

Rapport bibliographique - Projet PRAT

Débruitage de séries temporelles d'images SAR

Réalisé par Marie Diez

Superviseur : Florence TUPIN

13 février 2021

Table des matières

1	Introduction	3
2	Historique	4
3	Rapport sur la partie bibliographique	5
3.1	A tutorial on synthetic aperture radar (Partie I) [6]	5
3.1.1	Présentation	5
3.1.2	Principe de base	5
3.1.3	La formation du speckle	5
3.2	SAR Image Despeckling by Deep Neural Networks : from a Pre-Trained Model to an End-to-End Training Strategy [1]	6
3.2.1	Utilisation de modèle de débruitage pré-entraîné AWGN	6
3.2.2	Création d'un jeu de données et entraînement d'un CNN (SAR-CNN)	8
3.2.3	Approche hybride	8
3.2.4	Résultats	9
3.3	SAR2SAR : a semi-supervised despeckling algorithm for SAR images [2]	10
3.3.1	Statistique SAR	10
3.3.2	A) Approche auto-supervisé	10
3.3.3	B) Fine tuning	11
3.3.4	C) Étape de raffinement	12
3.3.5	Résultats	12
3.4	Ratio-Based Multitemporal SAR Images Denoising : RABASAR [3]	13
3.4.1	Principe de base	13
3.4.2	Résultats	14
3.5	Exploiting multi-temporal information for improved speckle reduction of Sentinel-1 SAR images by deep learning [4]	15
3.5.1	Résultats	15
3.6	MuLoG, or How to Apply Gaussian Denoisers to Multi-Channel SAR Speckle Reduction ? (Partie I et II) [5]	16
3.6.1	Adaptation des débruiteurs gaussiens au speckle dans le cas "single-channel"	16
4	Application à mon project	17
5	Implémentation réalisée	17
6	Implémentation	18
6.1	Vérification des statistiques du bruit résiduel sur le réseau de base	18
6.2	Réseau de base	20
6.2.1	Dupliquer l'information	21
6.3	Prise en compte des changements temporels	22
6.3.1	Carte binaire	23

6.3.2	Carte en niveau de gris - SC_MAP	23
6.4	Réseau multi-temporel avec prédiction de carte de changements	24
6.4.1	SC_MAP x1x1	24
6.4.2	SC_MAP x1x2	27
6.4.3	SC_MAP x2x1	30
6.5	Invariance du réseau aux permutations temporelles	33
6.5.1	Optimisation de l'hyperparamètre γ	35
6.6	Comparaison des approches implémentées	37
6.6.1	Evaluation quantitative	37
6.6.2	Evaluation visuelle	38
7	Entraînement	39
8	Conclusion	40
9	Bibliographie	41

1 Introduction

Les radars SAR (synthetic aperture radar ou radar à synthèse d'ouverture en français) sont un type particulier de radar qui répondent à des besoins spécifiques. Pour avoir une très bonne résolution spatiale sur les images d'acquisitions, il est important d'avoir une grande antenne (grande focale et donc grande ouverture). Cependant dû aux contraintes physiques évidentes qui ne permettent pas d'envoyer dans l'espace une antenne de plusieurs dizaines de mètres, il est nécessaire de simuler ce phénomène. La synthèse d'ouverture permet de simuler une antenne de cette envergure et donc une grande ouverture pour obtenir des images de hautes résolutions tout en évitant ces contraintes, d'où le nom ouverture de synthèse.

La simulation de cette grande ouverture est le résultat d'un traitement complexe des données brutes avec une combinaison appropriée des signaux reçus. Les images obtenues contiennent en chacun des pixels, deux types d'informations : l'amplitude du signal rétrodiffusé par le sol vers le radar (la réflectivité) ainsi que sa phase, celle-ci dépend du temps de trajet aller retour de l'onde. Ces deux paramètres sont donc liés à la nature du sol observé mais aussi à la distance entre le sol et le satellite. À la différence des capteurs optiques, les radars SAR sont dit "actifs" car ils ne s'appuient pas sur la lumière du soleil pour observer la terre. Ils émettent des ondes électromagnétiques à la surface du globe pour l'illuminer, ce qui les rend insensibles à la couverture nuageuse, donc au climat et au rythme jour/nuit. C'est la puissance avec laquelle un objet réfléchit le signal qui est alors mesurée. L'interprétation des images SAR n'est pas intuitive car elle ne montre pas ce qu'on l'on verrait à l'oeil nu. Mais apporte des informations complémentaires sur les propriétés géométriques et la nature des objets (rugosité de surface / type de matériaux / humidité...) Par exemple les objets urbains auront tendance à réfléchir beaucoup les ondes, car ils sont composés de nombreux angles et de matériaux potentiellement très réfléchissants. Contrairement aux zones naturelles ou matériaux peu réfléchissant qui auront une réflectivité plus faible. Les images SAR ont donc de nombreux avantages et de nombreuses applications, cependant leurs inconvénients majeurs est le bruit particulier qui leur est propre et qui les rend tel quel inexploitable : le phénomène de speckle. Les papiers étudiés à travers ce rapport bibliographique ont pour but d'atténuer ce bruit.

2 Historique

Différentes approches de débruitage du speckle ont été mises en place dans les dernières décennie. Les premières méthodes et très longtemps utilisées étaient des méthodes basées sur la sélection. C'est-à-dire que l'on faisait un moyennage des pixels de différentes vues (chaque image contenant du speckle) et cela permet de diminuer ce bruit. C'est ce qu'on appelle une image multivues ou multilooks en anglais qui consiste à effectuer la moyenne de N images "1 look" obtenue sur la même scène observée. On peut remarquer que l'espérance d'une image "N looks" est la même qu'une image "1 look" :

$$E \left[\frac{1}{N} \sum_{i=1}^N I_i \right] = \frac{1}{N} \sum_{i=1}^N E[I_i] = E[I_i]$$

Cependant la variance est divisé par un facteur N :

$$VAR \left[\sum_{i=1}^N \frac{1}{N} I_i \right] = \frac{1}{N^2} VAR \left[\sum_{i=1}^N I_i \right] = \frac{1}{N} VAR[I_i]$$

Plutôt que d'attendre que le satellite passe plusieurs fois sur la région attendue pour obtenir différentes vues (ce qui serait très long) on peut utiliser une fenêtre de taille $k \times k$ dans une image déterminant le nombre de vues, cependant cela diminuera la résolution spatiale :

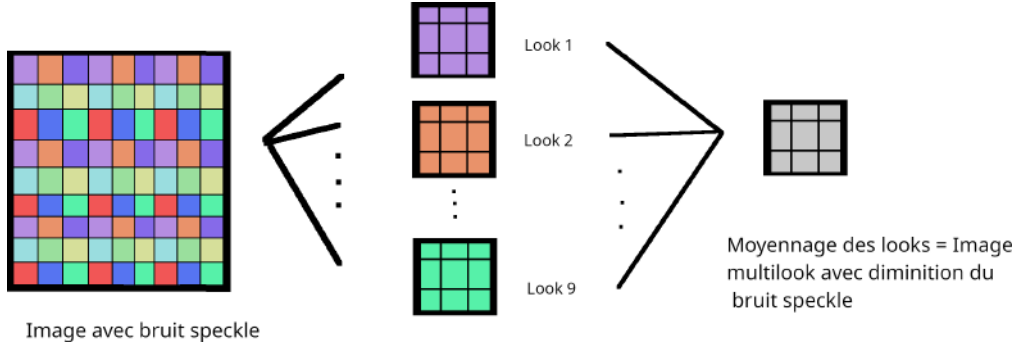


FIGURE 1 – Approche multivues

Des méthodes plus récentes basées sur du deep learning ainsi que l'usage de séries temporelles permettent d'obtenir de meilleures performances et de conserver la résolution spatiale. Cette méthode sert cependant de référence en terme de qualité de débruitage avec l'ENL (equivalent number of look).

3 Rapport sur la partie bibliographique

3.1 A tutorial on synthetic aperture radar (Partie I) [6]

3.1.1 Présentation

Les systèmes SAR sont basés sur un radar pulsé installé sur une plateforme qui se déplace vers l'avant (la direction de vol = azimuth). Le système radar transmet des impulsions électromagnétiques et reçoit les échos du signal rétrodiffusé de manière séquentielle. L'impulsion émise interagit avec la surface de la Terre et seulement une partie est rétrodiffusée vers l'antenne de réception (qui peut être la même que l'antenne d'émission). Les données du signal d'écho reçu forment une matrice de données d'échantillons complexes, où chaque échantillon est constitué de sa partie réelle et sa partie imaginaire, représentant ainsi une valeur d'amplitude et de phase. L'amplitude et la phase du signal rétrodiffusé dépendent des propriétés physiques de la surface observée. Ces systèmes peuvent aussi contenir de nombreuses autres informations. On peut en extraire des paramètres géo-physiques et bio-physiques de la surface terrestre, grâce à une grande flexibilité dans l'orientation du faisceau de l'antenne radar et donc dans la sélection de différents modes d'imagerie : interférométrie, polarimétrie, tomographie...

Polarimétrie

La polarimétrie SAR est une technique largement utilisée et essentielle pour extraire des informations géo-physiques et bio-physiques de différentes surfaces : terre, neige, océans... Cette méthode est basée sur la mesure et l'exploration des propriétés polarimétriques des diffuseurs, en émettant des ondes de différentes polarimétries.

Inférométrie

L'inférométrie SAR est une technique qui permet de mesurer avec une grande précision d'importants paramètres géo-physiques tels que la topographie de surface, la déformation du sol ainsi que les mouvements des glaciers par exemple. L'idée principale de cette méthode est de comparer pour une scène donnée la phase de deux ou plusieurs images SAR qui ont été acquises à partir de positions légèrement différentes ou à des moments différents.

Tomographie

La tomographie SAR cherche à faire une représentation volumique de la surface. Elle est basée sur le concept d'ouverture synthétique, en utilisant plusieurs passages du capteur SAR sur la même zone, mais à des positions différentes (tout comme l'inférométrie). Une application intéressante de la tomographie SAR est dans des zones végétalisées, les résultats peuvent être utilisés pour l'estimation de la topographie du sol sous le feuillage par exemple.

3.1.2 Principe de base

Les systèmes SAR émettent et réceptionnent des ondes électromagnétiques, la combinaison cohérente et appropriée des signaux reçus permet la construction d'une ouverture virtuelle beaucoup plus longue que la longueur de l'antenne physique, elle représente une mesure de la réflectivité de la scène. Ils se déplacent dans la direction de l'azimut, la zone observée à la surface (en jaune sur la Figure 1) est liée à la direction perpendiculaire d'émission et de la durée de la prise de données.

