image débruitée
$L=1$	PSNR	11.46	19.29
	SSIM	0.1438	0.4573
	UQI	0.3902	0.7660
$L=4$	PSNR	15.56	21.83
	SSIM	0.2969	0.5121
	UQI	0.6595	0.8633
$L=10$	PSNR	18.95	22.47
	SSIM	0.4354	0.5311
	UQI	0.8110	0.8816

D'après les résultats obtenus, ce filtre peut provoquer un lissage excessif dans certaines régions, entraînant une perte partielle de textures fines et de détails structurels, ce qui se traduit par le critère SSIM.

IV.4.1.2. Filtre de Kuan

En suivant la même méthodologie que celle adoptée pour le filtre de Lee, le filtre de Kuan a été appliqué aux images SAR bruitées afin de réduire le bruit speckle, tout en prenant en compte de sa nature multiplicative. Ce filtre repose également sur une approche statistique locale, similaire à celle du filtre de Lee, mais utilise une formulation mathématique différente qui lui permettant une meilleure adaptation à certaines variations de texture.

Le code ci-dessous présente l'implémentation du filtre de Kuan et son application au débruitage des images SAR.

```
def kuan_filter(img, size=WINDOW_SIZE, L=1):
    img = img.astype(np.float32)
    img_mean = uniform_filter(img, size)
    img_sqr_mean = uniform_filter(img**2, size)
    img_variance = img_sqr_mean - img_mean**2
    img_variance = np.maximum(img_variance, 0)
    img_mean = np.where(img_mean == 0, EPS, img_mean)
    Ci = np.sqrt(img_variance) / img_mean
    Ci = np.where(Ci == 0, EPS, Ci)
    Cu = 1.0 / np.sqrt(L)
    Cmax = np.sqrt(1 + 2.0 / L)
    weights = (1 - (Cu**2 / Ci**2)) / (1 + Cu**2)
    weights = np.where(Ci <= Cu, 0.0, weights)
    weights = np.where(Ci >= Cmax, 1.0, weights)
    weights = np.clip(weights, 0, 1)
    output = img_mean + weights * (img - img_mean)
    return np.clip(output, 0, 1).astype(np.float32)
```

Les résultats obtenus avec le filtre de Kuan sur les 150 images de test, pour trois niveaux de *looks*, sont présentés dans le tableau IV.2. Ils montrent que ce filtre permet une réduction efficace du bruit speckle. L'image reconstruite présente ainsi une amélioration notable par rapport à l'image bruitée. Toutefois, certaines zones texturées subissent une légère perte de précision, traduisant une limitation dans la préservation des détails fins.

En comparaison avec le filtre de Lee, le filtre de Kuan présente globalement de meilleures performances selon les critères d'évaluation considérés.

Tableau IV. 2. Evaluation des performances moyennes (PSNR, SSIM, UQI) du filtre de Kuan sur 150 images de test pour différents nombres de *looks* ($L=1$, 4 et 10).

<i>looks</i>	Critères	Image bruitée	Image débruitée
$L=1$	PSNR	11.46	19.37
	SSIM	0.1438	0.4750
	UQI	0.3902	0.7722
$L=4$	PSNR	15.56	21.87
	SSIM	0.2969	0.5171
	UQI	0.6595	0.8650
$L=10$	PSNR	18.95	22.50
	SSIM	0.4354	0.5343
	UQI	0.8110	0.8826

IV.4.1.3. Filtre de Frost

Le filtre de Frost est un filtre adaptatif de débruitage basé sur une pondération exponentielle dépendant de la distance entre les pixels et de la variance locale de l'image. Il permet de réduire le bruit speckle tout en adaptant le lissage en fonction des caractéristiques locales de l'image.

De manière similaire aux filtres de Lee et de Kuan, les résultats obtenus sont présentés dans le tableau IV.3. Le code Python utilisé pour son implémentation est présenté par la suite.

Tableau IV. 3. Evaluation des performances moyennes (PSNR, SSIM, UQI) du filtre de Frost sur 150 images de test pour différents nombres de *looks* ($L=1$, 4 et 10).

<i>looks</i>	Critères	Image bruitée	Image débruitée
$L=1$	PSNR	11.46	17.97
	SSIM	0.1438	0.3424
	UQI	0.3902	0.7341
$L=4$	PSNR	15.56	23.99
	SSIM	0.2969	0.6700
	UQI	0.6595	0.9237
$L=10$	PSNR	18.95	25.45
	SSIM	0.4354	0.7299
	UQI	0.8110	0.9441

```

def Frost_filter(img):
    img = img.astype(np.float32)
    h, w = img.shape
    temp = np.zeros((h, w), dtype=np.float32)
    N = Half_window_size
    size = 2 * N + 1
    img = np.pad(img, ((N, N), (N, N)), mode='edge')
    # Distance matrix
    D = np.zeros((size, size), dtype=np.float32)
    for k in range(size):
        for l in range(size):
            D[k, l] = np.sqrt((k - N)**2 + (l - N)**2)
    for i in range(h):
        for j in range(w):
            box = img[i:i+size, j:j+size]
            mul = np.mean(box)
            var = np.var(box)
            afa = np.sqrt(a * var / (mul**2 + eps))
            if afa < 1e-6:
                temp[i, j] = mul
                continue
            W = np.exp(-D * afa)
            W_sum = np.sum(W) + eps
            temp[i, j] = np.sum(box * W) / W_sum
    temp = np.clip(temp, 0.0, 1.0)
    return temp.astype(np.float32)

```

Le filtre de Frost permet une réduction efficace du bruit speckle, avec un comportement adaptatif selon les régions de l'image. Les zones homogènes sont fortement lissées, tandis que certaines structures sont mieux conservées. Cependant, un effet de flou peut apparaître dans certaines régions détaillées.

Les filtres classiques de débruitage (Lee, Kuan et Frost) permettent tous une réduction du bruit speckle avec des performances globalement proches. Cependant, le filtre de Frost se distingue légèrement par son caractère plus adaptatif, offrant un meilleur compromis entre la réduction du bruit et la préservation des structures de l'image. Malgré cela, aucun de ces filtres ne permet une suppression optimale du speckle, ce qui limite leur efficacité par rapport aux méthodes modernes basées sur l'apprentissage profond. Dans ce contexte, la section suivante propose une approche basée sur des réseaux de neurones convolutifs (CNN), visant à améliorer les performances de débruitage des images SAR.

IV.4.2. Approche basée sur les réseaux de neurones convolutifs (CNN)

La seconde catégorie de notre méthode repose sur une approche basée sur les réseaux de neurones convolutifs (CNN), appelée *Image Despeckling Convolutional Neural Network* (ID-CNN).

La méthode proposée présentée dans ce mémoire s'inspire du modèle DnCNN. Dans ce contexte, Chierchia et al. [11] ont proposé une architecture composée de couches convolutionnelles, appelé SAR-CNN, intégrant des blocs Conv-BN-ReLU ainsi qu'une couche résiduelle basée sur une division élément par élément pour estimer le bruit speckle. Afin de traiter la nature multiplicative du bruit, cette approche adopte une méthode homomorphique combinant des transformations logarithmiques et exponentielles, associées à une mesure de similarité adaptée à la distribution du speckle.

Par ailleurs, Wang et al. [12] ont utilisé une architecture similaire à DnCNN, composée de huit blocs Conv-BN-ReLU. Dans leur approche, la fonction de perte basée sur l'erreur quadratique moyenne (MSE) est remplacée par une combinaison d'une perte euclidienne et d'une perte de variation totale, permettant d'obtenir des résultats plus lisses.

Dans notre travail, nous adoptons également une architecture basée sur des blocs Conv-BN-ReLU, avec une sortie de type résiduel. L'image SAR bruitée est modélisée comme la somme de l'image propre et du bruit speckle. L'architecture globale du modèle proposé est illustrée dans la Figure IV.6. Grâce à cette architecture, le réseau apprend une fonction de transformation permettant de passer directement d'une image SAR bruitée à une image débruitée.

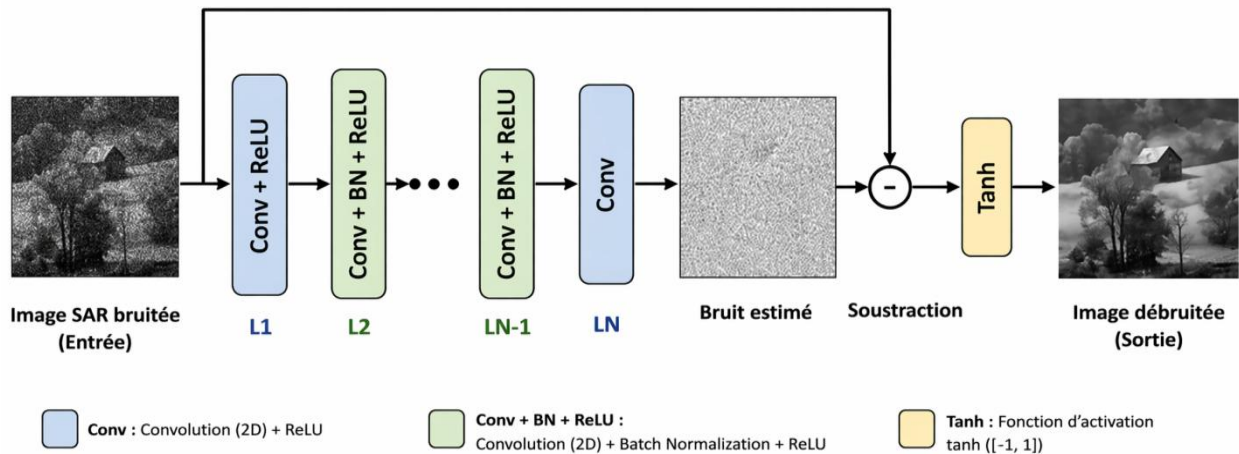


Figure IV. 6. Architecture du réseau ID-CNN proposée pour le débruitage des images SAR. Dans ce cadre, le réseau apprend directement à estimer le bruit à partir de l'image d'entrée. La sortie du modèle correspond alors à une estimation du bruit speckle, et l'image débruitée est obtenue en soustrayant ce bruit estimé de l'image d'entrée. L'équation du modèle résiduel est donnée par :

$$I_{propre} = I_{bruitée} - \hat{N} \quad (IV.10)$$

Afin d'étudier l'impact du nombre de couches convolutionnelles sur les performances du modèle, plusieurs architectures ont été proposées. Plus précisément, une architecture minimale, notée ID-CNN-2L, est composée d'une couche d'entrée (Conv+ReLU) et d'une couche de sortie (Conv). Des architectures plus profondes sont ensuite construites en ajoutant des couches intermédiaires de type Conv-BN-ReLU : une couche intermédiaire pour ID-CNN-3L, deux couches intermédiaires pour ID-CNN-4L, et trois couches intermédiaires pour ID-CNN-5L. Ainsi, les architectures étudiées comportent entre 2 et 5 couches convolutionnelles. Le détail de ces architectures est présenté dans les sous-sections suivantes.

IV.4.2.1. Conception du modèle neuronal CNN

La conception du modèle proposé repose sur une architecture CNN adaptée au débruitage des images SAR. L'objectif principal est d'apprendre une représentation efficace du bruit speckle afin de permettre une reconstruction fidèle de l'image originale.

Le modèle prend en entrée une image de taille $256 \times 256 \times 1$, traitée par une première couche convolutionnelle composée de 64 filtres de taille 3×3 , permettant l'extraction des caractéristiques de bas niveau telles que les contours et les textures locales.

Ensuite, selon la profondeur choisie, des blocs convolutionnels sont empilés, chacun composé de couches Conv, Batch Normalization et ReLU. Ces blocs permettent une extraction hiérarchique des caractéristiques, allant des structures simples vers des représentations plus complexes liées au bruit speckle et à la structure globale de l'image.

Les Batch Normalization jouent un rôle essentiel dans la stabilisation de l'apprentissage en réduisant les variations internes des activations, ce qui permet d'accélérer la convergence du modèle. Les fonctions ReLU introduisent la non-linéarité nécessaire pour modéliser des relations complexes entre les pixels.

La dernière couche du réseau est une couche de convolution composée d'un seul filtre, permettant d'estimer directement le bruit speckle présent dans l'image. L'estimation obtenue est ensuite utilisée dans une opération résiduelle afin de reconstruire l'image débruitée. Enfin, cette sortie est passée à travers une fonction d'activation tanh afin de normaliser les valeurs dans l'intervalle $[-1, 1]$.

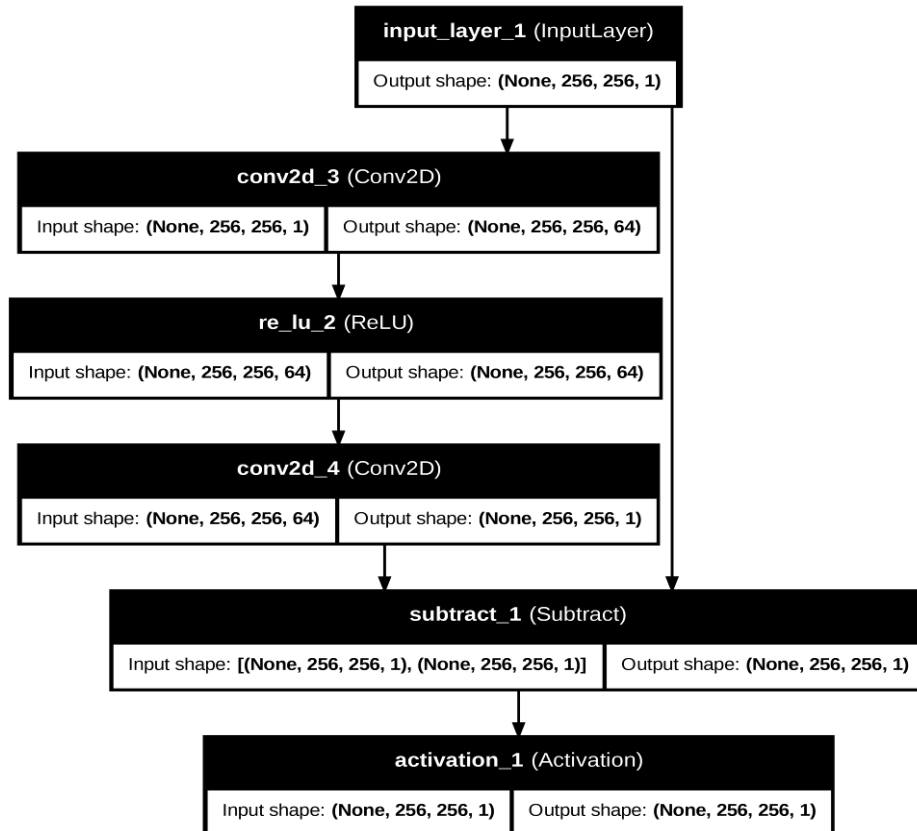
Le réseau proposé comporte ainsi entre 2 et 5 couches convolutionnelles, comme résumé dans le tableau IV.4, correspondant aux architectures ID-CNN-2L, ID-CNN-3L, ID-CNN-4L, ID-CNN-5L. Les Figures IV.7, IV.8, IV.9 et IV.10 présentent les différentes architectures ID-CNN implémentées sous Python selon leur niveau de complexités.