3.1.3 La formation du speckle

Le phénomène de speckle dans les images SAR est dû à la somme de nombreux diffuseurs élémentaires dans une cellule de résolution qui entraîne de fortes fluctuations de la rétrodiffusion d'une cellule de résolution à l'autre. Par conséquent, l'intensité et la phase de l'image finale ne sont plus déterministes. Une propriété importante du speckle est son caractère multiplicatif. Les papiers étudiés par la suite ont pour but de réduire ce phénomène.

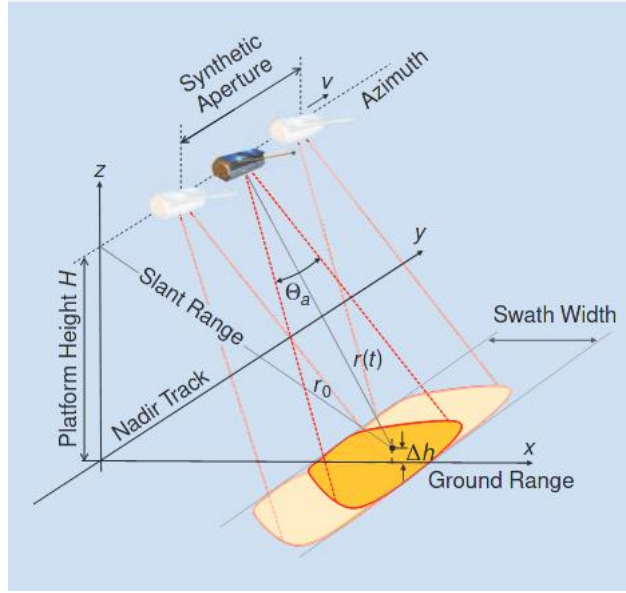


FIGURE 2 – SAR [6]

3.2 SAR Image Despeckling by Deep Neural Networks : from a Pre-Trained Model to an End-to-End Training Strategy [1]

Le problème majeur des approches par deep learning pour la suppression du bruit speckle dans des images SAR est le manque de données débruitées (speckle-free en anglais). Pour cela, ce papier analyse différentes stratégies et notamment l'intérêt de construire un jeu de données d'images sans speckle, celui-ci propose 3 méthodes pour cette étude :

- Utiliser des modèles de débruitage de bruit additif blanc gaussien (Additif white gaussian noise - AGWN) sur des images naturelles à un cadre de suppression du speckle dans des images SAR.
 1. Approche homographique
 2. Approche variationnelle : filtrage itératif avec MuLog
- Construire un jeu de données d'images SAR sans speckle pour entrainer un modèle de type CNN.
- Méthodes hybrides : Entrainer un CNN de débruitage AWGN à partir du jeu de données sans speckle.

Le deep learning permet d'apprendre un modèle, une représentation des données avec plusieurs niveaux d'abstraction, avec sa démocratisation ces dernières années de nombreux algorithmes de débruitage AWGN dans des images naturelles ont été mis en place et de plus en plus d'algorithmes de débruitage de speckle basés sur du deep learning ont vu le jour.

3.2.1 Utilisation de modèle de débruitage pré-entraîné AWGN

Le bruit peut se modéliser sous 2 formes :

- Bruit additif : $y = x + n$
- Bruit multiplicatif : $y = x * n$

Avec y l'image bruitée, x l'image sans bruit et n la composante de bruit.

De nombreuses approches ont été mises en place pour débruiter des images et plus récemment des réseaux résiduels sont utilisés, où l'on cherche à estimer la composante de bruit \hat{n} tel que $\hat{x} = y - \hat{n}$.

Le bruit gaussien est un bruit additif, cependant le bruit de type speckle est multiplicatif et suit une

distribution gamma selon le modèle de Goodman's :

$$p(n) = \frac{L^L}{\Gamma(L)} n^{L-1} \exp(-Ln)$$

Avec $L \geq 1$ le nombre de vues, et $\Gamma(\cdot)$ la fonction gamma.

Pour obtenir un bruit additif à partir d'un bruit multiplicatif il faut passer dans le domaine logarithmique. On obtient alors $\tilde{y} = \tilde{x} + \tilde{n}$ où $\tilde{\cdot}$ représente le logarithme. Le log speckle \tilde{n} suit alors une distribution Fisher-Tippett :

$$p(\tilde{n}) = \frac{L^L}{\Gamma(L)} e^{L\tilde{n}} \exp(-Le^{\tilde{n}})$$

Les modèles de débruitages AWGN doivent alors être adaptés à cette distribution différente du modèle gaussien habituel. Différentes approches ont pu être mises en place mais le papier se concentre sur un modèle résiduel de débruitage AWGN précis : DnCNN.

1. Approche homographique

L'idée est de réduire la distribution Fisher-Tippett au cas gaussien non-centré en ajoutant un biais :

$$\hat{x} = f_{\psi(1,L)}(y) + \underbrace{(\log L - \Psi(L)) \mathbf{1}_n}_{\text{debiasing}}$$

où $f_{\psi(1,L)}(y)$ est un débruiteur gaussien.

Le passage dans le domaine logarithmique permet d'obtenir un bruit additif, cependant il faut faire attention à ajouter une étape exponentielle pour revenir dans le domaine original :

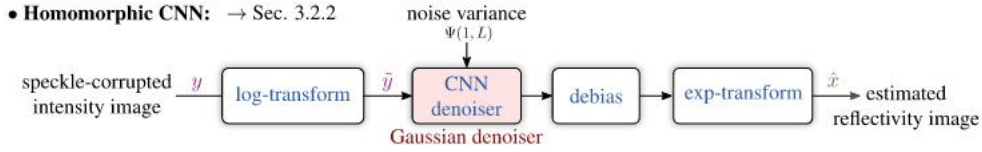


FIGURE 3 – Approche homographique [1]

2. Approche variationnelle : filtrage itératif avec MuLog

L'approche variationnelle voit le problème de débruitage comme un problème d'optimisation. L'idée est de prendre en compte directement la distribution Fisher-Tippett avec MuLog, à travers un schéma itératif qui alterne un débruiteur gaussien AGWN (DnCNN par exemple) et une étape de correction non linéaire dans le but d'optimiser (itérativement) une fonction prenant directement en compte la distribution gamma (ou Fisher-Tippett dans le domaine logarithmique). Voir MuLog 3.6.1

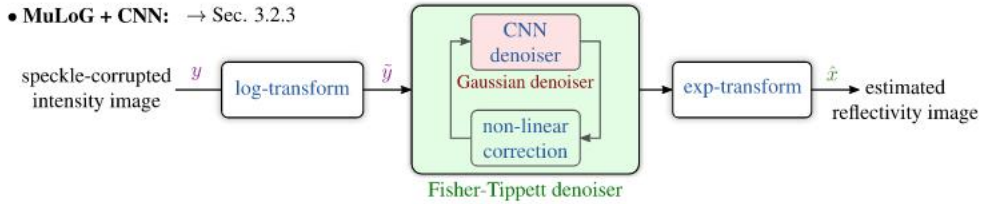


FIGURE 4 – Approche par filtrage itératif avec MuLog+CNN [1]

Remarque : Lorsque que le nombre de vues est important, il n'est pas nécessaire de passer par des débruiteurs spécifiques, comme le montre les courbes ci-dessous, un débruiteur classique AGWN serait suffisant.

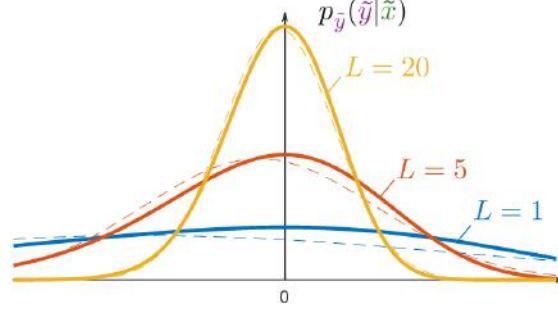


FIGURE 5 – Comparison de la distribution Fisher-Tippett en ligne continue et la distribution gaussienne en pointillé avec différents nombres de vues L [1]

3.2.2 Création d'un jeu de données et entraînement d'un CNN (SAR-CNN)

Passer par la création d'un jeu de données spécifique aux images SAR permet d'éviter de modéliser les statistiques spéciales liées à ces données. Cependant, la rareté des données brutes sans bruit rend compliqué la création d'un tel jeu.

Le papier propose de débruiter des images SAR par des approches multivues et avec le modèle Mu-Log+BM3d pour construire le jeu de données de vérités-terrains. Le bruit speckle synthétique est généré sur la base des modèles statistiques SAR afin de produire les paires d'images nécessaires pour l'entraînement. Cependant, le modèle de Goodman's suppose un speckle non corrélé dans l'espace, cela impose donc de faire une hypothèse sur la non-corrélation du speckle dans les images, ce qui n'est pas le cas en réalité. Pour cela le débruitage d'images réelles nécessitera donc de supprimer les corrélations spatiales par un sous-échantillonnage ce qui entraînera une perte de résolution spatiale.

Différentes architectures et fonctions d'erreurs peuvent être utilisées, d'après le papier une augmentation de la profondeur du réseau avec une loss l1 : $\sum_{i=1}^N \|f_{CNN}(\tilde{y}_i) - \tilde{x}_i + (\psi(L) - \log(L)) \cdot \mathbf{1}\|_1$ serait intéressante à modifier.

Application of a CNN trained specifically for speckle reduction:

- Approach proposed by Chierchia *et al.* and this paper: → Sec. 3.3

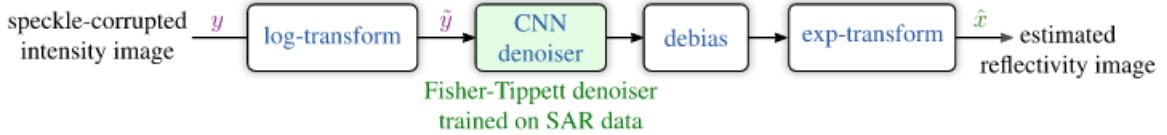


FIGURE 6 – Création d'un jeu d'entraînement et apprentissage d'un CNN [1]

3.2.3 Approche hybride

Les approches hybrides consistent à utiliser les approches homographiques et variationnelles avec un entraînement des débruiteurs AWGN sur le jeu de données précédemment construit, afin de voir l'influence du jeu d'entraînement sur les résultats.

3.2.4 Résultats

Après de nombreux tests de qualité en fonction des index PSNR (Peak-signal-to-noise ratio), SSIM (Structural Similarity Index) et des résultats visuels, le papier montre que sur des images avec génération de speckle la méthode SAR-CNN est la plus intéressante en terme de performances. La méthode MuLog+CNN (pré-entraîné ou non) dépasse aussi la méthode de référence : MuLog+BM3d. Pour des images réelles ces 2 méthodes sont aussi plus performantes que la méthode de référence, l'estimateur de performance utilisé est l'ENL (estimation of number of look) étant donné qu'il n'y a pas de vérités-terrains. Sur des images multi-channels il est plus intéressant d'opter pour MuLog+CNN qui est plus performant car il est fait pour appliquer une restauration à des images SAR multi-channels et qu'il est difficile pour SAR-CNN d'apprendre les corrélations entre les channels. L'entraînement sur un jeu de données d'images SAR montre en effet de meilleures performances. Cependant, seul un petit gain de performance est observé lorsque MuLoG+CNN est pré-entraîné sur des images SAR. Ainsi, si la création d'un jeu de données est possible, il est conseillé de l'utiliser pour entraîner un modèle de bout en bout tel que le SAR-CNN. Les approches homographique, variationnelle et SAR-CNN ne prennent cependant pas en compte la corrélation spatiale du speckle.

3.3 SAR2SAR : a semi-supervised despeckling algorithm for SAR images [2]

Le problème de nombreux algorithmes de débruitage du speckle est qu'ils ne tiennent souvent pas compte d'un phénomène trop difficile à modéliser : la corrélation spatiale du speckle, ce qui entraîne une perte de résolution. Le but de SAR2SAR est de prendre en compte cette corrélation. Pour cela il va faire appel à un apprentissage semi-supervisé où dans un premier temps le réseau va apprendre sur des données avec une génération du speckle puis sera affiné sur des images réelles. Le problème des changements temporels sera abordé par une stratégie de compensation.

SAR2SAR est une extension de noise2noise qui prends en compte les statistiques spécifiques du speckle.

3.3.1 Statistique SAR

Le bruit speckle est un bruit multiplicatif suivant une distribution gamma selon le modèle de Goodman's :

$$p(n) = \frac{L^L}{\Gamma(L)} n^{L-1} \exp(-Ln)$$

Avec $L \geq 1$ le nombre de vues, et $\Gamma(\cdot)$ la fonction gamma.

Pour obtenir un bruit additif à partir d'un bruit multiplicatif il faut passer dans le domaine logarithmique. On obtient alors $\tilde{y} = \tilde{x} + \tilde{n}$ où $\tilde{\cdot}$ représente le logarithme. Le log speckle \tilde{n} suit alors une distribution Fisher-Tippett :

$$p(\tilde{n}) = \frac{L^L}{\Gamma(L)} e^{L\tilde{n}} \exp(-Le^{\tilde{n}})$$

Cependant, le modèle de Goodman's ne prends pas en compte la corrélation spatiale du speckle. D'où l'intérêt d'une approche semi-supervisé pour prendre en compte cette corrélation par un affinement du modèle.

L'algorithme SAR2SAR est basé sur plusieurs étapes d'apprentissage :

- A) Apprentissage dans un cadre auto-supervisé
- B) Affinage (fine tuning sur des images réelles et prise en compte des changements temporels)
- C) Étape de raffinement (prise en compte des corrélations spatiales)

3.3.2 A) Approche auto-supervisé

L'algorithme SAR2SAR repose sur noise2noise [7] qui évolue dans un cadre auto-supervisé.

La différence avec un cadre supervisé réside dans l'accès aux vérités-terrains :

Dans un cadre supervisé le but est de trouver les paramètres θ tel que $f_\theta : y \rightarrow \hat{x} = f_\theta(y)$ en minimisant une fonction d'erreur :

$$\hat{\theta}_{\text{supervised}}^{(\ell_2)} \in \underset{\theta}{\operatorname{argmin}} E_X [||f_\theta(y) - x||^2]$$

Cependant dans un cadre auto-supervisé la donnée x n'est pas disponible. L'algorithme noise2noise propose alors d'utiliser 2 images bruitées y_1 et y_2 tel que :

$$\hat{\theta}_{\text{self-supervised}}^{(\ell_2)} \in \underset{\theta}{\operatorname{argmin}} EX [||f_\theta(y_1) - y_2||^2]$$

L'approximation $\hat{\theta}_{\text{self-supervised}}^{(\ell_2)} = \hat{\theta}_{\text{supervised}}^{(\ell_2)}$ fonctionne car le bruit est centré, cela permet alors si le jeu de données est conséquent d'obtenir les paramètres θ en fonction d'un apprentissage auto-supervisé. Pour converger plus vite vers cette équivalent SAR2SAR propose d'utiliser fonction d'erreur de type log-likelihood :

$$\hat{\theta}_{\text{self-supervised}}^{(\text{lik})} \in \underset{\theta}{\operatorname{argmin}} E_{\mathbf{X}} [-\log \mathbf{p}(\mathbf{y}_2 | f_\theta(\mathbf{y}_1))]$$

Pour prendre en compte les statistiques spécifiques du speckle, il faut adapter la fonction d'erreur pour prendre en compte la distribution Fisher-Tipette suivie par $p_{Y|X}(y|x)$:

$$\ell(f_\theta(y_1), y_2) = -\log \mathbf{p}(y_2 | f_\theta(y_1)) = \sum_k f_\theta([y_1]_k) - [y_2]_k + \exp([y_2]_k - f_\theta([y_1]_k)) \quad (1)$$

L'idée est alors de générer du bruit speckle sur 2 images y_1 et y_2 et de minimiser cette fonction d'erreur (1) entre l'image de sortie $f_\theta(y_1) = \hat{x}_1$ et la seconde image y_2 .

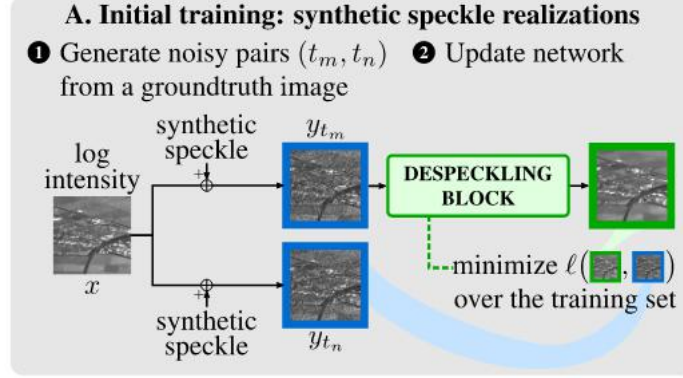


FIGURE 7 – Étape A de l'algorithme SAR2SAR [2]

3.3.3 B) Fine tuning

Cette étape permet de prendre en compte les changements temporels grâce à une stratégie de compensation dans l'image y_2 tel que : $y_2 = y_2 - \hat{x}_2 + \hat{x}_1$. Il faut alors minimiser la différence entre $f_\theta(y_1)$ et y_2 . Le premier bloque de débruitage de l'algorithme A) a été entraîné sur images avec du bruit généré selon le modèle de Goodman's ne prenant donc pas en compte la corrélation spatiale du speckle. Lorsque l'on donne des images réelles à l'algorithme il faut supprimer la corrélation existante afin que l'algorithme se concentre sur l'apprentissage des changements temporels. Il faut alors sous-échantillonner (puis sur-échantillonner) les images par un facteur λ .

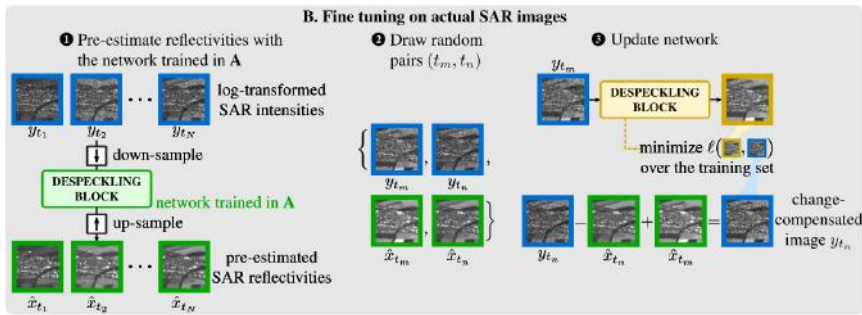


FIGURE 8 – Étape B de l'algorithme SAR2SAR [2]

3.3.4 C) Étape de raffinement

Cette étape est une étape de raffinement, c'est la même structure que l'étape B mais avec le réseau de B et non de A comme bloque de débruitage. Le but est de prendre en compte la corrélation spatiale du speckle sur des images réelles, il faut donc supprimer le sous-échantillonnage de l'étape B.

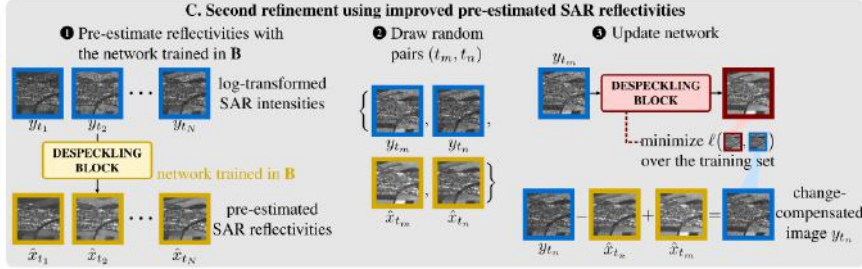


FIGURE 9 – Étape C de l'algorithme SAR2SAR [2]

3.3.5 Résultats

Les résultats en fonction de l'index PSNR montre que SAR2SAR obtient de très bons résultats, supérieurs à ceux de SAR-CNN vu précédemment. On peut de plus remarquer que les étapes A) et B) de l'algorithme sont cruciales pour avoir de bonnes performances et l'étape C) pour prendre en compte la corrélation spatiale du speckle. Les performances sont particulièrement remarquables pour des images 1 vue, SAR2SAR est alors très intéressant pour des images avec peu de vues. En effet plus le nombre de vues augmente moins il est intéressant de passer par des débruiteurs spécifiques SAR.

3.4 Ratio-Based Multitemporal SAR Images Denoising : RABASAR [3]

Grâce à la constellation de satellites Sentinel-1, l'accès des séries multi-temporelles d'images SAR est devenu plus important et cela a ouvert de nouvelles portes aux méthodes de débruitages du speckle. En effet, les séries temporelles offrent l'accès à des informations redondantes dans les séries et peuvent être exploitées pour améliorer le débruitage, en s'inspirant notamment des méthodes multivues. Contrairement aux méthodes précédentes qui n'utilisent pas les séries temporelles en entrée mais des "singles images", ici le papier aborde le débruitage d'images multi-temporelles SAR basé sur le ratio afin d'exploiter ces nouvelles informations. L'objectif est d'avoir une image moyennée qui permet de diminuer le speckle et donc d'avoir de potentiels meilleurs résultats de débruitage, en temps normal cela entraînerait une perte de résolution trop importante, les séries temporelles permettent ici d'éviter ce problème.