Tableau IV. 4. Configuration du réseau.

Couche	Opérations	Taille du filtre	Nombre de filtres	Taille de sortie
Entrée	Conv ReLu	3×3×1	64	256×256×64 256×256×64
Intermédiaires	Conv BN ReLu	3×3×64	64	256×256×64 256×256×64 256×256×64
Sortie	Conv	3×3×64	1	256×256×1

Pour l'implémentation de ce réseau sous Python, le code suivant illustre la manière dont ces couches sont empilées à l'aide de la bibliothèque tf.keras.

```
def create_model(input_shape=(256,256,1), Number_Layer=2):
    input_layer = Input(shape=input_shape)
    x = Conv2D(64, (3,3), padding='same')(input_layer)
    x = ReLU()(x)
    for _ in range(Number_Layer-2):
        x = Conv2D(64, (3,3), padding='same')(x)
        x = BatchNormalization()(x)
        x = ReLU()(x)
    noise = Conv2D(1, (3,3), padding='same')(x)
    output = tf.keras.layers.Subtract()([input_layer, noise])
    output = Activation('tanh')(output)
    return Model(inputs=input_layer, outputs=output)
```

**Figure IV. 7.** Architecture ID-CNN-2L implémentée sous Python.

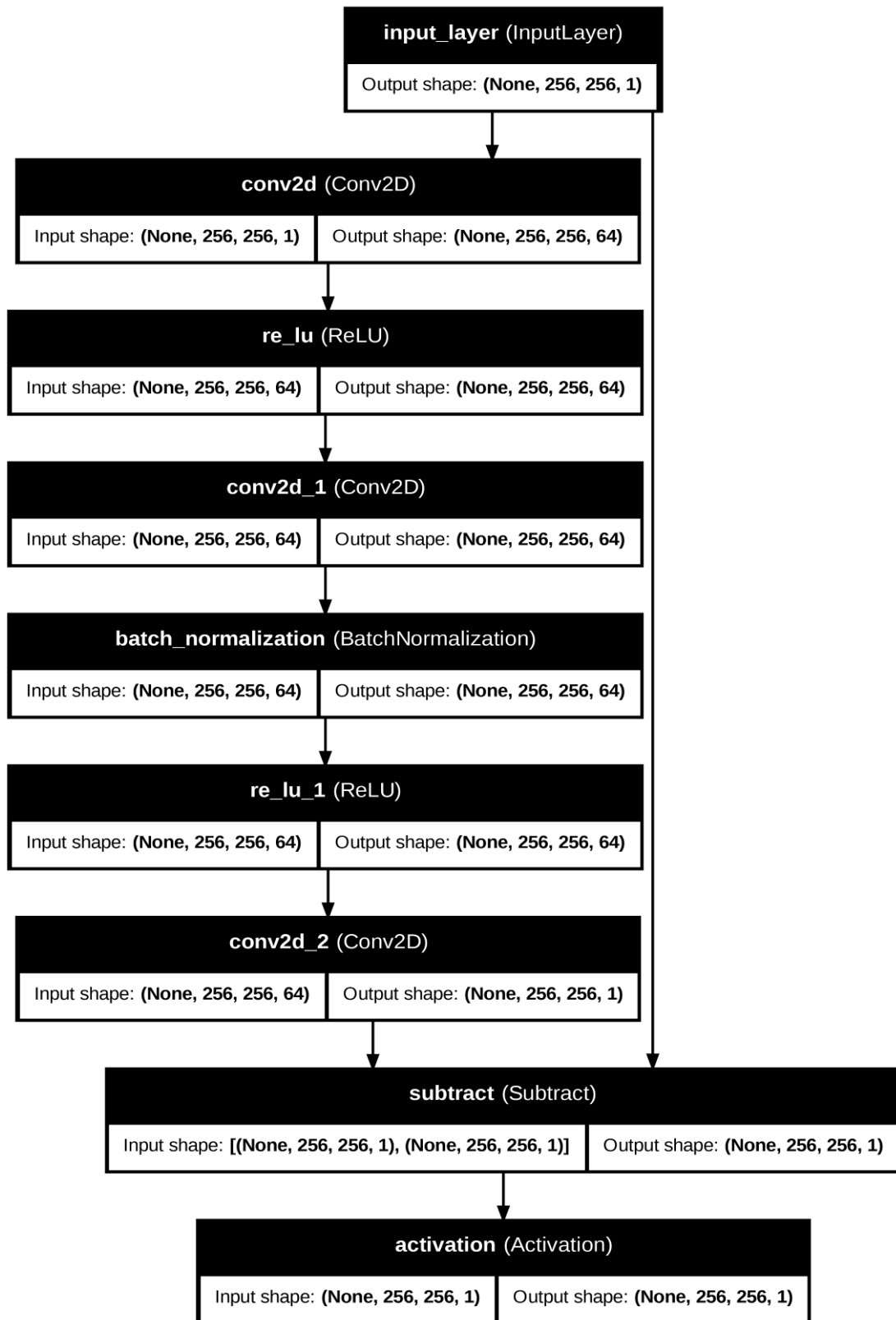


Figure IV. 8. Architecture ID-CNN-3L implémentée sous Python.

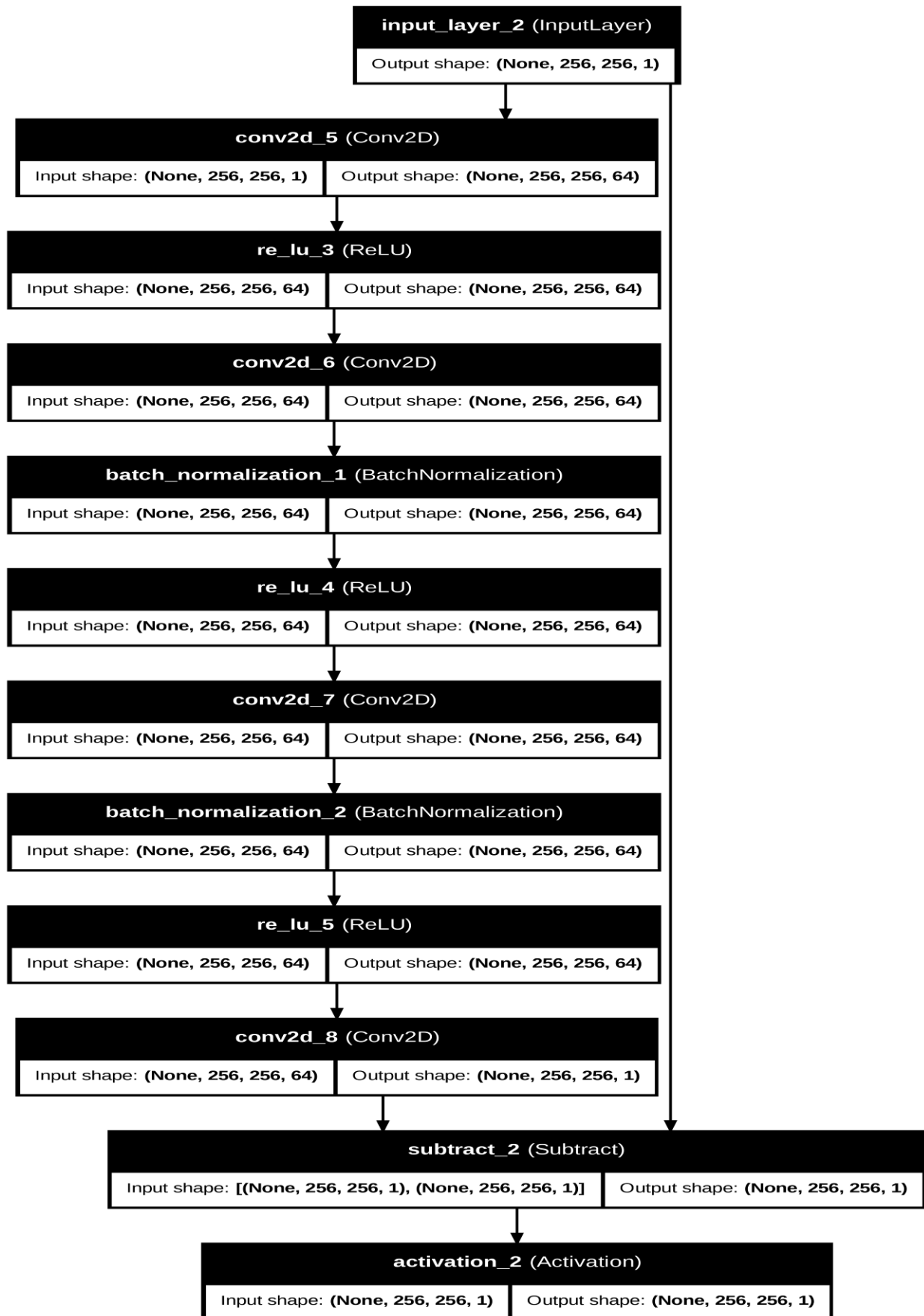


Figure IV. 9. Architecture ID-CNN-4L implémentée sous Python.

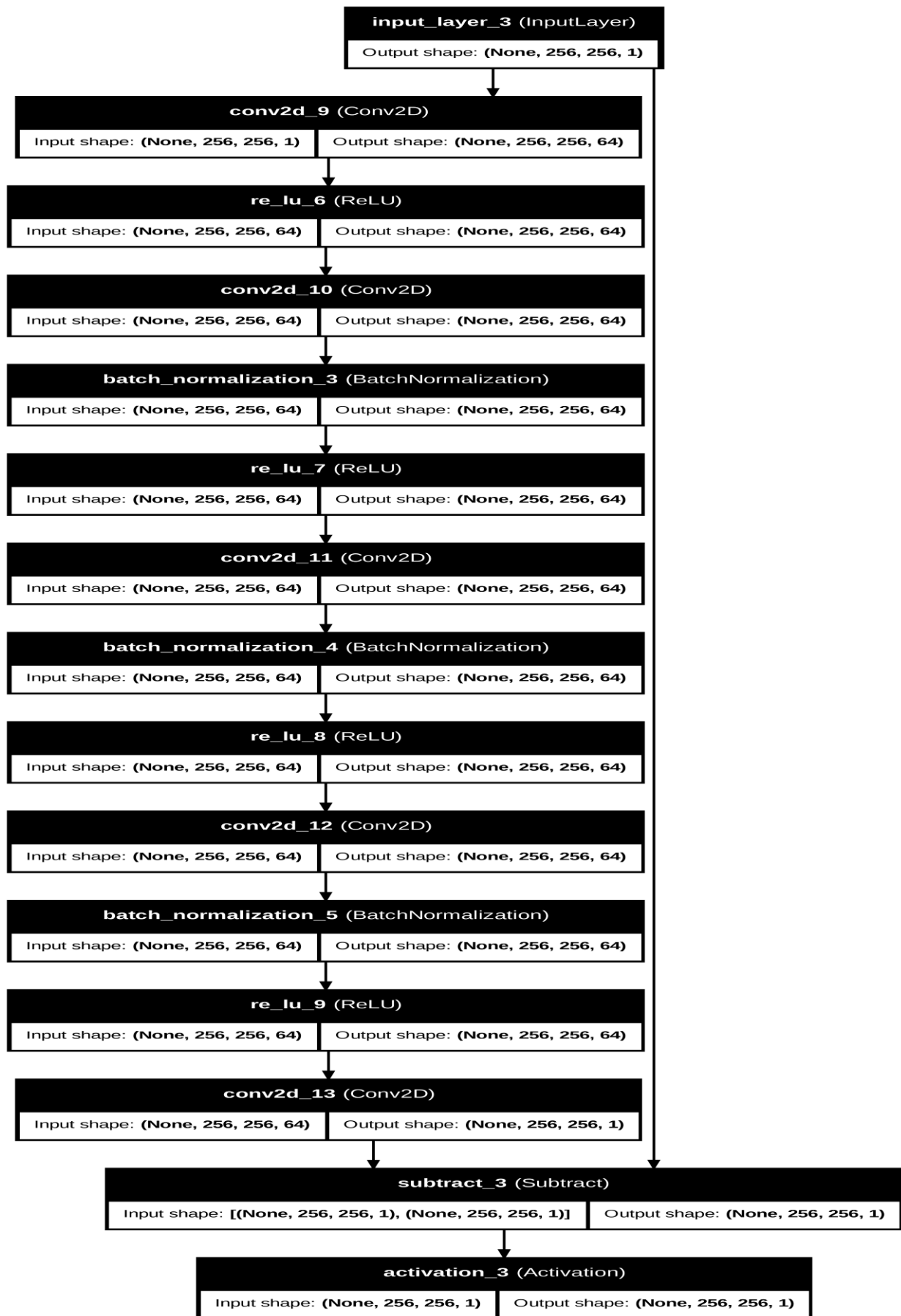


Figure IV. 10. Architecture ID-CNN-5L implémentée sous Python.