3.4.1 Principe de base

Les séries temporelles vont permettre de mettre en place une approche multivues temporelles qui permet donc de diminuer le speckle grâce au multivue tout en préservant la résolution grâce aux séries temporelles (la moyenne étant faite sur l'axe temporel) cette image est appelée la super image \hat{u}_m :

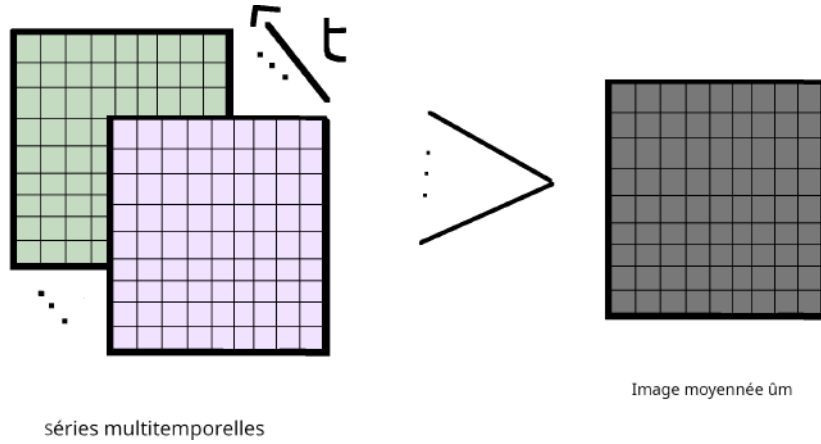


FIGURE 10 – Approche multivues sur l'axe temporel

Ce papier propose d'exploiter la super-image pour construire un débruiteur d'images SAR multi-temporels basé sur le ratio. La méthode RABASAR est constituée de différentes étapes :

1. Calcul de la super image \hat{u}_m . Il faut cependant faire attention à choisir des dates où le changement est peu visible pour négliger les fluctuations temporelles.
2. Calcul de l'image ratio : $\tau_t = \frac{v_t}{\hat{u}_m}$
3. Débruitage de $\tau_t \longrightarrow \rho_t$
4. Calcul de la nouvelle image $u_t = \hat{u}_m * \rho_t$

1. Calcul de la super image

Il existe différentes façons de calculer la super-image, le papier va en étudier 2 :

- Moyenne arithmétique - AM :

$$\hat{u}_m^{\text{AM}}(s) = \frac{1}{T} \sum_{t=1}^T v_t(s) \quad 1 \leq t \leq T$$

- Moyenne arithmétique binaire pondérée - BWAM (seulement les échantillons avec une réflectivité similaires et stables sont comptés, un poids est donc associé à chaque image de la série) :

$$\hat{u}_m^{\text{BWAM}}(s) = \frac{1}{\sum_{t'=1}^T w_{t,t'}(s)} \sum_{t=1}^T w_{t,t'}(s) v_{t'}(s)$$

Une étape de débruitage de la super-image peut aussi être mise en place, *MuLog – BM3d* est utilisé pour cela.

2. Calcul du ratio

Le ratio est défini comme suit :

$$\tau_t = \frac{v_t}{\hat{u}_m}$$

Avec v_t l'image originale bruitée et \hat{u}_m la super-image.

3. Débruitage du ratio

Le ratio τ_t doit alors être débruité, cependant comme il reste des fluctuations du speckle dans la super-image \hat{u}_m le bruit multiplicatif de celui-ci ne suit alors pas une distribution gamma mais une distribution Fisher-Tippett. Pour obtenir un bruit additif on passe dans le domaine logarithmique et le log speckle suit alors une distribution z-Fisher pdf. Contrairement au cas habituel dans lequel MuLog est appliqué avec un speckle suivant une distribution gamma et le log speckle suivant une distribution Fisher-Tippett. Pour le débruitage l'algorithme MuLog doit alors être modifié légèrement en prenant en compte cela, le papier à alors mis en place un débruiteur spécifique RuLog, on obtient ρ_t le ratio débruité. Dans le cas où la série temporelle est très grande on peut estimer que l'image \hat{u}_m ne contient plus de fluctuations et alors MuLog serait suffisant, cependant si la série n'est pas suffisamment grande il est important d'utiliser RuLog.

Étant donné que MuLog est entraîné sur des images avec génération du speckle il est nécessaire de mettre en place une étape de sous-échantillonnage du ratio pour supprimer la corrélation spatiale en cas de réelle image SAR.

4. Calcul de la nouvelle image

La nouvelle image est calculée comme suit : $u_t = \hat{u}_m * \rho_t$

3.4.2 Résultats

Finalement les tests menés par le papier montre que RABASAR offre de meilleures performances en terme d'index PSNR et SSIM que les autres méthodes de références et que l'intérêt du débruitage de la super-image est intéressant si le nombre d'images dans la série temporelles est faibles. Pour finir, la moyenne BWAM (ou DBWAM) est intéressante à utiliser plutôt que AM (ou DAM) si il y a des fluctuations temporelles dans la série.

Cette approche permet de prendre en compte la redondance des informations pour améliorer les performances, mais pas la corrélation spatiale à cause de l'algorithme de débruitage qui ne la prends pas en compte, ce qui entraine une perte de résolution. Pour répondre à ce besoin la méthode RABASAR-SAR2SAR décrite ci-dessous à été mise en place.

3.5 Exploiting multi-temporal information for improved speckle reduction of Sentinel-1 SAR images by deep learning [4]

Depuis le lancement des satellites de la mission Sentinel-1 des longues séries temporelles d'images SAR sont disponibles. Cela a ouvert les portes à de nouvelles méthodes de débruitages, par exemple RABASAR.

De nombreuses méthodes de débruitages de speckle sont entraînées sans prendre en compte les corrélations spatiales du speckle et nécessitent donc dans le cas de données réelles de supprimer cette corrélation, avec par exemple un sous-échantillonnage. L'algorithme SAR2SAR permet de résoudre ce problème en prenant en compte cette corrélation avec un entraînement spécifique sur des images réelles. RABASAR-SAR2SAR permet alors d'utiliser la redondance des informations grâce aux séries temporelles et prends en compte la corrélation spatiale du speckle, pour tenter d'améliorer les performances.

L'idée de ce papier est d'étendre l'algorithme SAR2SAR en exploitant les informations multi-temporelles, c'est-à-dire de combiner l'algorithme SAR2SAR avec RABASAR.

L'extension est donc basée sur RABASAR, méthode décrite précédemment avec 4 étapes majeures :

1. Calcul de la super image \hat{u}_m . Il faut cependant faire attention à choisir des dates où le changement est peu visible pour négliger les fluctuations temporelles.
2. Calcul de l'image ratio : $\tau_t = \frac{v_t}{\hat{u}_m}$
3. Débruitage de $\tau_t \rightarrow \rho_t$
4. Calcul de la nouvelle image $u_t = \hat{u}_m * \rho_t$

Lors du débruitage de l'image ratio, si la super-image ne contient plus de fluctuations du speckle alors on peut utiliser l'algorithme MuLog pour le débruitage. Cependant si elle contient encore des fluctuations du speckle il est nécessaire de prendre en compte la distribution spécifique du ratio avec RuLog (comme expliqué précédemment 3.4.1). Le problème de ces 2 méthodes c'est qu'elles ne prennent pas en compte la corrélation spatiale du speckle et donc sur des images réelles il faut supprimer cette corrélation, en sous-échantillonnant par exemple. Cela entraîne une perte de résolution, l'idée est d'utiliser ici l'algorithme SAR2SAR pouvant préserver la corrélation spatiale et donc la résolution.

Cependant il faut adapter SAR2SAR aux images ratios, en effet les ratios ont une gamme dynamique beaucoup plus faible que les intensités des images SAR. Et les réseaux sont très sensibles à la gamme dynamique des images. Afin de préserver la gamme originale de u dans le domaine logarithmique quand on calcule le ratio τ , il faut normaliser la super-image $\hat{u}_m \rightarrow \hat{u}_m'$:

1. Calcul de la super image \hat{u}_m'
2. Calcul de l'image ratio : $\tau_t' = \frac{v_t}{\hat{u}_m'}$
3. Débruitage de $\tau_t' \rightarrow \rho_t'$
4. Calcul de la nouvelle image $u_t' = \hat{u}_m' * \rho_t'$

3.5.1 Résultats

La combinaison de RABASAR et SAR2SAR sur des séries temporelles d'images SAR améliore la qualité de la restauration.

3.6 MuLoG, or How to Apply Gaussian Denoisers to Multi-Channel SAR Speckle Reduction ? (Partie I et II) [5]

Le but du papier est d'inclure les débruiteurs gaussiens dans un contexte de débruitage du speckle pour des images multi-channels SAR.

Une image "single-channel" correspond à l'intensité (réflectivité) de l'image SAR, on peut cependant obtenir de nombreuses autres informations : interférométriques, polarimétriques, tomographiques... Comme expliqué au début 3.1.1

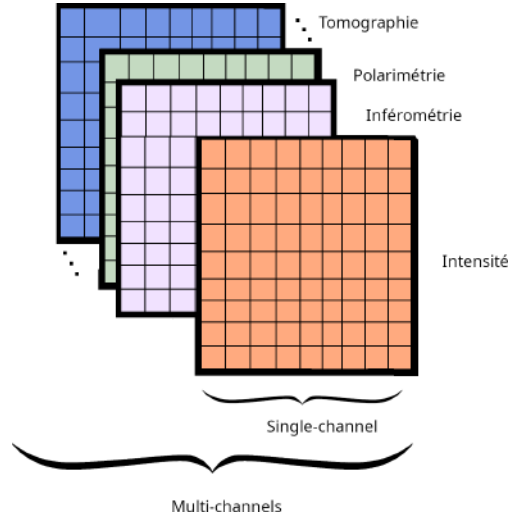


FIGURE 11 – Image SAR single/multi-channel(s)

L'idée du papier est alors d'étendre les débruiteurs gaussiens aux phénomènes du speckle à travers la prise en compte de sa distribution particulière ainsi que d'étendre le débruitage d'images SAR "single-channel" au débruitage multi-channels.

3.6.1 Adaption des débruiteurs gaussiens au speckle dans le cas "single-channel"

Approche homographique

L'idée est d'approcher la distribution de Fisher-Tippett par une distribution gaussienne non centrée, cela conduit à la procédure d'estimation suivante :

$$\hat{x} = f_{\psi(1,L)}(y) + \underbrace{(\log L - \Psi(L))\mathbf{1}_n}_{\text{debiaing}}$$

où $f_{\psi(1,L)}(y)$ est un débruiteur gaussien. Voir schéma 3.2.1.