IV.4.2.2. Apprentissage des modèles CNN

Après la conception des différentes architectures ID-CNN, les modèles ont été entraînés sur les jeux de données SAR. Cette étape nécessite la définition d'une fonction de perte adaptée au problème de débruitage, ainsi que la mise en place de la stratégie et des paramètres d'apprentissage. Ces aspects sont présentés dans la suite de cette section.

IV.4.2.2.1. Fonction de perte

Les fonctions de perte jouent un rôle essentiel dans l'apprentissage des réseaux de neurones, notamment dans les tâches de restauration d'images. Elles permettent de mesurer l'écart entre la sortie du modèle et l'image de référence (ground truth), afin d'optimiser les paramètres du réseau [12].

Dans ce travail, nous utilisons la fonction de perte euclidienne, également appelée perte L2, qui est l'une des plus couramment employées dans les problèmes de restauration d'images. Elle consiste à minimiser la différence quadratique entre l'image reconstruite par le réseau et l'image originale.

Soit un couple d'images $\{X, Y\}$, où Y représente l'image bruitée en entrée et X l'image de référence (ground truth). Le réseau apprend une fonction ϕ permettant de produire une estimation de l'image débruitée $\hat{X} = \phi(Y)$

La fonction de perte Euclidienne est définie par :

$$L_E = \frac{1}{W \times H} \sum_{w=1}^W \sum_{h=1}^H \left(\phi(Y_{w,h}) - X_{w,h} \right)^2 \quad (\text{IV.11})$$

où :

- $W \times H$ représente les dimensions de l'image ;
- $X_{w,h}$ est le pixel de l'image de référence ;
- $\phi(Y_{w,h})$ est le pixel prédit par le réseau.

Au cours de la phase d'entraînement, l'évolution de la perte euclidienne (L2 Loss) est suivie afin d'analyser le processus d'apprentissage du modèle. Cette évolution est représentée graphiquement en fonction des différentes époques d'entraînement.

IV.4.2.2.2. Stratégie et paramètres d'apprentissage

La stratégie d'apprentissage adoptée repose sur une approche de considération supervisée, où le réseau est entraîné à partir de paires d'images : une image dégradée par le bruit de speckle et sa version originale correspondante dite propre. Le processus a été structuré en plusieurs phases pour garantir la stabilité et la robustesse du modèle :

Phase 1: Initialisation et reproductibilité

Pour assurer la fiabilité des expériences et la comparabilité des résultats, une étape cruciale consiste à fixer les conditions de reproductibilité. Un grain aléatoire (Seed=42) a été défini pour contrôler toutes les sources de variabilité dans les bibliothèques (NumPy, Random, TensorFlow). Cette étape garantit l'obtention de résultats identiques lors de chaque exécution du modèle sous les mêmes conditions.

```
SEED = 42
np.random.seed(SEED)
random.seed(SEED)
tf.random.set_seed(SEED)
```

Phase 2: Définition des paramètres d'apprentissage

Les paramètres opérationnels du réseau ont été définis avec précision :

- **Nombre d'époques:** Fixé à 100 pour offrir un temps d'apprentissage suffisant à la capture des motifs complexes.
- **Taux d'apprentissage (Learning rate):** Une valeur de 2×10^{-3} a été choisie pour équilibrer la vitesse de convergence et la stabilité des poids.
- **Taille des lots (Batch size):** Fixée à 16 images, constituant un compromis entre la précision de l'estimation du gradient et l'efficacité de la mémoire GPU.
- **Dimensions des images:** Redimensionnées à 256×256 pixels pour préserver les détails structurels tout en optimisant les calculs.
- **Architecture du réseau :** Le modèle est composé de 2 à 5 couches convolutionnelles (ID-CNN-2L, ID-CNN-3L, ID-CNN-4L et ID-CNN-5L) pour extraire les caractéristiques profondes liées au bruit.

Phase 3: Choix de l'optimiseur

L'optimisation des poids du réseau est réalisée via l'algorithme Nadam (Nesterov-accelerated Adaptive Moment Estimation). Cet optimiseur combine les avantages de l'adaptation du taux d'apprentissage d'Adam et la force d'inertie de Nesterov. Ce choix permet d'accélérer la convergence tout en réduisant les oscillations autour des minima locaux, un aspect vital lors du traitement d'images SAR fortement bruitées.

Phase 4: Préparation des données et génération du bruit

Les images propres sont chargées, puis un bruit de speckle est ajouté dynamiquement à chaque lot (batch) selon une distribution Gamma, avec un nombre de *looks* L variant entre 1 et 10. Cette stratégie permet au modèle d'être exposé à des variantes de bruit différentes à chaque itération, renforçant ainsi sa capacité de généralisation (robustesse). Enfin, les images

subissent une normalisation dans l'intervalle $[-1,1]$ pour stabiliser le processus d'entraînement.

Phase 5: Stratégie d'apprentissage du modèle

Au lieu de tenter d'apprendre directement l'image propre, le modèle adopte une stratégie de régression résiduelle. Le réseau estime la carte de bruit présente dans l'image bruitée. L'image débruitée est ensuite obtenue par simple soustraction de ce bruit estimé à l'image d'entrée.

Phase 6: Fonction de perte (Loss Function)

Cette étape a été présentée dans la section précédente.

Phase 7: Entraînement et validation

L'entraînement s'effectue de manière itérative. Les ensembles d'apprentissage et de validation, construits précédemment, sont utilisés respectivement pour la mise à jour des poids du modèle (700 images) et pour l'évaluation des performances sur des données non vues (150 images). Cette étape permet de surveiller la capacité de généralisation du modèle tout au long de l'apprentissage.

Phase 8: Sauvegarde du meilleur modèle

Pour protéger le modèle contre le surapprentissage (overfitting), deux mécanismes ont été intégrés :

1. **Sauvegarde du meilleur modèle:** Les poids ne sont sauvegardés que lorsque la perte de validation atteint son minimum historique.

```
patience = 10
if v_loss_epoch < best_val_loss:
    best_val_loss = v_loss_epoch
    counter = 0
    # Save best model
    model.save(best_model_path)
    print(f"Best model updated (Val Loss={v_loss_epoch:.6f})")
else:
    counter += 1
    print(f"No improvement ( {counter}/{patience} )")
```

2. **Arrêt anticipé (Early Stopping):** L'entraînement est interrompu si aucune amélioration n'est observée sur l'ensemble de validation pendant 10 époques consécutives.

```
if counter >= patience:
    break
```

Phase 9 : Suivi et analyse des performances

À la fin de chaque époque, les métriques statistiques (PSNR, SSIM, UQI) sont enregistrées et des courbes d'apprentissage sont générées. Ces visualisations permettent d'analyser le comportement du réseau et de confirmer la convergence entre les courbes d'entraînement et de validation.

Phase 10 : Test des performances des modèles ID-CNN

Le modèle optimal est évalué sur l'ensemble de test constitué de 150 images, afin de mesurer ses performances sur des données totalement non vues lors de l'entraînement.

IV.4.2.3. Résultats et discussion des modèles CNN

Une fois les deux aspects présentés précédemment (fonction de perte ainsi que stratégie et paramètres d'apprentissage) définis, les différents modèles sont entraînés sur Google Colab. L'entraînement du premier modèle, ID-CNN-2L, composé de deux couches convolutionnelles, est illustré dans la Figure IV.11. On observe que la perte euclidienne diminue progressivement au fil des itérations, tandis que les métriques PSNR, SSIM et UQI augmentent, ce qui traduit une amélioration continue des performances du modèle. Pour cette architecture, une perte de validation minimale d'environ 0.015 est obtenue, correspondant à un PSNR maximal de 25.05, un SSIM de 0.68 et un UQI proche de 0.94. Ces résultats indiquent que cette architecture est adaptée au débruitage des images SAR.

La deuxième architecture, ID-CNN-3L, composée de trois couches convolutionnelles et présentée dans la Figure IV.12, montre un comportement légèrement différent. En effet, un phénomène de surapprentissage apparaît après environ cinq itérations, se traduisant par une divergence entre les courbes d'entraînement et de validation, notamment pour la perte euclidienne et le PSNR. Malgré cela, les performances restent élevées, avec une perte de validation minimale d'environ 0.01, un PSNR maximal de 25.25, un SSIM de 0.69 et un UQI atteignant 0.95.

Les architectures plus profondes, ID-CNN-4L et ID-CNN-5L (Figures IV.13 et IV.14), présentent globalement de meilleures performances. L'architecture ID-CNN-4L atteint un PSNR maximal de 25.74, un SSIM de 0.72 et un UQI de 0.95. L'architecture ID-CNN-5L améliore légèrement ces résultats, avec un PSNR maximal de 25.97, un SSIM de 0.72 et un UQI de 0.95.

Ces résultats montrent que l'augmentation du nombre de couches convolutionnelles permet d'améliorer les performances du modèle, bien qu'un compromis doive être trouvé afin d'éviter le surapprentissage.

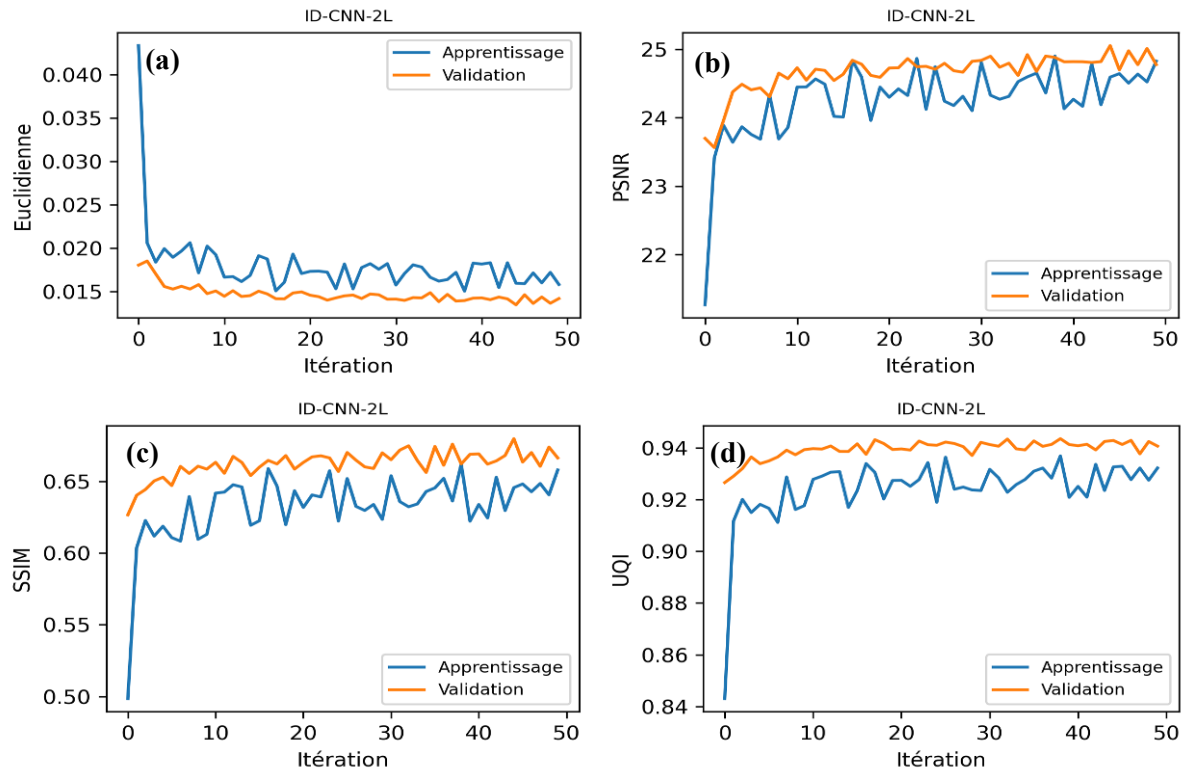


Figure IV. 11. Évolution de la perte euclidienne pour l'architecture ID-CNN-2L durant l'entraînement et la validation, accompagnée du suivi des métriques PSNR, SSIM et UQI : (a) perte euclidienne, (b) PSNR, (c) SSIM, (d) UQI.

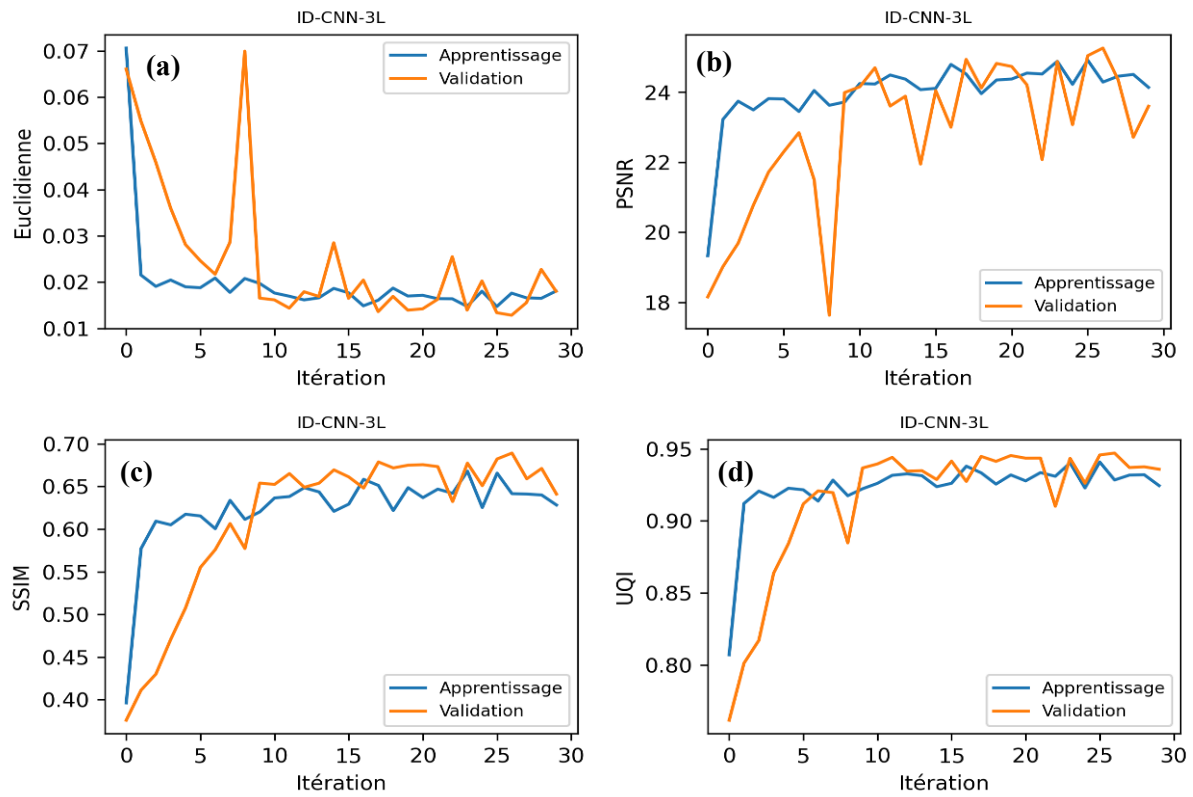


Figure IV. 12. Évolution de la perte euclidienne pour l'architecture ID-CNN-3L durant l'entraînement et la validation, accompagnée du suivi des métriques PSNR, SSIM et UQI : (a) perte euclidienne, (b) PSNR, (c) SSIM, (d) UQI.

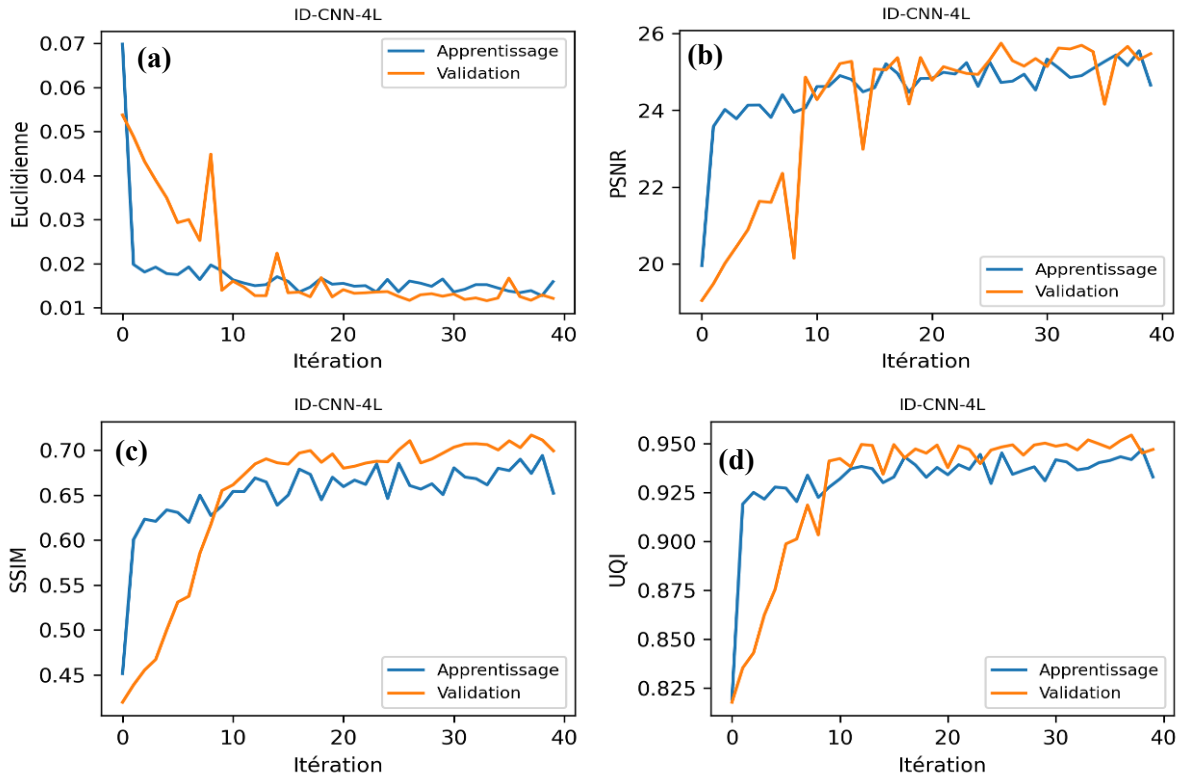


Figure IV. 13. Évolution de la perte euclidienne pour l'architecture ID-CNN-4L durant l'entraînement et la validation, accompagnée du suivi des métriques PSNR, SSIM et UQI : (a) perte euclidienne, (b) PSNR, (c) SSIM, (d) UQI.

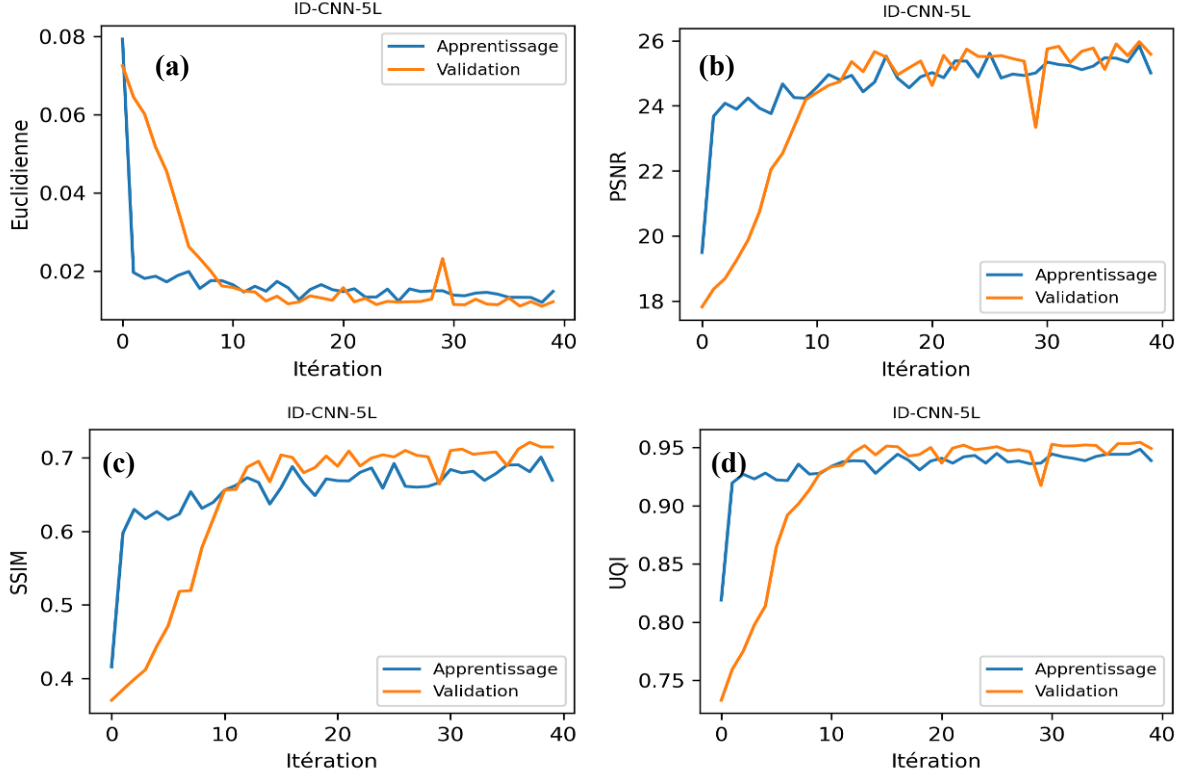


Figure IV. 14. Évolution de la perte euclidienne pour l'architecture ID-CNN-5L durant l'entraînement et la validation, accompagnée du suivi des métriques PSNR, SSIM et UQI : (a) perte euclidienne, (b) PSNR, (c) SSIM, (d) UQI.

Après l'entraînement des modèles ID-CNN, les performances ont été évaluées sur un ensemble de test composé de 150 images. Le tableau IV.5 présente les résultats obtenus pour différentes profondeurs du réseau (de ID-CNN-2L à ID-CNN-5L) et pour plusieurs niveaux de bruit speckle ($L=1, 4$ et 10).

L'analyse de ces résultats met en évidence une amélioration significative des performances des modèles CNN par rapport aux images bruitées, et ce pour toutes les valeurs de L .

Pour $L=1$, correspondant à un niveau de bruit élevé, les performances de l'image bruitée sont faibles (PSNR=11.46dB, SSIM=0.1438 et UQI=0.3902). Après application du modèle ID-CNN, une amélioration notable est observée. L'architecture ID-CNN-2L donne les meilleurs résultats dans ce cas, avec un PSNR atteignant 20.40dB, ainsi qu'une augmentation significative du SSIM et du UQI. Cela montre que les architectures peu profondes sont plus adaptées aux situations de bruit intense.

Pour $L=4$, les performances s'améliorent davantage. Le PSNR atteint environ 25.11dB avec l'architecture ID-CNN-5L, tandis que le SSIM dépasse 0.68 et le UQI 0.94. Ces résultats traduisent une meilleure préservation des structures de l'image ainsi qu'une réduction efficace du bruit speckle.

Pour $L=10$, correspondant à un faible niveau de bruit, les modèles atteignent leurs meilleures performances. L'architecture ID-CNN-5L obtient un PSNR de 26.67dB, un SSIM de 0.7401 et un UQI de 0.9586, indiquant une excellente qualité de reconstruction.

De manière générale, les résultats montrent que les modèles ID-CNN améliorent significativement la qualité des images par rapport aux images bruitées initiales, pour tous les niveaux de bruit. Une augmentation claire des métriques PSNR, SSIM et UQI est observée, confirmant l'efficacité de l'approche proposée.

Par ailleurs, les architectures les plus profondes, notamment ID-CNN-5L, offrent les meilleures performances pour des niveaux de bruit modérés à faibles ($L=4$ et $L=10$). En revanche, pour un bruit très élevé ($L=1$), les architectures plus simples comme ID-CNN-2L semblent mieux adaptées.

Globalement, ces résultats confirment que la méthode proposée basée sur les ID-CNN surpasse largement les images bruitées initiales et reste robuste face à différents niveaux de bruit speckle.

Tableau IV. 5. Evaluation des performances moyennes (PSNR, SSIM, UQI) des modèles ID-CNN sur 150 images de test pour différents nombres de *looks* ($L=1$, 4 et 10).

<i>looks</i>	Critères	Image bruitée	ID-CNN-2L	ID-CNN-3L	ID-CNN-4L	ID-CNN-5L
$L=1$	PSNR	11.46	20.40	18.76	18.87	18.46
	SSIM	0.1438	0.4552	0.4044	0.3872	0.3670
	UQI	0.3902	0.8326	0.7750	0.7780	0.7671
$L=4$	PSNR	15.56	24.17	24.36	24.88	25.11
	SSIM	0.2969	0.6362	0.6485	0.6728	0.6816
	UQI	0.6595	0.9300	0.9340	0.9400	0.9425
$L=10$	PSNR	18.95	26.08	25.77	26.34	26.67
	SSIM	0.4354	0.7310	0.7246	0.7374	0.7401
	UQI	0.8110	0.9537	0.9524	0.9569	0.9586

IV.5. Comparaison entre ID-CNN et les filtres classiques

L'évaluation des performances du modèle proposé par rapport aux filtres classiques de référence (Lee, Kuan et Frost) met en évidence des différences significatives, notamment en fonction de l'intensité du bruit, caractérisée par le nombre de *looks* L .

Pour une analyse visuelle entre les différentes méthodes classiques et l'approche basée sur les réseaux de neurones convolutifs (ID-CNN), la Figure IV.15 présente un exemple appliqué à une image de test. Cette illustration permet de comparer visuellement l'image originale, l'image bruitée et les images débruitées obtenues par les différentes méthodes.

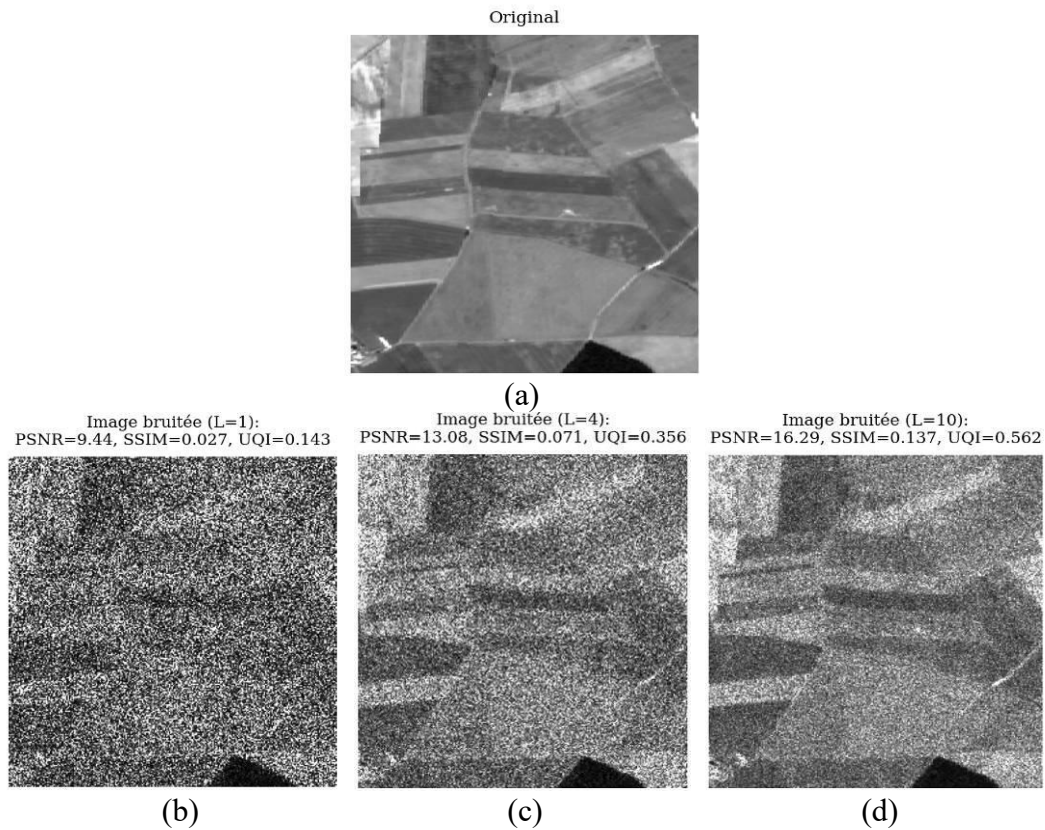


Figure IV. 15. (a) Image de test propre (clean) et images bruitées par speckle pour différents nombres de *looks* : (b) $L=1$, (c) $L=4$ et (d) $L=10$.