Approche variationnelle

Plutôt que d'appliquer l'approche homographique afin de réduire le problème original au cas gaussien, une alternative propose d'optimiser une fonction prenant directement en compte la distribution gamma de l'image bruitée I :

$$\hat{R} \in \operatorname{argmin}_{R \in R_+^n} -\log p_I(I | R) + \mathcal{R}(R), \quad \text{où} \quad -\log p_I(I | R) = L \sum_{k=1}^n \log R_k + \frac{I_k}{R_k} + \text{Cst.}$$

On peut aussi utiliser cette approche dans le domaine logarithmique où l'on doit alors approcher la distribution Fisher-Tippett.

Cette procédure d'optimisation s'effectue par un schémas itératif présenté ici 3.2.1.

On peut par la suite étendre la méthode au cas multi-channels, ce que je ne vais pas détailler ici car cela sort de l'objectif du projet.

4 Application à mon projet

Le but de mon projet est d'étudier la pertinence d'utiliser une petite série temporelle (2 ou 3 images) en entrée d'un réseau de type SAR-CNN. Dans l'objectif de voir si la redondance des informations fournies par la petite série temporelle peut améliorer les performances de débruitage. Notamment en forçant le réseau à prendre en compte les nouvelles informations avec d'une carte de changements.

Voici le plan :

1. Comprendre et tester le modèle SAR-CNN
2. Tester le modèle SAR-CNN avec 2 images d'une série temporelle en entrée et voir si le réseau est capable de prendre en compte les nouvelles informations.
3. Forcer le réseau à prendre en compte ces nouvelles informations en mettant en place une carte de changements. Il faudra réfléchir à une solution efficace de forcer le réseau à prendre en compte cette carte à travers la fonction de coût.
4. Comparer et évaluer les performances des méthodes testées.

5 Implémentation réalisée

Lors de ce projet j'ai travaillé sur les points détaillés en partie 4.

1. Dans un premier temps, nous avons mis en place les résultats de références, qui sont les résultats du réseau au début du projet. Ce réseau sera nommé par la suite *Réseau de base*.
2. Par la suite nous avons ajouté de l'information au réseau en dupliquant l'informations, c'est-à-dire en lui donnant 2 fois la même date en entrée avec 2 simulations de bruit différentes, afin de voir si l'ajout d'informations aide le réseau à améliorer les résultats.
3. Ensuite, nous avons mis en place une carte de changements pour forcer le réseau à prendre en compte les nouvelles informations. Ce réseau sera nommé par la suite *SC_MAP*. Deux possibilités de formulation de la carte de changements se sont présentées :
 - carte de changements binaire
 - carte de changements en niveau de gris
4. Nous avons jugé plus intéressant de continuer à travailler sur une carte de changements en niveau de gris. A partir de là nous avons mis en place différentes expérimentations :
 - Regarder si la prédiction de la carte de changements entre 2 mêmes dates avec 2 simulations de bruits différentes est nulle.
 - Regarder et comparer les performances de débruitage du réseau avec la prédiction de la carte de changements avec 2 dates différentes bruitées en entrée.
 - Regarder si le réseau est invariant aux permutations en regardant les prédictions pour les dates inversées. Il s'avère que le réseau n'est pas invariant aux permutations et nous avons alors modifié le réseau pour le rendre invariant en mettant en place un réseau siamois. Ce réseau sera nommé par la suite *SCM_SIAMESE*.

6 Implémentation

6.1 Vérification des statistiques du bruit résiduel sur le réseau de base

Le bruit de type speckle est un bruit multiplicatif et suit une distribution gamma selon le modèle de Goodman's :

$$p(n) = \frac{L^L}{\Gamma(L)} n^{L-1} \exp(-Ln)$$

Avec $L \geq 1$ le nombre de vues, et $\Gamma(.)$ la fonction gamma.

Pour obtenir un bruit additif à partir d'un bruit multiplicatif nous sommes passé dans le domaine logarithmique. On obtient alors $\tilde{y} = \tilde{x} + \tilde{n}$ où $\tilde{\cdot}$ représente le logarithme. Le log speckle \tilde{n} suit alors une distribution Fisher-Tippett :

$$p(\tilde{n}) = \frac{L^L}{\Gamma(L)} e^{L\tilde{n}} \exp(-Le^{\tilde{n}})$$

Cela est valable lorsque l'on travaille avec des images en intensité, pour des images en amplitude le bruit suit la loi de Rayleigh : $p(n) = \frac{n}{2\gamma} \exp\left(-\frac{n^2}{4\gamma}\right)$

Comme nous travaillons avec des images en amplitude, pour vérifier les statistiques du bruit nous allons travailler avec des images en intensité, tel que :

$$A = \sqrt{I} \Leftrightarrow I = A^2$$

Avec I l'image en intensité et A l'image en amplitude.

L'image bruitée y est alors normalisée avec la fonction *normalize_sar* avant de passer dans le réseau. En sortie, il faut alors dénormalisée avec : *denormalize_sar*.

Pour évaluer la qualité du débruitage pour les 2 images ci-dessous, on peut s'intéresser aux méthodes suivantes :

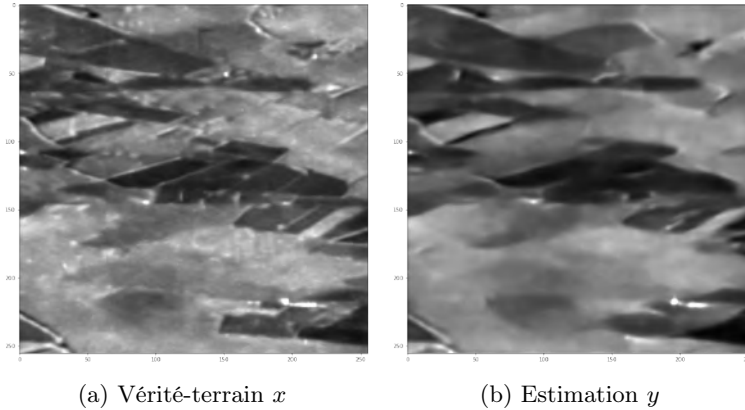


FIGURE 12 – Image d'évaluation Marais1_10

- Bruit résiduel

Le bruit résiduel est le bruit qui a été supprimé lors du débruitage, c'est donc du speckle pur qui doit suivre en intensité, la distribution Gamma. Le bruit résiduel n_{res}^2 dans cette image, suit alors la distribution Gamma, tel que :

$$n_{res}^2 = \left(\frac{x_{noisy}}{y}\right)^2 \sim \Gamma(.)$$

$$E[n] = 1 \text{ et } Var[n] = 1/L$$

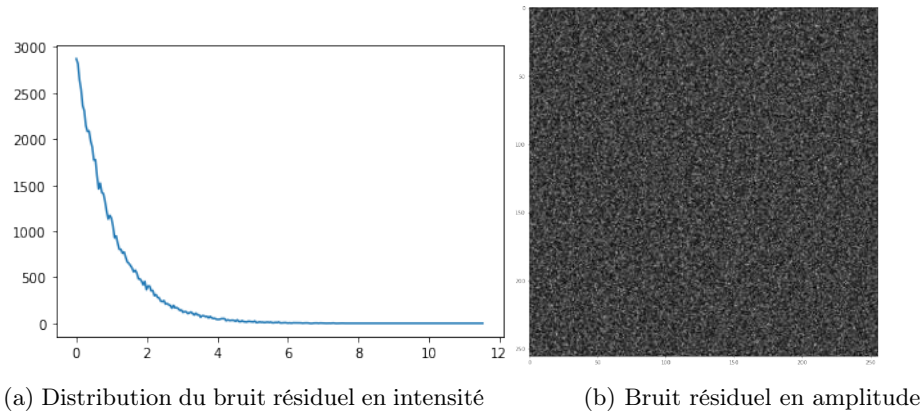


FIGURE 13 – Bruit résiduel prédit par le réseau de base

Avec $\mu = 0.99$ et $\sigma^2 = 0.95$

Nous retrouvons alors bien les statistiques du speckle attendu.

- Différence entre estimation et vérité-terrain

La différence suivante nous permet de regarder la qualité structurelle du débruitage, notamment de voir les structures résiduelles qui n'ont pas été retrouvées lors du débruitage.

$$\text{diff} = |y - x|$$

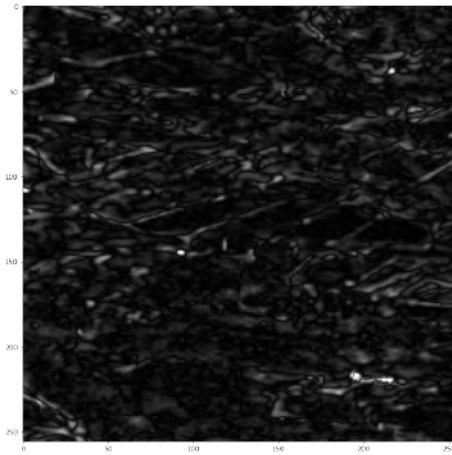


FIGURE 14 – Différence absolue entre l'estimation et la vérité-terrain

- Ratio entre estimation et vérité-terrain

Sur le ratio suivant on peut trouver le bruit restant dans l'image estimée ainsi que les structures perdu lors du débruitage, celles qui sont dans la vérité-terrain et non dans l'estimation. Celles-ci sont plus facilement visibles sur ce ratio que sur la différence précédente.

$$\text{ratio} = \frac{y}{x}$$

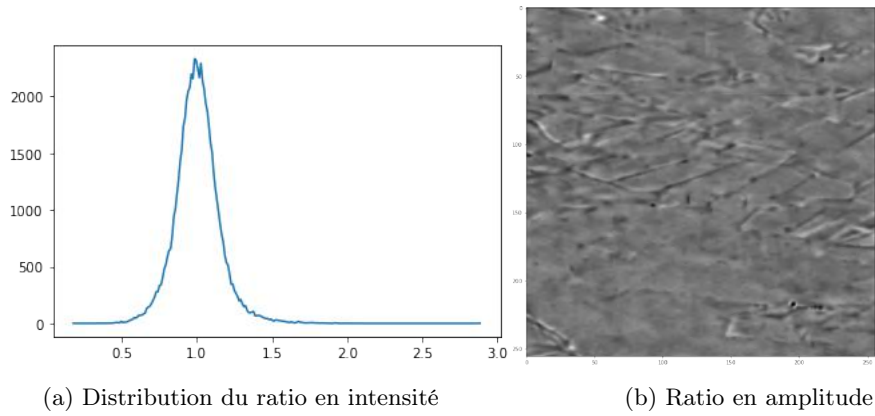


FIGURE 15 – Ratio

Avec $\mu = 1.00$ et $\sigma^2 = 0.02$

On peut retrouver une moyenne de 1 et une variance faible, car l'estimation contient effectivement peu de bruit.

On voit bien des structures résiduelles qui indiquent que le débruitage fait perdre des détails structurels à la prédiction par rapport à la vérité-terrain. Le but de ce projet est d'améliorer le débruitage de l'estimation par rapport à la vérité-terrain. C'est-à-dire diminuer le plus possible le bruit et limiter la perte de structures.