L'image originale se caractérise par une bonne qualité visuelle, sans dégradation notable. Après l'ajout du bruit speckle, l'image devient fortement dégradée, ce qui rend les détails difficiles à distinguer.

Comme l'illustrent les Figures IV.16, IV.17 et IV.18, l'application des filtres classiques (Lee, Kuan, Frost) ainsi que de l'approche basée sur les réseaux ID-CNN permet d'obtenir une amélioration significative, avec une réduction du bruit et une meilleure lisibilité de l'image. Toutefois, un léger effet de lissage apparaît dans certaines zones texturées, ce qui peut entraîner une perte partielle des détails fins.

L'évaluation quantitative a été réalisée à l'aide des métriques PSNR, SSIM et UQI pour une image de test issue de la base de données Sentinel-2, sous différents niveaux de looks ($L=1, 4, 10$).

Pour $L=1$, les performances de l'image bruitée sont faibles, avec un PSNR de 9.44dB, un SSIM de 0.027 et un UQI de 0.143. Après application des filtres classiques, une amélioration notable est observée. Parmi ces méthodes, le filtre de Kuan offre les meilleurs résultats, avec un PSNR de 19.40 dB, un SSIM de 0.537 et un UQI de 0.717. Concernant les approches basées sur les ID-CNN, la meilleure performance est obtenue avec l'architecture ID-CNN-2L, atteignant un PSNR de 19.74 dB, un SSIM de 0.274 et un UQI de 0.676. Cela montre une meilleure capacité des réseaux convolutifs à réduire le bruit tout en préservant les structures de l'image.

Pour $L=4$, comme illustré dans la Figure IV.17, l'image bruitée présente un PSNR de 13.08, un SSIM de 0.071 et un UQI de 0.356. Après débruitage, les meilleures performances sont obtenues avec l'architecture ID-CNN-5L, avec un PSNR de 25.79, un SSIM de 0.594 et un UQI de 0.914. Ce modèle présente des performances très proches de ID-CNN-5L, avec une légère amélioration en termes de PSNR et UQI, tandis que ID-CNN-5L conserve une légère supériorité sur le SSIM, traduisant une meilleure préservation structurelle.

Pour $L=10$, la Figure IV.18 montre que l'image bruitée possède un PSNR de 16.29, un SSIM de 0.137 et un UQI de 0.562. Après traitement, les meilleures performances sont également obtenues avec l'architecture ID-CNN-5L, atteignant un PSNR de 28.01dB, un SSIM de 0.699 et un UQI de 0.949. Ces résultats montrent qu'une architecture plus profonde devient plus efficace lorsque le niveau de bruit diminue.

Globalement, ces résultats confirment l'efficacité et la robustesse des architectures ID-CNN, notamment ID-CNN-5L, pour le débruitage des images SAR affectées par le bruit speckle. Elles offrent des performances supérieures aux filtres classiques en termes de PSNR, SSIM et UQI, tout en garantissant une meilleure préservation des détails structurels et texturaux.

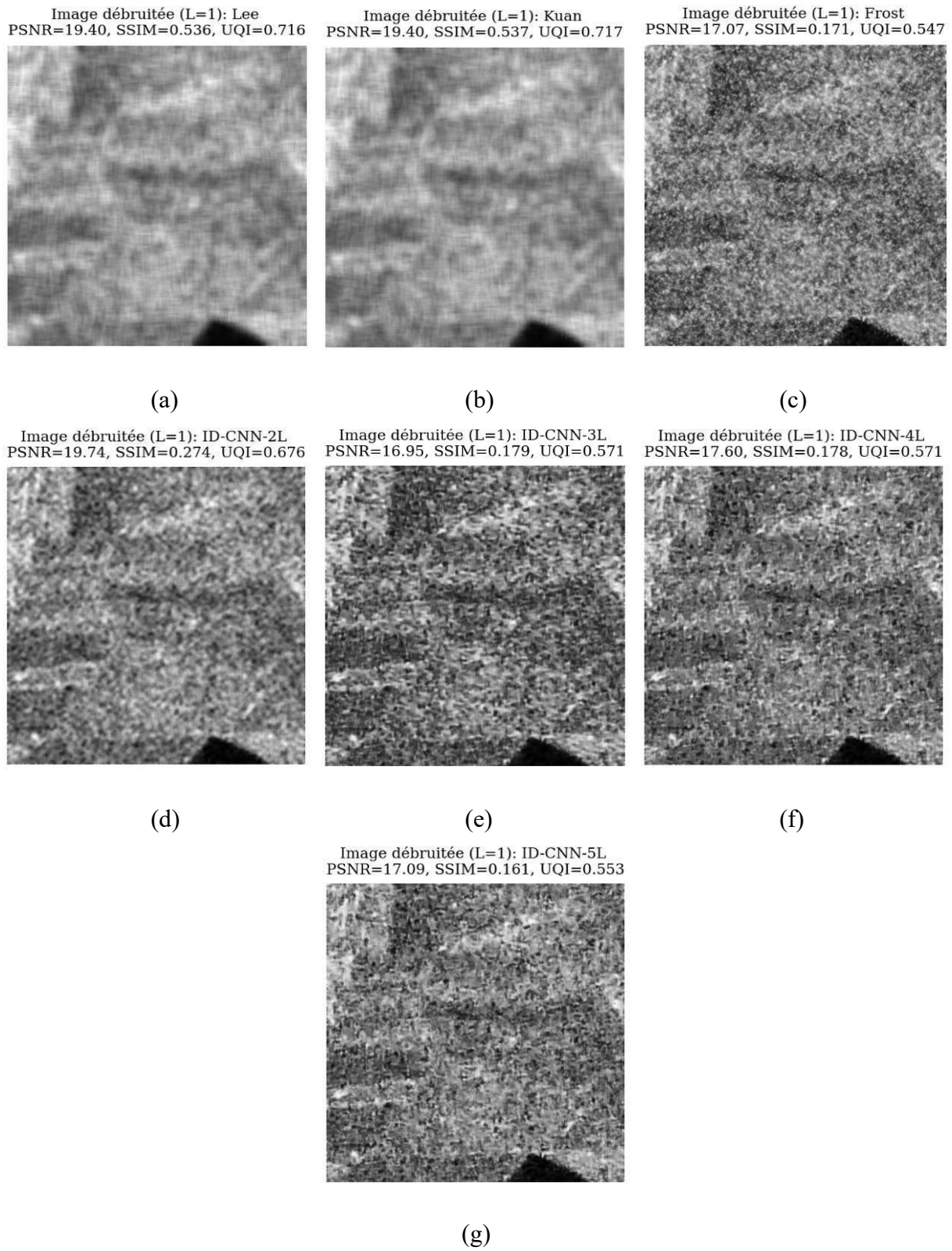


Figure IV. 16. Application des filtres (a)–(c) : Lee, Kuan et Frost, et des différentes architectures ID-CNN (d)–(g) sur une image de test pour un nombre de *looks* $L=1$.

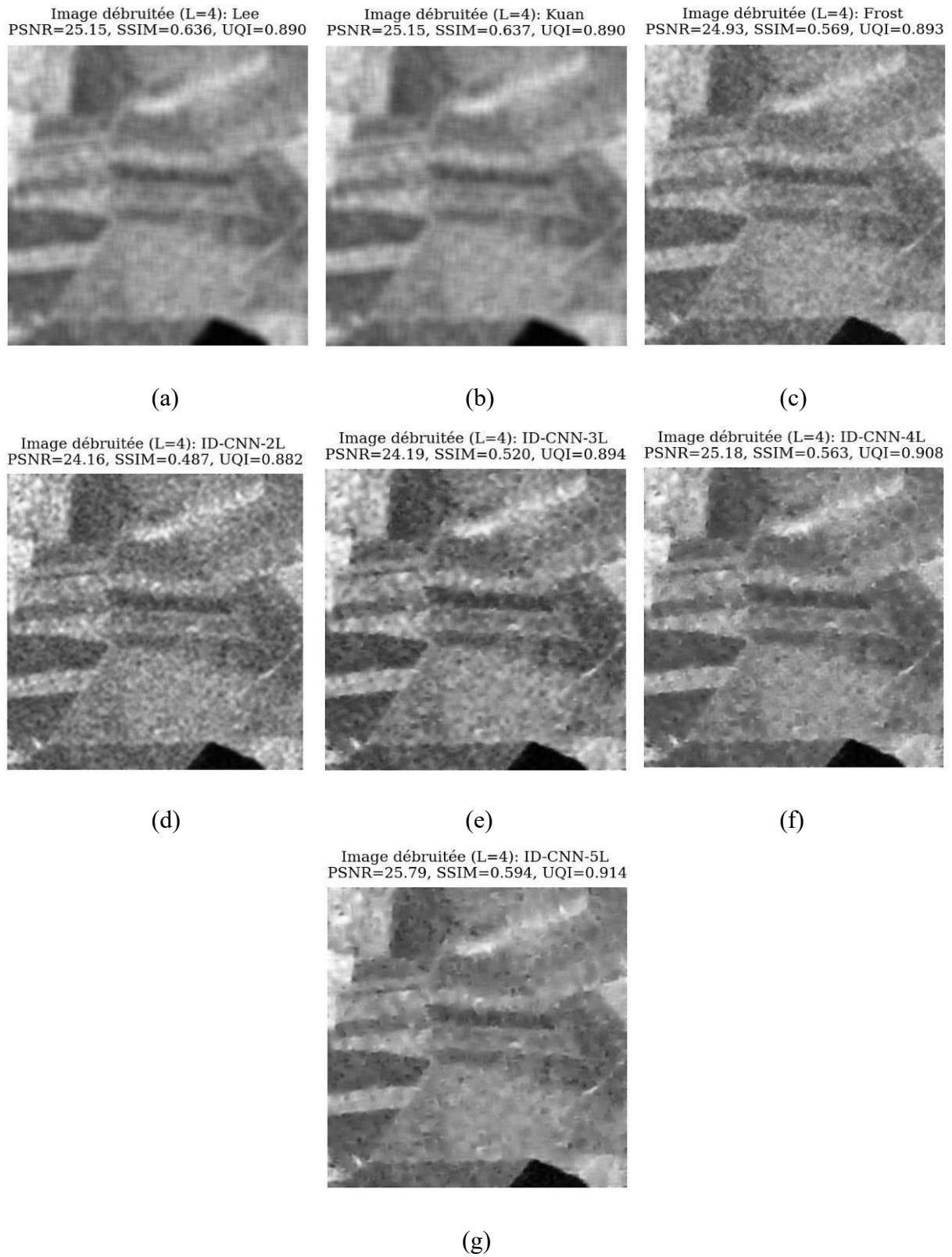


Figure IV. 17. Application des filtres (a)–(c) : Lee, Kuan et Frost, et des différentes architectures ID-CNN (d)–(g) sur une image de test pour un nombre de *looks* $L=4$.

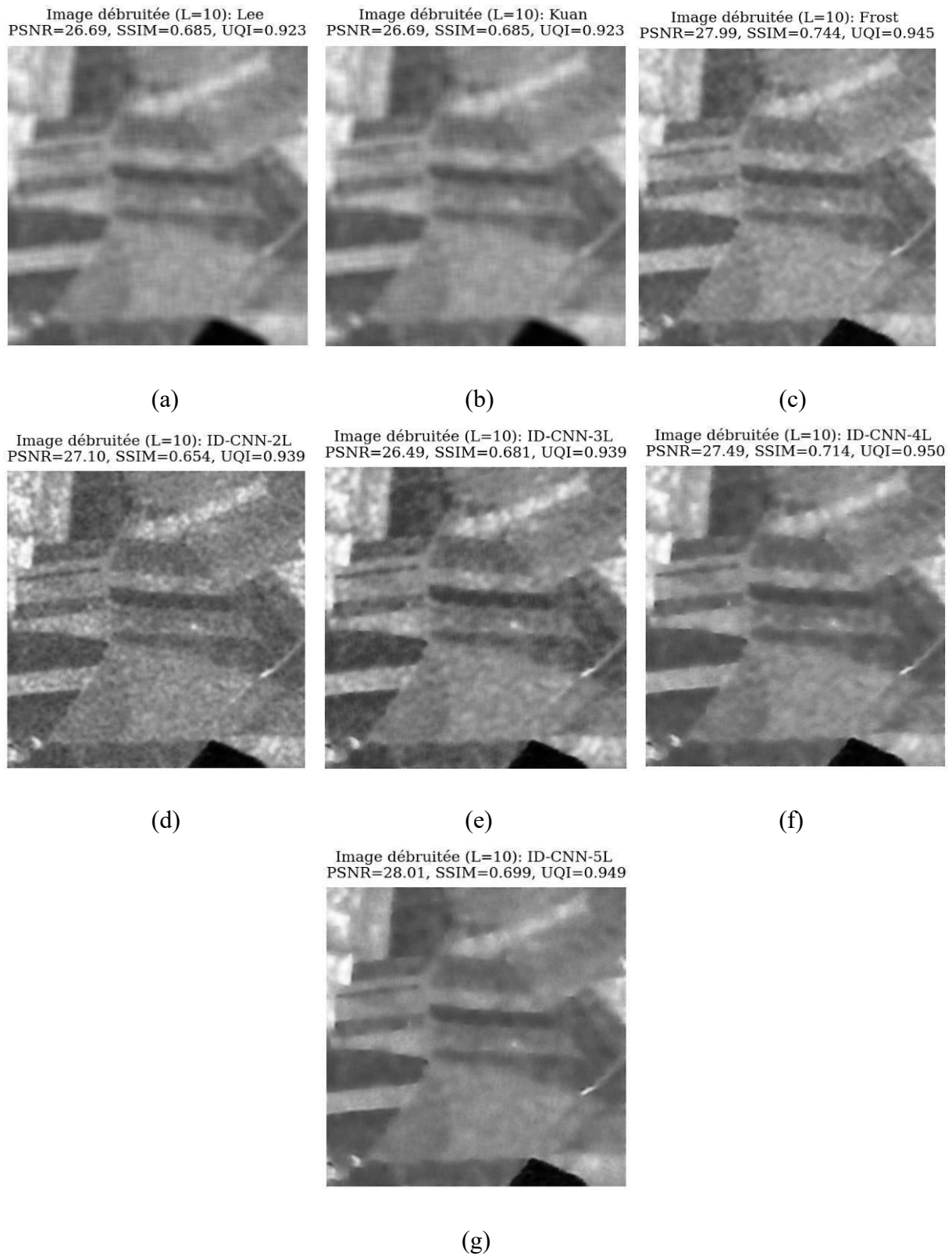


Figure IV. 18. Application des filtres (a)–(c) : Lee, Kuan et Frost, et des différentes architectures ID-CNN (d)–(g) sur une image de test pour un nombre de *looks* $L=10$.

Les résultats précédents, présentés dans les Figures IV.16, IV.17 et IV.18, ont été obtenus à partir d'une seule image de test. Afin de généraliser l'analyse et d'effectuer une comparaison globale entre les différentes méthodes, les Figures IV.19, IV.20 et IV.21 présentent les performances en termes de PSNR, SSIM et UQI, respectivement, calculées sur l'ensemble des 150 images SAR de test.

Ces figures montrent clairement que l'approche basée sur les réseaux de neurones convolutifs (ID-CNN) offre de meilleures performances que les filtres classiques. En effet, une amélioration significative de la qualité des images débruitées est observée pour tous les niveaux de bruit speckle.

Ainsi, le modèle proposé démontre une grande robustesse face aux différentes conditions de bruit, en assurant à la fois une réduction efficace du speckle et une meilleure préservation des structures de l'image.

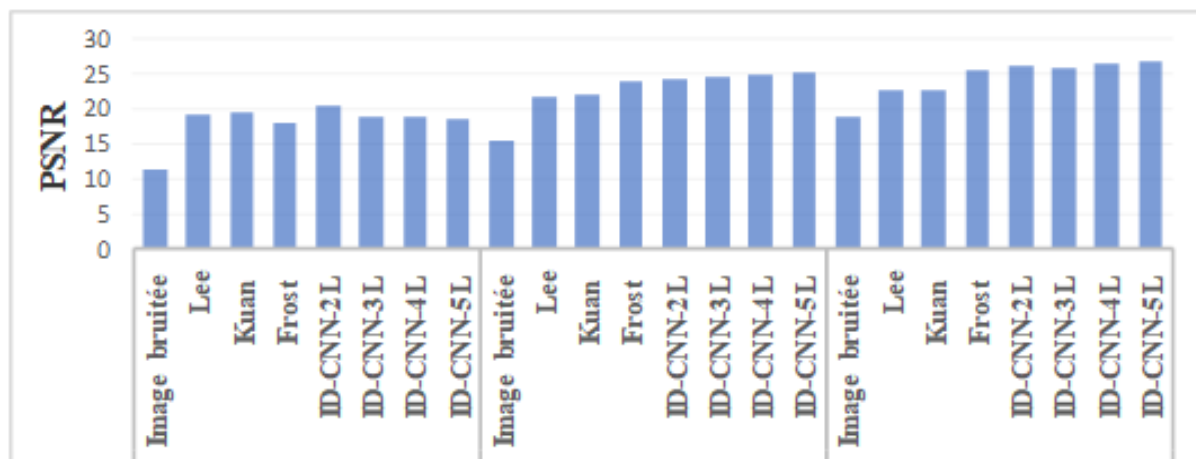


Figure IV. 19. Comparaison entre ID-CNN et les filtres classiques en termes de PSNR.

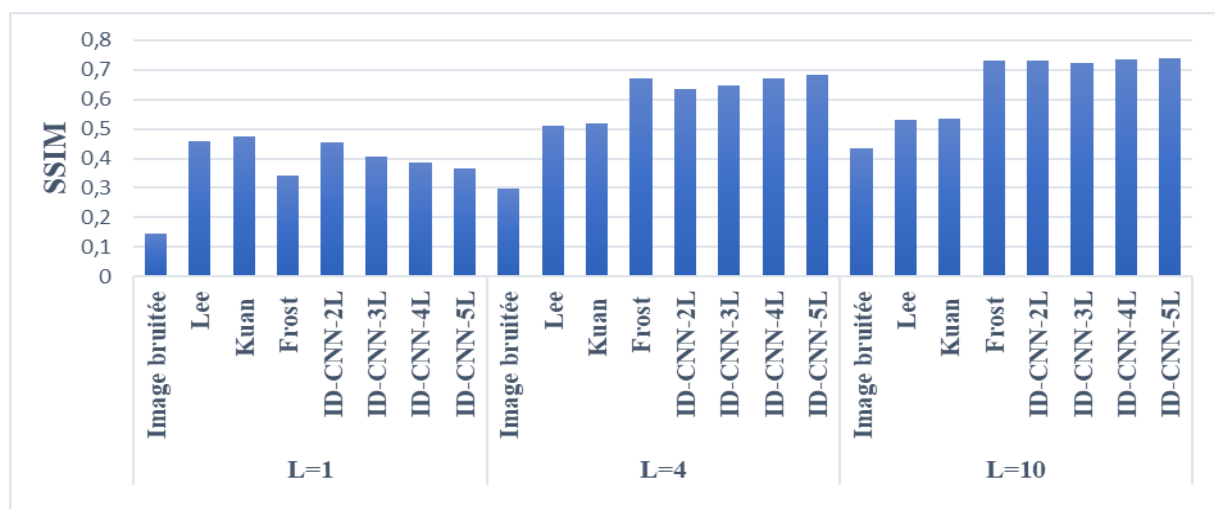


Figure IV. 20. Comparaison entre ID-CNN et les filtres classiques en termes de SSIM.

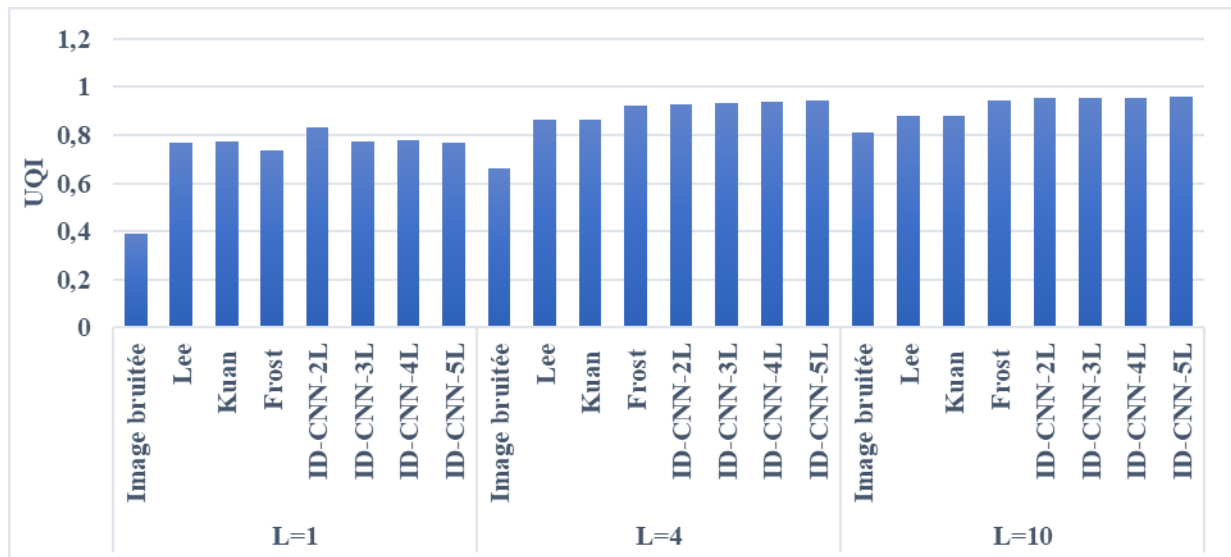


Figure IV. 21. Comparaison entre ID-CNN et les filtres classiques en termes de UQI.

Dans la suite de ce travail, le modèle ID-CNN-5L a été appliqué directement sur des images SAR réelles afin d'évaluer sa capacité de débruitage dans des conditions d'acquisition réelles. Les Figures IV.22, et IV.23 illustrent l'efficacité de la méthode proposée sur des images issues des données Sentinel-1. Ces données ont été retenues en raison de leur capacité à acquérir des images radar à synthèse d'ouverture (SAR) indépendamment des conditions météorologiques et de l'éclairement solaire, ce qui les rend particulièrement adaptées aux applications d'analyse, de surveillance et de traitement d'images en télédétection.

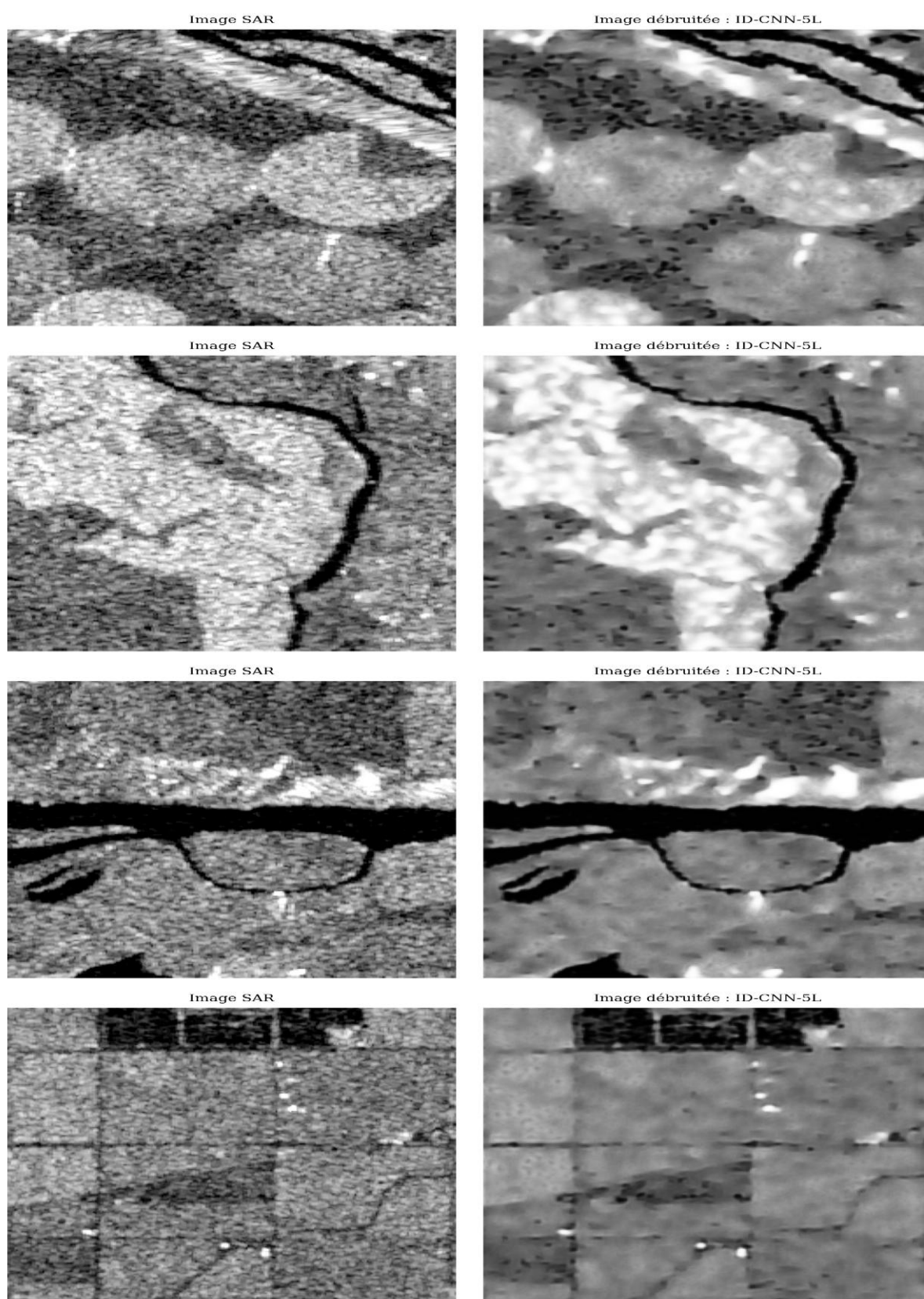


Figure IV. 22. Application du modèle ID-CNN-5L sur les images SAR du jeu de test 1.