6.2 Réseau de base



FIGURE 16 – Réseau de base

Le réseau de base que Inès Meraoumia m'a fourni au début de ce projet est un réseau de type Unet. Il prend en entrée une image bruitée et prédit l'estimation débruitée, de façon à minimiser la différence absolue avec la vérité-terrain, la fonction de coût est la suivante :

$$Loss = \left(\frac{1}{batch_size} \sum_{i=0}^{N-1} |X_i - Y_i + cn| \right)$$

Avec X le batch de la vérité-terrain, Y le batch de l'estimation débruitée et cn une constante pour corriger les biais du réseau.

Les batches sont formés de plusieurs patches de l'image à débruiter de la façon suivante :

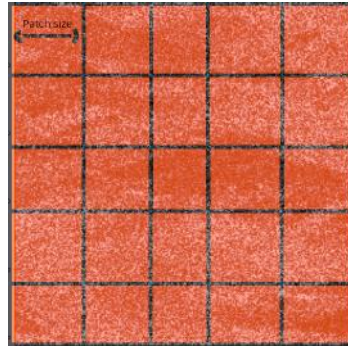


FIGURE 17 – Construction du batch

Chaque patch à une dimension de $patch_size * patch_size$ et la taille du batch est alors de $\frac{image_size}{patch_size}$. Le réseau va alors débruiter chaque patch avant de reconstruire l'image correctement.

6.2.1 Dupliquer l'information

On peut s'intéresser au comportement du réseau lorsque que l'on duplique l'information bruitée en entrée. On s'attend à avoir une meilleure définition des structures dans l'estimation étant donné que les images données au réseau sont les mêmes, seul le bruit diffère. Nous allons donc donner en entrée de notre réseau une pile contenant 2 fois la même image temporelle $x1$ de même temps $t1$ mais avec 2 simulations de bruit différentes :

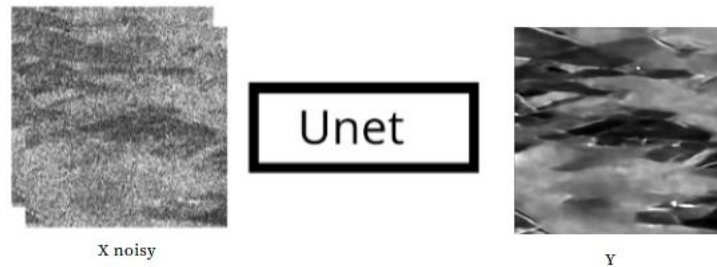


FIGURE 18 – Réseau de base avec duplication d'informations en entrée

Evaluation quantitative

Voici une évaluation quantitative basée sur les indices PSNR et SSIM dont la définition est donnée en partie 6.6.1.

evaluation	Réseau de base	Réseau de base x1x1
Mairais1_10 PSNR	31.549	33.067
Mairais1_10 SSIM	0.862	0.888

test	Réseau de base	Réseau de base x1x1
Limagne1_10 PSNR	28.577	30.481
Limagne1_10 SSIM	0.840	0.876

Evaluation visuelle

L'évaluation quantitative n'est pas suffisante à elle-même, il est intéressant de comparer visuellement les résultats :

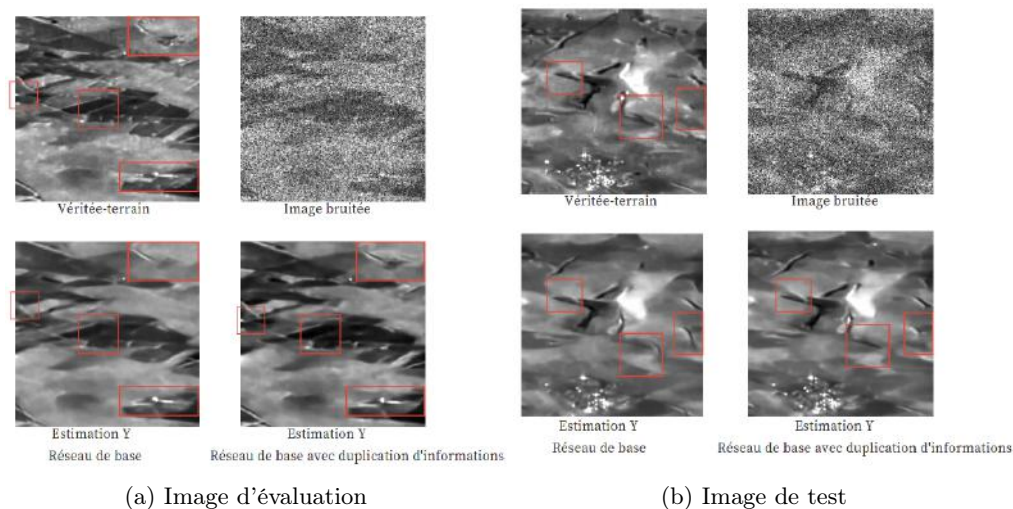


FIGURE 19 – Comparaison avec et sans ajout d'informations dans le réseau de base

L'ajout d'informations permet bien d'améliorer les structures et les détails du débruitage, on obtient également de meilleures valeurs pour les coefficients PSNR et SSIM. Nous allons maintenant ajouter de l'informations en introduisant la multi-temporalité.

6.3 Prise en compte des changements temporels

Pour introduire la multi-temporalité nous allons donner 2 images de date t_1 et t_2 différentes en entrée au réseau c'est-à-dire une série multi-temporelle de 2 images. Cela devrait permettre au réseau d'être plus robuste aux variations temporelles et ainsi d'améliorer le débruitage. Afin de forcer le réseau à prendre en compte les nouvelles informations, nous allons mettre en place une seconde prédiction : une carte de changements entre les 2 dates de la série temporelle.

Nous avons alors besoin de définir la carte de changements, en effet nous avons besoin de la construire en amont de la prédiction du réseau en tant que vérité-terrain. On peut penser à deux possibilités :

- Carte de changement binaire
- Carte de changement en niveau de gris

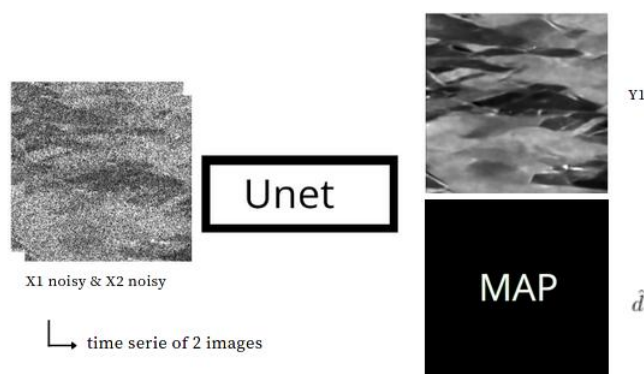


FIGURE 20 – Introduction de la multi-temporalité et prédiction d'une carte de changements

6.3.1 Carte binaire

Une façon de penser une carte de changements binaire est de seuiller la différence absolue entre les 2 dates.

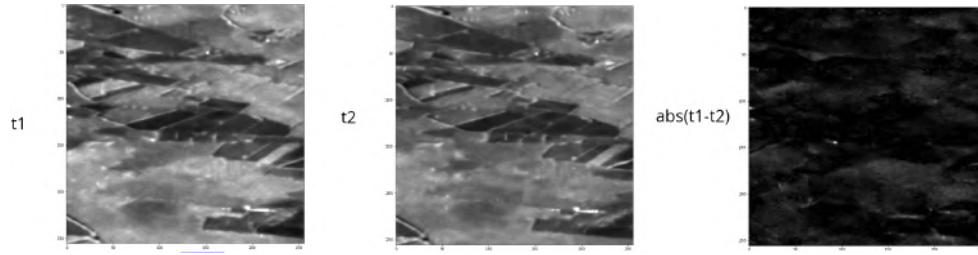


FIGURE 21 – Valeur absolue de la différence entre 2 dates

Après seuillage à différents niveaux :

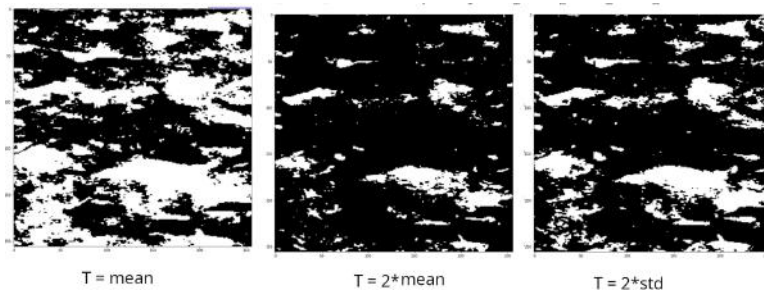


FIGURE 22 – Seuillage de la différence en valeur absolue

On peut remarquer qu'il n'est pas évident de définir un seuil, quelle valeur définie le mieux une carte de changements ? Il est alors plus simple de travailler directement avec une carte de changements en niveau de gris.

6.3.2 Carte en niveau de gris - SC_MAP

Le réseau devra donc prédire l'estimation débruitée y_1 de la première vérité-terrain de la pile x_1 ainsi qu'une carte de changements en niveau de gris \hat{d} entre les 2 dates t_1 et t_2 représentés par les images bruitées x_{1noisy} et x_{2noisy} .

Nous allons nommer par la suite le réseau avec prise en compte de changement temporels au sein d'une pile multi-temporelle SC_MAP. Voici le réseau en question :

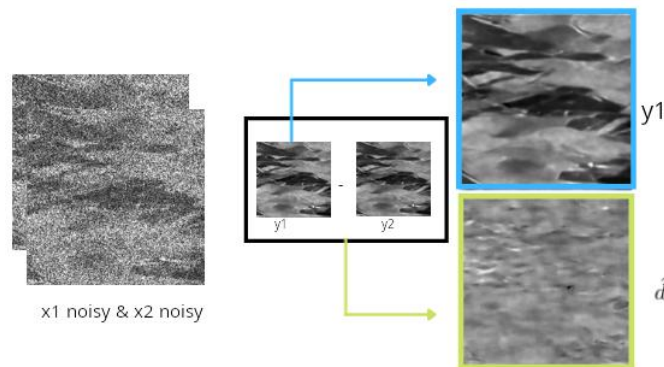


FIGURE 23 – SC_MAP

Nous avons donc en entrée du réseau une série temporelle de 2 images et en sortie nous avons la prédiction du débruitage y_1 et la carte de changements binaire \hat{d} . On peut maintenant définir la fonction de coût pour la prédiction de la carte comme suit :

$$Loss_2 = |\hat{d} - y_1 + x_2|$$

Cela sous-entend que l'on cherche à prédire la carte optimale $d^* = y_1 - x_2$ et donc que le réseau sera implicitement obligé de débruiter toute la série temporelle de sorte que $\hat{d} = y_1 - y_2$ doit le plus proche possible de d^* . Tout en prenant en compte que la meilleure estimation de y_1 par la première fonction de coût est $y_1^* = x_1$

On ajoute donc ce nouveau terme à la fonction de coût précédente de la sorte : $Loss = Loss_1 + Loss_2$
Notre fonction de coût est alors définie comme :

$$Loss = (\frac{1}{batch_size} \sum_{i=0}^{N-1} |X_{1i} - Y_{1i} + cn|)) + (\frac{1}{batch_size} \sum_{i=0}^{N-1} |\hat{d} - Y_{1i} + X_{2i}|)$$

La redondance du terme Y_1 dans les 2 termes de la fonction de coût et le débruitage implicite de la série temporelles complète, devrait forcer le réseau à prendre en compte ces nouvelles informations pour l'aider au débruitage.