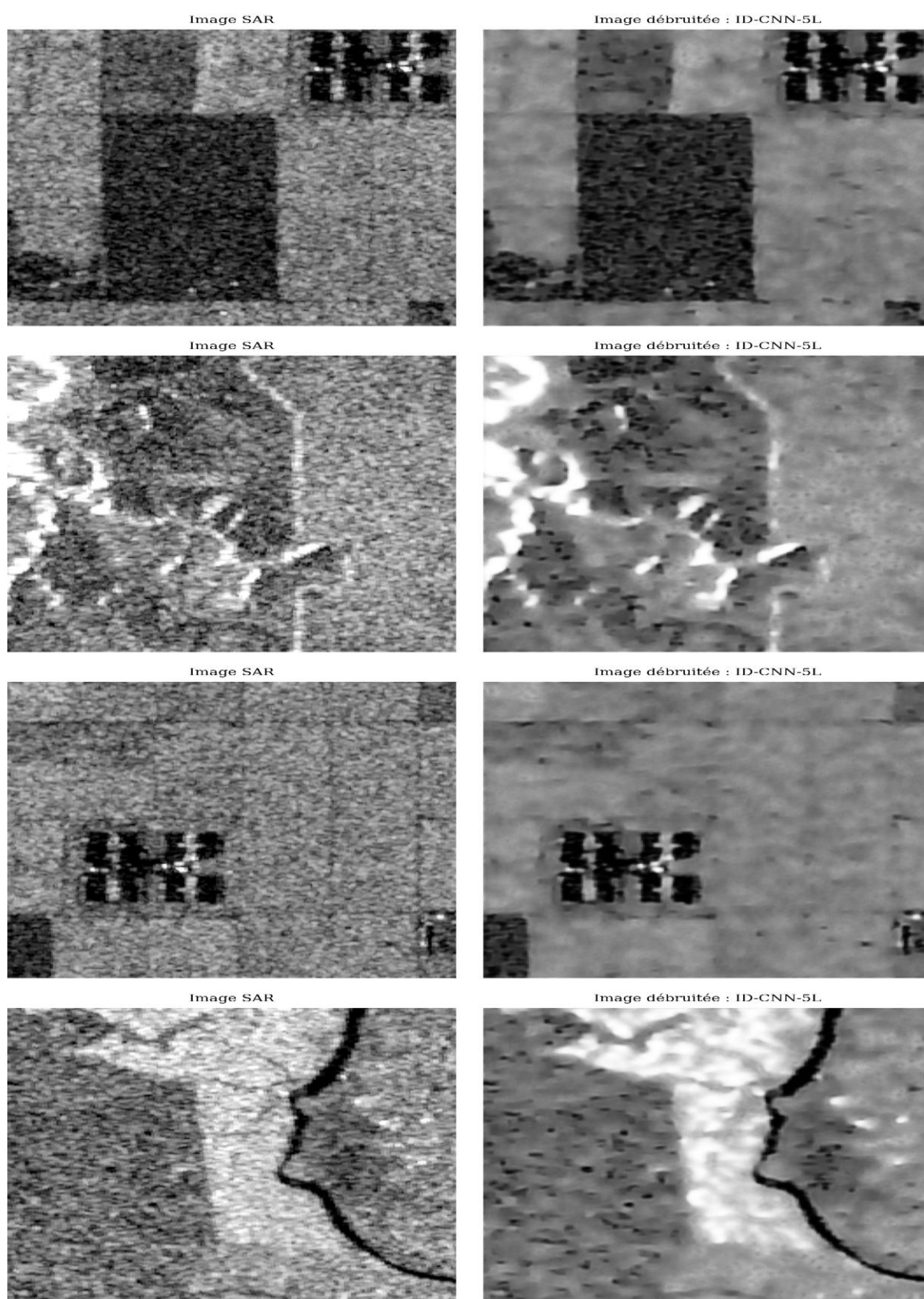


Figure IV. 23. Application du modèle ID-CNN-5L sur les images SAR du jeu de test 2.

IV.6. Conclusion

Dans ce chapitre, une méthode de débruitage des images SAR basée sur les réseaux de neurones convolutifs (CNN) a été présentée et évaluée. Après la génération synthétique du bruit speckle et la préparation des données d'apprentissage, le modèle a été entraîné à l'aide de la fonction de perte Euclidienne (L2).

Les résultats expérimentaux obtenus à travers les métriques PSNR, SSIM et UQI montrent une amélioration significative de la qualité des images débruitées pour différents niveaux de bruit ($L=1$, $L=4$ et $L=10$). Comparée aux méthodes classiques de filtrage, la méthode proposée basée sur les CNN offre de meilleures performances en termes de réduction du bruit et de préservation des détails de l'image.

Ces résultats confirment l'efficacité et la robustesse du modèle proposé pour le débruitage des images SAR affectées par le bruit speckle.

Conclusion générale

Conclusion générale

Ce mémoire a été consacré à l'étude du débruitage dans les images radar à synthèse d'ouverture (SAR) et à la recherche d'une méthode performante de réduire efficacement l'effet du bruit speckle. Après avoir présenté les fondements de l'imagerie SAR ainsi que les limites des approches classiques de débruitage, nous avons proposé une solution fondée sur l'apprentissage profond, et plus particulièrement sur les réseaux de neurones convolutionnels (CNN). L'objectif principal était de concevoir, mettre en œuvre et évaluer un modèle capable de réduire le bruit multiplicatif tout en préservant les structures et les détails essentielles des images.

Les expérimentations réalisées ont montré que l'approche proposée surpasse les méthodes traditionnelles telles que les filtres de Lee, de Kuan et de Frost, tant sur le plan qualitatif que quantitatif. Les résultats obtenus, évalués à l'aide des métriques PSNR, SSIM et UQI, mettent en évidence une amélioration significative de la qualité des images restaurées. Ils confirment également la capacité des CNN à apprendre des représentations complexes adaptées aux caractéristiques spécifiques des images SAR, permettant ainsi une meilleure préservation des contours, des textures et des détails structuraux. Les principales contributions de ce travail résident dans la conception et la mise en œuvre d'une architecture de réseau optimisée pour le débruitage des images SAR, ainsi que dans l'intégration d'une fonction de perte euclidienne favorisant une reconstruction fidèle des images.

Malgré les résultats encourageants obtenus, certaines limites méritent toutefois d'être soulignées. Les performances du modèle dépendent fortement de la qualité, de la quantité et de la diversité des données d'apprentissage disponibles, ce qui peut limiter sa généralisation à des images acquises par des capteurs SAR issues de capteurs différents. Par ailleurs, le coût computationnel associé aux réseaux de neurones profonds reste relativement élevé, ce qui complique leur déploiement en temps réel ou sur des systèmes embarqués. Enfin, les essais réalisés sur des données réelles complexes indiquent que des améliorations supplémentaires sont nécessaires afin d'accroître la robustesse et la fiabilité du modèle dans des conditions opérationnelles diverses.

Dans la continuité de ce travail, plusieurs perspectives de recherche peuvent être envisagées. L'amélioration des performances du modèle pourrait notamment passer par l'exploration d'architectures plus avancées, telles que U-Net, les réseaux antagonistes génératifs (GAN), les Transformers ou encore des architectures hybrides combinant plusieurs paradigmes d'apprentissage. L'intégration de stratégies d'apprentissage plus robustes, notamment l'apprentissage semi-supervisé, auto-supervisé ou auto-encodé, permettrait également de

réduire la dépendance vis-à-vis des bases de données annotées. Enfin, une validation approfondie sur des ensembles de données SAR réelles, diversifiées et issues de différents capteurs, contribuerait à renforcer la capacité de généralisation du modèle et à favoriser son adoption dans des applications opérationnelles de télédétection, de surveillance et d'observation de la Terre.

Bibliographie

Bibliographie

- [1] A. Boufellah and N. Rahma, “Speckle filtering in SAR images using deep learning methods,” Université de Saad Dahlab Blida 1, Mémoire de Master, Option : Spatiales Telecommunications, 2023.
- [2] M. Shimada, “Monitoring tropical deforestation using SAR data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, n°11, pp. 3144–3156, 2006.
- [3] J. S. Lee and E. Pottier, “Polarimetric Radar Imaging: From Basics to Applications,” 2009.
- [4] J. S. Lee, “Speckle analysis and smoothing of synthetic aperture radar images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 20, n°4, pp. 367–378, 1981.
- [5] A. Lopes, R. Touzi, and E. Nezry, “Adaptive speckle filters and scene heterogeneity,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28, n°6, pp. 992–1000, 1990.
- [6] J. S. Lee, “Refined filtering of image noise using local statistics,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 15, n°3, pp. 380–389, 1980.
- [7] M. Dabov, “Image denoising by sparse 3D transform-domain collaborative filtering,” *IEEE Transactions on Image Processing*, vol. 16, n°8, pp. 2080–2095, 2007.
- [8] H. Zhang, “SAR image despeckling using deep convolutional neural networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, n°12, pp. 1928–1932, 2017.
- [9] L. Zhang, “Learning deep CNN denoiser prior for SAR image restoration,” *IEEE Transactions on Image Processing*, vol. 28, n°1, pp. 162–176, 2019.
- [10] Y. Wang, “A deep learning approach for SAR image despeckling,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, n°2, pp. 514–525, 2019.
- [11] G. Chierchia, D. Cozzolino, G. Poggi, L. Verdoliva, “SAR image despeckling through convolutional neural networks,” *In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium, Fort Worth, TX, USA*, pp. 23–28, 2017.
- [12] P. Wang, H. Zhang, V. Patel, “SAR image despeckling using a convolutional neural network,” *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1–5, 2017.
- [13] Q. Zhang, Q. Yuan, J. Li, Z. Yang, X. Ma., “Learning a dilated residual network for SAR image despeckling,” *Remote Sensing*, vol. 10, no. 2, p. 196, 2018.

-
- [14] D. TOUAHRI, “Détection et suivi des cibles mobiles à partir d’images radar,” Mémoire de Master, Spécialité : Informatique, Option : IA, Université 8 Mai 1945-Guelma, 2011.
 - [15] A.Meziane and A.Benzedane, “Etude d’un radar à ouverture synthétique Application à la cartographie 2D,” Mémoire de Master, Option : Communication, Université Mouloud Mammeri de Tizi-Ouzou, 2008.
 - [16] H.Almorin, “Exploration de l’espace architectural à l’aide de la synthèse de haut-niveau: application au traitement radar,” *Thèse de Doctorat*, Nov. 2024.
 - [17] A.Karine, “Traitement et classification parcimonieuse des images radar pour l’aide à la reconnaissance de cibles,” Dec. 2018.
 - [18] I. Cumming, F. Wong, “Digital Processing of Synthetic Aperture Radar Data : Algorithms and Implementation,” Norwood, MA : Artech House, 2005.
 - [19] I.Tifouti, “Contribution à l’Etude de Systèmes Radar à Synthèse d’Ouverture,” Mémoire de Master, Option: Télécommunications, Université 8 Mai 1945-Guelma, 2012.
 - [20] S. Guillaso, “Complémentarité Polarimétrie/ Interférométrie pour la Détection et la Caractérisation de Cibles,” *Thèse de Doctorat, Université de Rennes1*, 2003.
 - [21] A. Veloso *et al.*, “Understanding the temporal behavior of crops using Sentinel-1 and Sentinel-2-like data for agricultural applications,” *Remote Sensing of Environment*, vol. 199, no. 415–426, 2017.
 - [22] O. Ghorbanzadeh, K. Didehban, H. Rasouli, K. V. Kamran, B. Feizizadeh, and T. Blaschke, “An application of sentinel-1, sentinel-2, and GNSS data for landslide susceptibility mapping,” *ISPRS International Journal of Geo-Information*, vol. 10, no. 9, p. 561, 2020.
 - [23] P.Formont, “Outils statistiques et géométriques pour la classification des images SAR polarimétriques hautement texturées,” *Thèse de Doctorat*, Dec. 2013.
 - [24] O. d’Hondt, “Analyse spatiale de texture non stationnaire dans les images SAR,” *Thèse de Doctorat*, Feb. 2006.
 - [25] G.Veyssière, “Apport des mesures du radar à synthèse d’ouverture de Sentinel-1 pour l’étude des propriétés du manteau neigeux,” *Thèse de Doctorat*, Mar. 2019.
 - [26] G.Vasile, “Imagerie Radar à synthèse d’ouverture interférométrique et polarimétrique. application au suivi des glaciers alpins,” *Thèse de Doctorat*, Jul. 2007.
 - [27] F.Comblet, “Détection, localisation et identification de cibles radar par imagerie électromagnétique bistatique,” *Thèse de Doctorat*, Dec. 2005.

-
- [28] J.Khaldoune, “Développement d’une méthode de mesure et de cartographie du gel du sol en milieu agricole à l’échelle d’un bassin versant à partir d’image radar monopolarisées (RadarSat-1),multipolarisées(Envisat) et polarimétriques(RadarSat-2\Convair-580),” Thèse de Doctorat, Université du Québec, 2006.
 - [29] N.Touati, “Optimisation des formes d’ondes d’un radar d’aide à la conduite automobile, robustes vis-à-vis d’environnements électromagnétiques dégradés,” *Thèse de Doctorat*, Nov. 2015.
 - [30] M. Skolnik, *Radar Handbook*. in Third Edition McGraw-Hill Education. 2008.
 - [31] C. Tauber, “Filtrage anisotrope robuste et segmentation par B-spline snake : Application aux images échographiques,” *Thèse de Doctorat*, 2000.
 - [32] G. Franceschetti, R. Lanari, “Synthetic aperture radar processing,” *CRC press*, 2018.
 - [33] F. Adragna, J. Et Nicolas, “Traitement des images de Radar à Synthèse d’Ouverture (RSO),” *Hermes Science Europe, Paris, France*, p. 328, 2001.
 - [34] J-S. Lee, Jong, “Digital image enhancement and noise filtering by use of local statistics,” *IEEE transactions on pattern analysis and machine intelligence* 2, pp. 165–168, 1980.
 - [35] I. Bloch *et al.*, “Le traitement des images-tome 2,” *Polycopié du cours ANIM*, Dec. 2005.
 - [36] A. Mcheik, “Modélisation statistique du Speckle en OCT : application à la segmentation d’images de la peau,” Thèse de Doctorat, Spécialité : Image, Information et Hypermédia, Université de Toulouse, 2010.
 - [37] A. Lopes, E. Nezry, R. Touzi, H. Laur, “Structure detection and statistical adaptive speckle filtering in SAR images,” *International Journal of Remote Sensing*, vol. 14, no. 9, pp. 1735–1758, 1993.
 - [38] D. T. Kuan, , A. A. Sawchuk, T. C. Strand, P. Chavel, “Adaptive noise smoothing filter for images with signal-dependent noise,” *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 165–177, 1985.
 - [39] S. Raslain, “Débruitage d’images médicales par modélisation stochastique spatiale,” Thèse de Doctorat , Option : Automatique et Traitement du signal, Université des frères Mentouri Constantine I, 2019.
 - [40] A. Achim, A. Bezerianos, and P. Tsakalides, “Novel Bayesian multiscale method for speckle removal in medical ultrasound images,” *IEEE Trans. Med. Imaging*, vol. 20, pp. 772–783, 2001.