Remarque : on peut retrouver le débruitage implicite y_2 directement grâce aux prédictions y_1 et \hat{d} :
 $\hat{d} = y_1 - y_2 \Leftrightarrow y_2 = y_1 - \hat{d}$

6.4 Réseau multi-temporel avec prédiction de carte de changements

6.4.1 SC_MAP x1x1

Afin de voir le comportement du réseau, nous pouvons essayer de lui donner en entrée 2 fois la même date t_1 avec 2 simulations de bruit différentes, on aura donc en entrée une pile composée de $x_{1noisy1}$ et $x_{1noisy2}$:

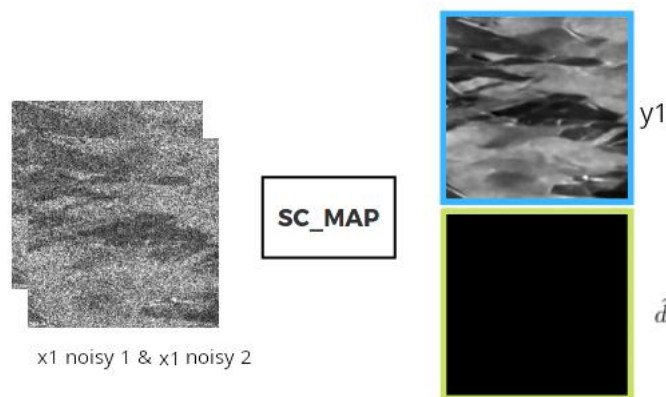


FIGURE 24 – SC_MAP x1x1

Nous devons théoriquement obtenir une carte de changements nulle, vérifions le comportement du réseau face à cette situation :

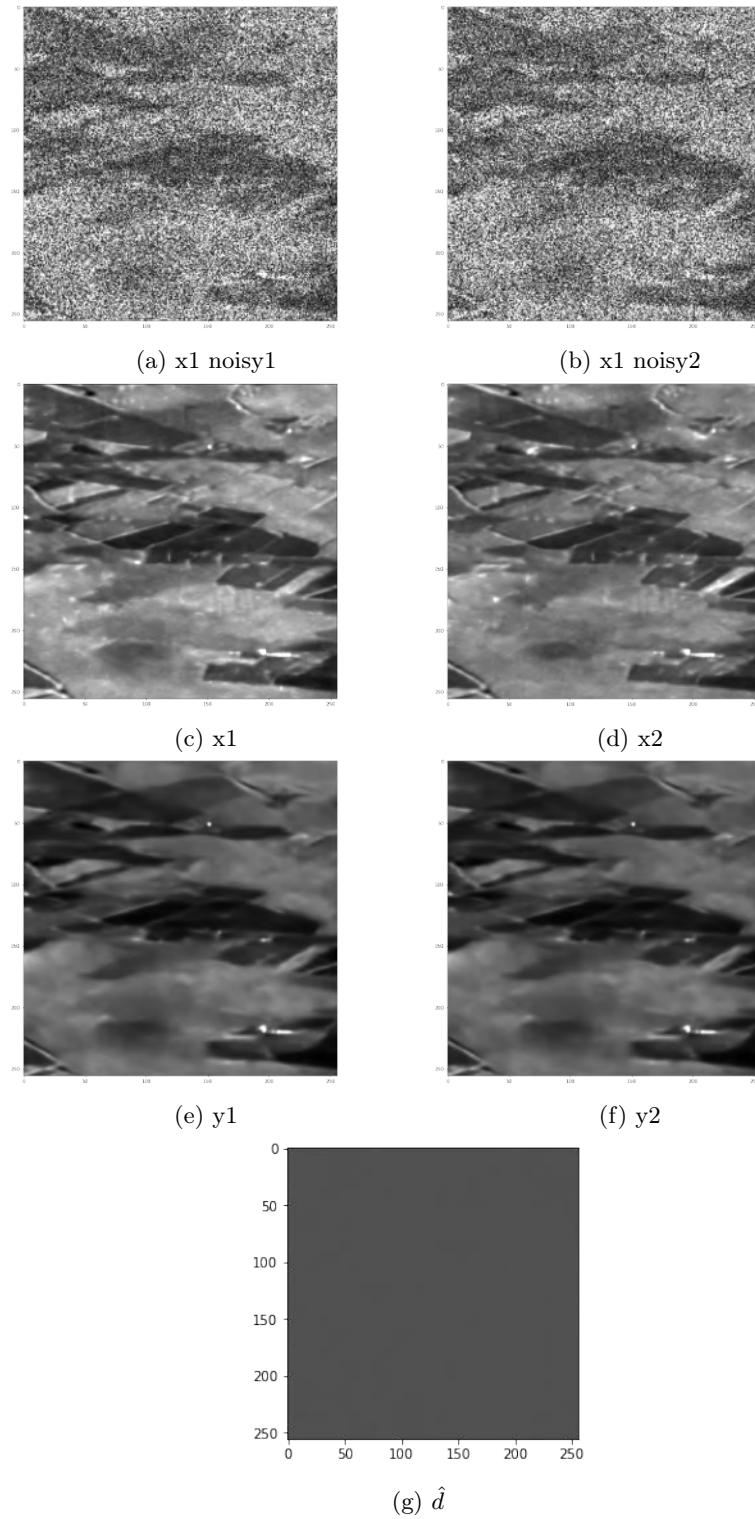


FIGURE 25 – Image d'évaluation

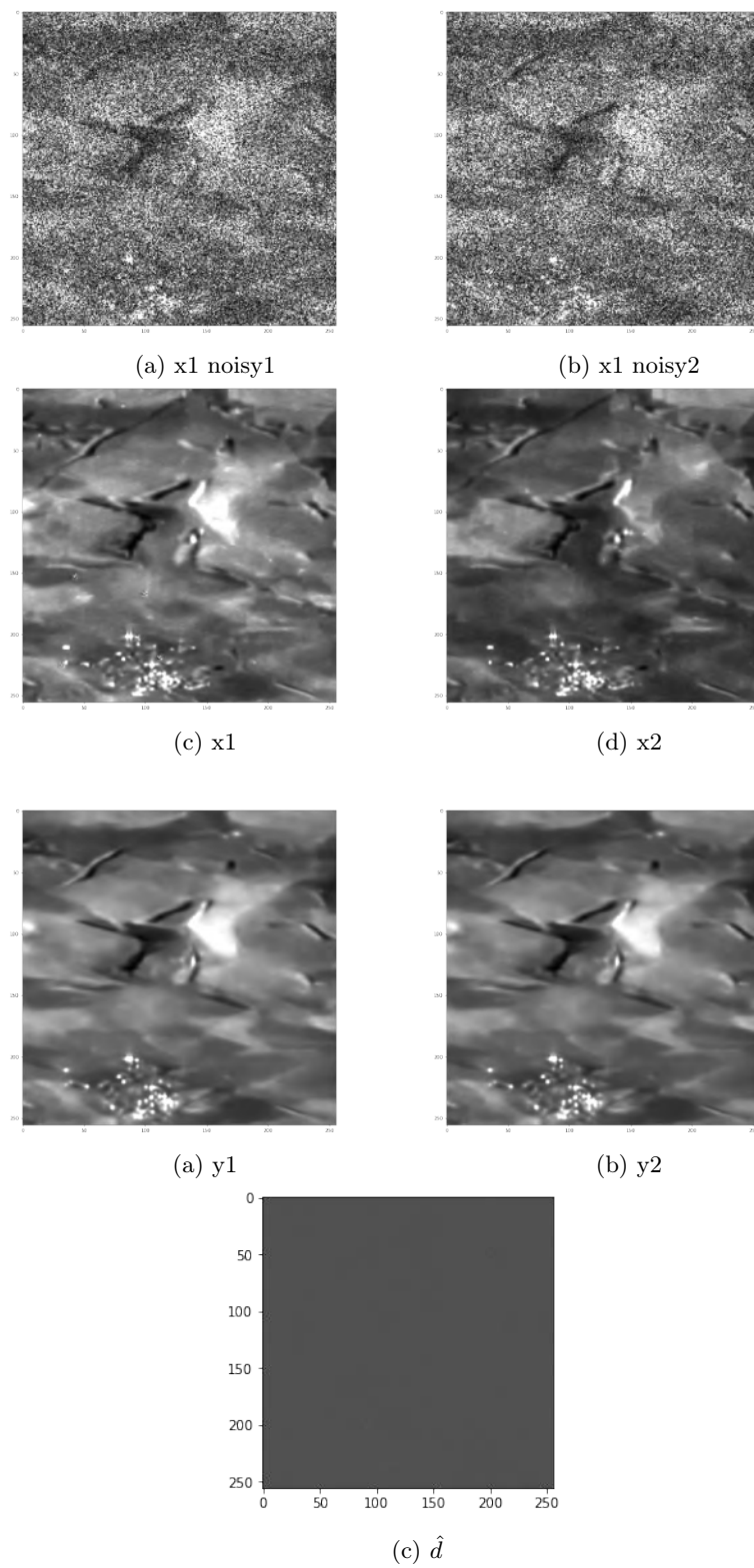


FIGURE 27 – Image de test

Nous pouvons voir dans un premier temps qu'en effet les 2 images de la série temporelle sont bien débruitées pour permettre l'estimation de la carte de changements.