-
- [41] K. Krissian, C. F. Westin, R. Kikini, and K. Vosburgh, "Oriented speckle reducing anisotropic diffusion," pp. 1412–1424, 2007.
 - [42] Y. Yu and S. T. Acton, "Speckle reducing anisotropic diffusion," 2002.
 - [43] P. Coupé, "Méthode de compensation des déformations cérébrales par imagerie ultrasonore intraopératoire pour la neurochirurgie guidée par l'image," Jan. 2008.
 - [44] V. S. Frost, J. A. Stile, K. S. Shanmugan, and J. C. Holtzman, "A model for radar images and its application to adaptive digital filtering of multiplicative noise," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4(2), pp. 157–166, 1982.
 - [45] N. Wiener, "Etrapolation, Interpolation, and Smoothing of Statio- nary Time Series : With Engineering Applications," *Technology Press of the Massachusetts Institute of Technology*, 1949.
 - [46] B. A. Abraham and Y. Kadah, "Speckle noise reduction method combining total variation and wavelet shrinkage for clinical ultrasound imaging," pp. 80–83, 2011.
 - [47] M. Bertero and P. Boccaccii, "Introdu ction to inverse problems in imaging," 1998.
 - [48] S. Parrilli, M. Poderico, C. V. Angelino, L., Verdoliva, "A nonlocal SAR image denoising algorithm based on LLMMSE wavelet shrinkage," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 2, pp. 606–616, 2011.
 - [49] S. A. Hussain and S. M. Gorashi, "Image Denoising based on Spatial/Wavelet Filter using Hybrid Thresholding Function," *Int. J. Comput. Appl*, vol. 42, pp. 5–13, 2012.
 - [50] M. Y. Jabarulla and H. N. Lee, "Computer aided diagnostic system for ultrasound liver images : A systematic review," *Optik*, vol. 140, pp. 1114–1126, 2017.
 - [51] A. Chambolle, "An algorithm for total variation minimizations and applications," *J. Math. Imaging Vis*, vol. 10, pp. 89–97, 2004.
 - [52] G. Aubert and J. F. Aujol, "A variational approach to removing multiplicative noise," *Siam J. Appl. Math*, vol. 68, pp. 925–946, 2008.
 - [53] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul*, vol. 4, pp. 490–530, 2005.
 - [54] P. Perone and J. Malik, "Scalespace and edge detection using anisotropic diffusion," vol. 12, pp. 629–639, 1990.
 - [55] L. Rudin, S. Osher, and E. Fatemi, "Non linear total variation based noise removal algorithms," pp. 25–268, 1992.
 - [56] S. Yuri and H. Woo, "Alternating minimization algorithm for shifted speckle re- duction variational model," pp. 4188–4191, 2011.

-
- [57] M. Hacini, F. Hachouf, and K. Djemal, “A new speckle filtering method for ultrasound images based on a weighted multiplicative total variation,” 2014.
 - [58] R. statistics, “New York : Wiley,” 2003.
 - [59] L. Ambrosio, N. Fusco, and D. Pallara, “Functions of bounded variation and free discontinuity problems,” 2000.
 - [60] J.Lettvin, H.Maturana, W.McCulloch, and W.Pitts, “What the frog’s eye tells the frog’s brain,” *Proceedings of the Institute of Radio Engineers*, vol. 47, pp. 1940–1959, 1959.
 - [61] Y.Mercadier, “Classification automatique de textes par réseaux de neurones profonds: application au domaine de la santé,” *Thèse de Doctorat*, Nov. 2020.
 - [62] G. L. Shaw, *Brain Theory*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986.
 - [63] F. Rosenblatt, “The perceptron, a perceiving and recognizing automaton,” *Cornell Aeronautical Laboratory*, 1957.
 - [64] A.Harendi and A.Arbaoui, “Calcul de l’MPPT par réseaux de neurone pour un système photovoltaïque,” Mémoire de Master,Filière : Génie électrique ,Spécialité : Electrotechnique Industrielle, Université Kasdi Merbah-Ouargla, 2014.
 - [65] M.Boughaba and B.Boukhris, “L’apprentissage profond (Deep Learning) pour la classification et la recherche d’images par le contenu,” Mémoire de Master,Filière : Informatique ,Spécialité : fondamentale, Université Kasdi Merbah-Ouargla, 2016.
 - [66] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Image Net Classification with Deep Convolutional Neural Networks.” 2012.
 - [67] B. Graham, “Fractional Max-Pooling,” arXiv :1412.6071, Dec. 2014. [Online]. Available: <https://arxiv.org/abs/1412.6071>
 - [68] L.Makhlouf, “L’apprentissage profond appliqué à la reconnaissance des anomalies mammaires,” Mémoire de Master,Filière : Informatique ,Option : Systèmes Informatiques, Université 8 Mai 1945-Guelma, 2022.
 - [69] J.Moolayil and S.John, “Learn Keras for Deep Neural Networks,” pp. 1–192, 2019.
 - [70] V.Nair and G.E.Hinton, “Rectified linear units improve restricted Boltzmann machines,” Jan. 2010.
 - [71] A.L.Maas, A.Y.Hannun, and A.Y.Ng, “Rectifier nonlinearities improve neural network acoustic models,” *Proceedings of the 30th International Conference on Machine Learning (ICML)*, vol. Vol. 30, No. 1, p. 3, Jun. 2013.
 - [72] A.Aggarwal, M.Bansal, T.Verma, and A.Sood, “Residual in Residual Cascade Network for Single-Image Super Resolution,” pp. 335–346, 2022.

-
- [73] X.Wang, G.Liang, Y.Zhang, H.Blanton, Z.Bessinger, and N.Jacobs, “Inconsistent performance of deep learning models on mammogram classification,” *Journal of the American College of Radiology*, vol. 17, no. 6, pp. 796–803, 2020.
 - [74] G.Montavon, G.Orr, and R.K.Müller, *Neural networks: tricks of the trade*, vol. Vol. 7700. Springer.
 - [75] K.Chandra *et al.*, “Gradient descent: The ultimate optimizer,” arXiv:1909.13371, 2019.
 - [76] D.Q.Zeebaree, “Robust watermarking scheme based LWT and SVD using artificial bee colony optimization,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 2, p. 1218-1229, 2021.
 - [77] M.Hardt, B.Recht, and Y.Singer, “Train faster, generalize better: Stability of stochastic gradient descent,” PMLR (Proceedings of Machine Learning Research), International Conference on Machine Learning (ICML), Jun. 2016.
 - [78] S.Ruder, “An overview of gradient descent optimization algorithms,” arXiv:1609.04747, 2016.
 - [79] S.Haji and A.M.Abdulazeez, “Comparison of optimization techniques based on gradient descent algorithm: A review,” *PalArch's Journal of Archaeology of Egypt/Egyptology*, vol. 18, no. 4, p. 2715-2743, 2021.
 - [80] E.M.Dogo, O.J.Afolabi, N.I.Nwulu, B.Twala, and C.O.Aigbavboa, “A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks,” *International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, p. 92-99, 2018.
 - [81] Z.Si, S.Wen, and B.Dong, “NOMA codebook optimization by batch gradient descent,” vol. 7, p. 117274-117281, 2019.
 - [82] D.Yi, S.Ji, and S.Bu, “Un schéma d’optimisation amélioré basé sur des méthodes de descente de gradient pour l’apprentissage automatique,” vol. 11, no. 7, p. 942 (article n°942), 2019.
 - [83] Z.Kaoudi, J.A.Quiané-Ruiz, S.Thirumuruganathan, S.Chawla, and D.Agrawal, “A cost-based optimizer for gradient descent optimization,” *Proceedings of the 2017 ACM International Conference on Management of Data*, p. 977-992, May 2017.
 - [84] F.Saeed, T.Al-Hadhrami, F.Mohammed, and E.Mohammed, *Advances on Smart and Soft Computing*. 2020.
 - [85] X.Cui, W.Zhang, Z.Tüske, and M.Picheny, “Evolutionary stochastic gradient descent for optimization of deep neural networks,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

-
- [86] R.Hallen, "A study of gradient-based algorithms," Mémoire de Master, Université de Lund (Suède), 2017.
 - [87] Z.Q.Zhao, P.Zheng, S.T.Xu, and X.Wu, "Object detection with deep learning: A review," vol. 30, no. 11, p. 3212-3232, 2019.
 - [88] Z.Yang, C.Wang, Z.Zhang, and J.Li, "Mini-batch algorithms with online step size," *Knowledge-Based Systems*, vol. 165, p. 228-240, 2019.
 - [89] L.Liu *et al.*, "On the variance of the adaptive learning rate and beyond," vol. arXiv:1908.03265, 2019.
 - [90] D.Li *et al.*, "AdaError: An adaptive learning rate method for matrix approximation-based collaborative filtering," *Proceedings of the 2018 World Wide Web Conference*, p. 741-751, Apr. 2018.
 - [91] N.N.Mohammed, M.Cawthorne, and A.M.Abdulazeez, "Detection of Genes Patterns with an Enhanced Partitioning-Based DBSCAN Algorithm," *Journal of Information and Communication Engineering*, vol. 4, no. 1, p. 188-195, 2018.
 - [92] D.Q.Zeebaree, A.M.Abdulazeez, O.M.S.Hassan, A.D.Zebari, and J.N.Saeed, "Hiding Image by Using Contourlet Transform," *probablement Journal of Applied Science and Technology*, vol. 83, p. 16979-16990, 2020.
 - [93] M.M.Öztürk, I.A.Cankaya, and D.Ipekci, "Optimizing echo state network through a novel fisher maximization based stochastic gradient descent," *Neurocomputing*, vol. 415, p. 215-224, 2020.
 - [94] H.Sadeeq and A.M.Abdulazeez, "Hardware implementation of firefly optimization algorithm using FPGAs," *International Conference on Advanced Science and Engineering (ICOASE)*, p. 30-35, 2018.
 - [95] D.Q.Zeebaree, H.Haron, A.M.Abdulazeez, and D.A.Zebari, "Machine learning and region growing for breast cancer segmentation," *International Conference on Advanced Science and Engineering (ICOASE)*, p. 88-93, Apr. 2019.
 - [96] F.Bargarai, A.Abdulazeez, V.Tiryaki, and D.Zeebaree, "Management of wireless communication systems using artificial intelligence-based software defined radio," 2020.
 - [97] R.Gylberth, R.Adnan, S.Yazid, and T.Basaruddin, "Differentially private optimization algorithms for deep neural networks," *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, p. 387-394, Oct. 2017.

-
- [98] A.V.Solanke and G.K.Patnaik, "Intrusion Detection using Deep Learning Approach with Different Optimization," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. 5, p. 128-134, 2020.
 - [99] N.Zhang, D. Lei, and J.F.Zhao, "An improved Adagrad gradient descent optimization algorithm," *Chinese Automation Congress (CAC)*, p. 2359-2362, 2018.
 - [100] M.Bai, H.Liu, H.Chen, S.Gu, and Z.Zhang, "An improved algorithm for radar adaptive beamforming based on machine learning," *Journal of Physics: Conference Series*, vol. 1325, no. 1, Oct. 2019.
 - [101] C.E.Nwankpa, "Advances in optimisation algorithms and techniques for deep learning," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 5, p. 563-577, 2020.
 - [102] Z.Qu, S.Yuan, R.Chi, L.Chang, and L.Zhao, "Genetic optimization method of pantograph and catenary comprehensive monitor status prediction model based on adadelta deep neural network," vol. 7, p. 23210-23221, 2019.
 - [103] F.Zou, L.Shen, Z.Jie, W.Zhang, and W.Liu, "A sufficient condition for convergences of Adam and RMSProp," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 11127-11135, 2019.
 - [104] D.A.Zebari, D.Q.Zeebaree, J.N.Saeed, N.A.Zebari, and A.Z.Adel, "Image steganography based on swarm intelligence algorithms: A survey," *People*, vol. 7, no. 8, p. 9, 2020.
 - [105] M.Yaqub *et al.*, "State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images," *Brain Sciences*, vol. 10, no. 7, p. 427, 2020.
 - [106] N.Fatima, "Enhancing performance of a deep neural network: A comparative analysis of optimization algorithms," *Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 9, no. 2, p. 79-90, 2020.
 - [107] S.Vani and T.M. Rao, "An experimental approach towards the performance assessment of various optimizers on convolutional neural network," *3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, p. 331-336, Apr. 2019.
 - [108] D.Yi, J.Ahn, and S.Ji, "An effective optimization method for machine learning based on ADAM," *Applied Sciences*, vol. 10, no. 3, p. 1073, 2020.
 - [109] A.D.E.M.Kemal and S. Kilicarslan, "Performance analysis of optimization algorithms on stacked autoencoder," *3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 1-4, Oct. 2019.

- [110] Y.Jia *et al.*, “Caffe: Convolutional architecture for fast feature embedding,” *Proceedings of the 22nd ACM International Conference on Multimedia*, p. 675-678, Nov. 2014.
- [111] Y.G.Jiang, Z.Wu, J.Tang, Z.Li, X.Xue, and S.F.Chang, “Modeling multimodal clues in a hybrid deep learning framework for video classification,” *IEEE Transactions on Multimedia*, vol. 20, no. 11, p. 3137-3147, 2018.
- [112] H.Chaoui, “Conception et comparaison de lois de commande adaptative à base de réseaux de neurones pour une articulation flexible avec non-linéarité dure,” Mémoire de Master, Université du Québec à Trois-Rivières, 2002.
- [113] É. Clabaut, S. Foucher, Y. Bouroubi, M. Germain, “Synthetic data for Sentinel-2 semantic segmentation,” *Remote Sensing*, vol. 16, no. 5, pp. 1–25, 2024.
- [114] A. Z. Seghir, “Evaluation de la qualité d’image,” Thèse de doctorat, Université de Mentouri–Constantine, 2012.
- [115] M. Azzoug , A. Belaidi, “Algorithmes de Prétraitement d’Images à Multi Copies Bruitées,” Mémoire de Master, Univ bba, 2023.
- [116] A. Chetouani, “Vers un système d’évaluation de la qualité d’image multi-critères,” Thèse de doctorat, Université Paris-Nord-Paris XIII, paris, 2010.