Nous avons une carte de changements totalement uniforme, aucun changement n'a été prédit. Elle n'est pas totalement nulle, mais en niveau de gris, car la prédiction y_1 n'est pas parfaite, il y a donc des changements entre y_1 et x_1 mais qui sont uniformes et non dû aux changements temporels, d'où l'uniformité de la carte. Si on relance l'apprentissage avec la carte de changements défini comme $d^* = x_1 - x_2$ ce qui donnerait la fonction de coût suivante (nous ne considérerons pas ce changement par la suite) :

$$Loss = (\frac{1}{batch_size} \sum_{i=0}^{N-1} |X_{1i} - Y_{1i} + cn|) + (\frac{1}{batch_size} \sum_{i=0}^{N-1} |\hat{d} - X_{1i} + X_{2i}|)$$

Nous obtenons en effet en évaluation ou en test une carte de changements complètement nulle :

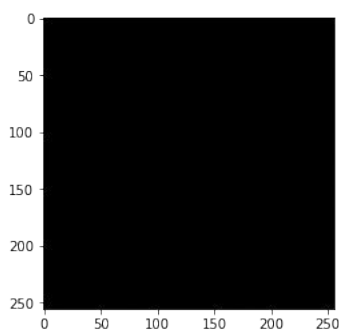


FIGURE 28 – \hat{d}^*

6.4.2 SC_MAP x1x2

Regardons maintenant le comportement du réseau face à une série multi-temporelle, lorsque la pile donnée au réseau est constituée de deux dates t_1 et t_2 différentes :

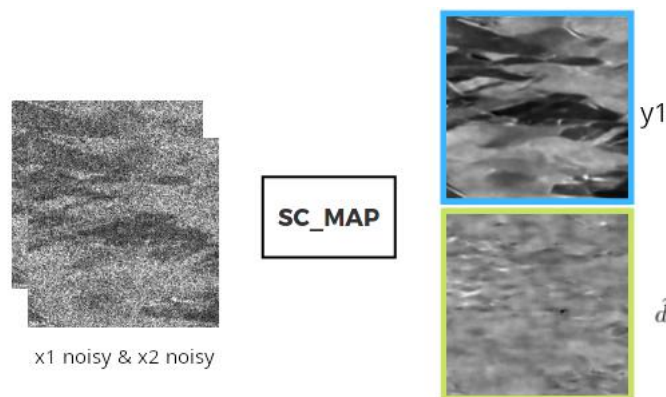


FIGURE 29 – SC_MAP x1x2

Nous obtenons les résultats suivant :

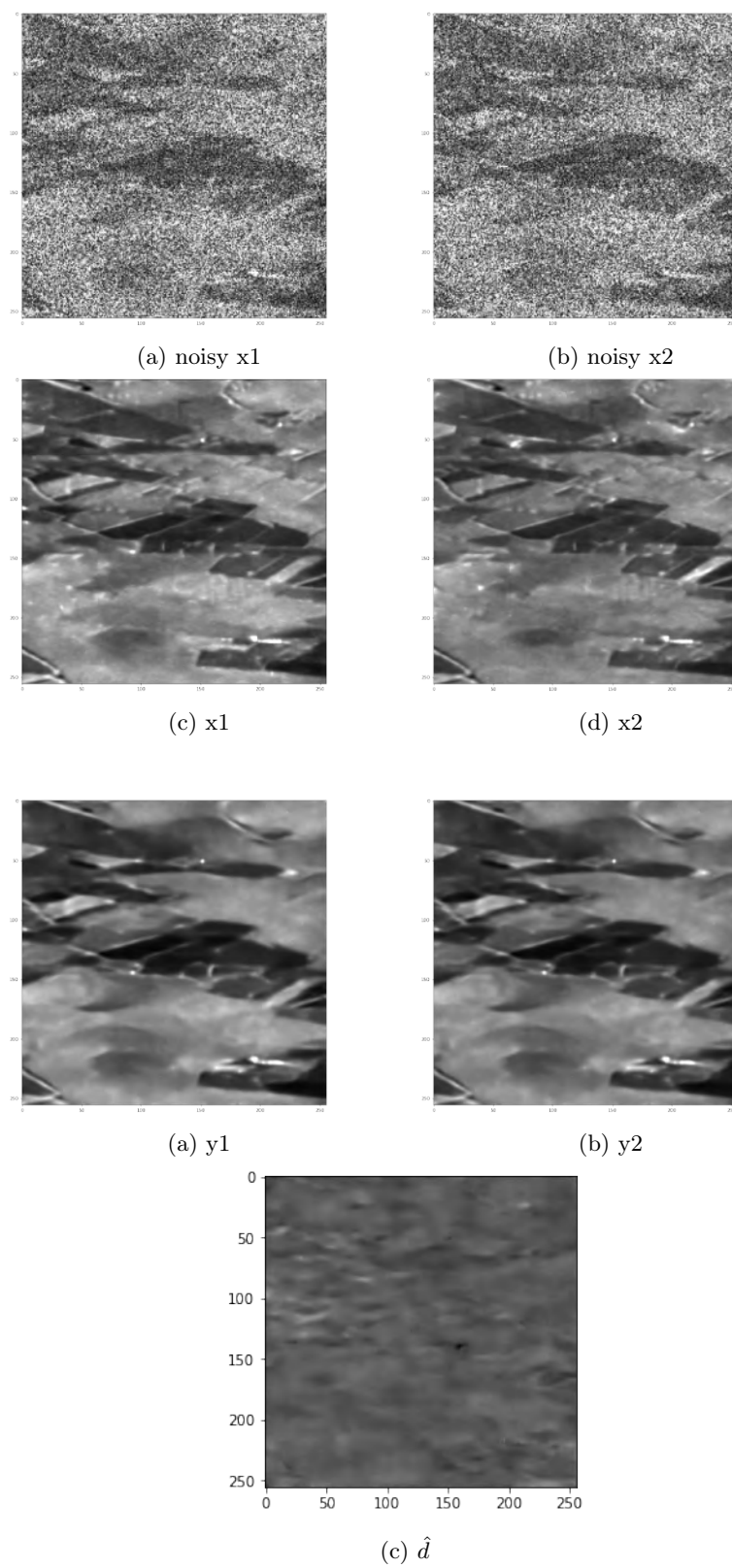


FIGURE 31 – Image d'évaluation

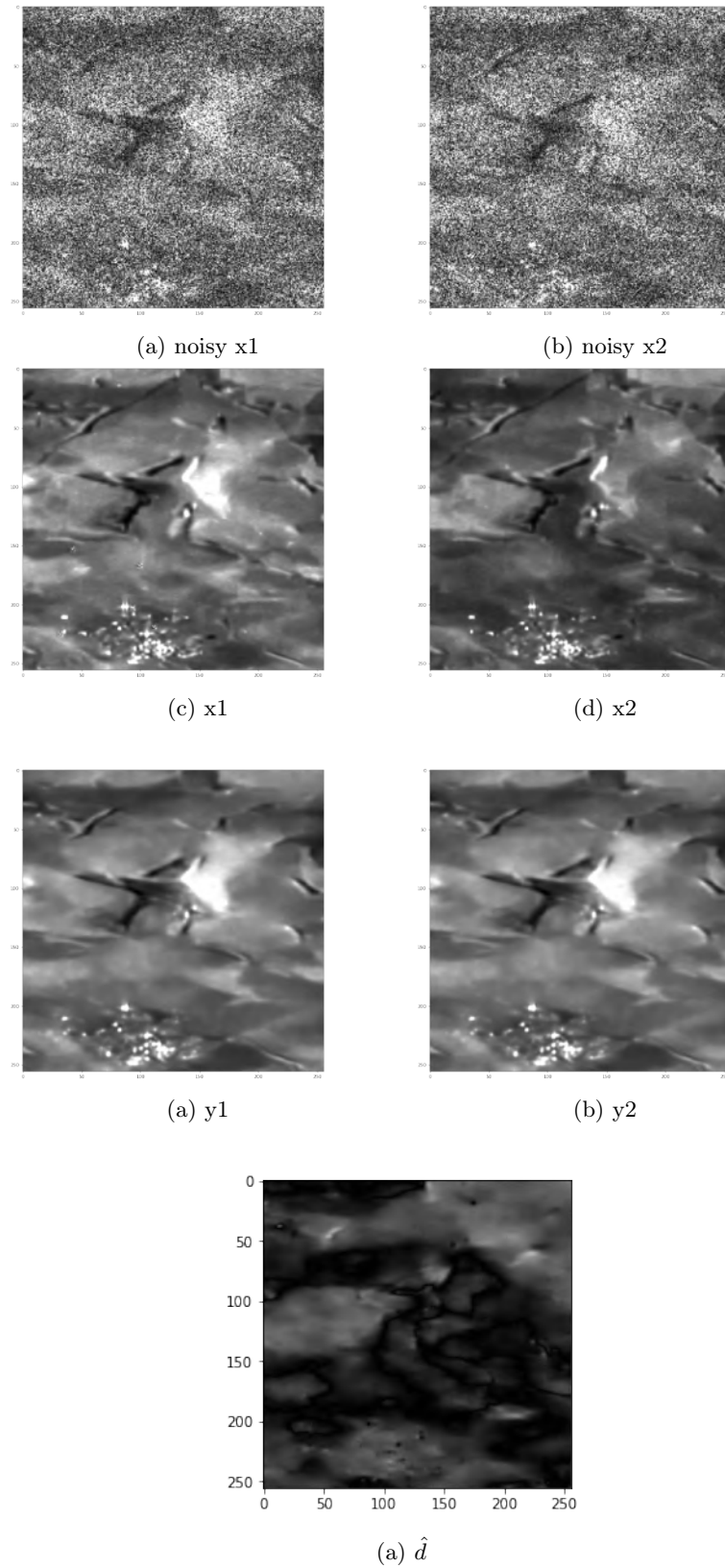


FIGURE 34 – Image de test

On peut voir que le débruitage est intéressant, les structures semblent bien conservées, nous verrons l'intérêt de la carte de changements sur le débruitage en partie 6.6 avec des comparaisons des approches effectuées.

6.4.3 SC_MAP x2x1

Regardons maintenant le comportement du réseau face aux permutations temporelles, nous allons inverser l'ordre des entrées qui ne seront plus les temps t_1 et t_2 mais respectivement t_2 et t_1 , on conservant les notations précédente l'image débruitée correspond alors à y_2 :

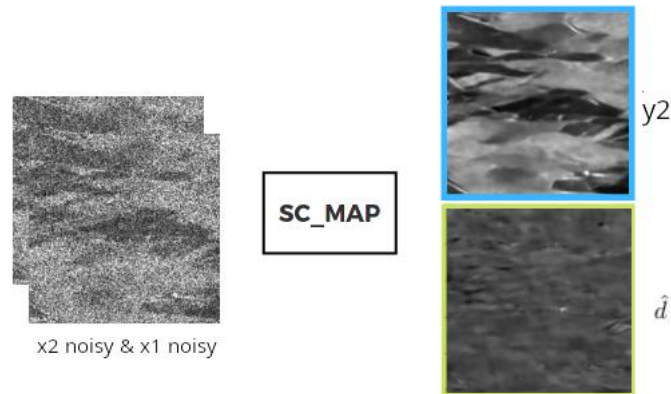
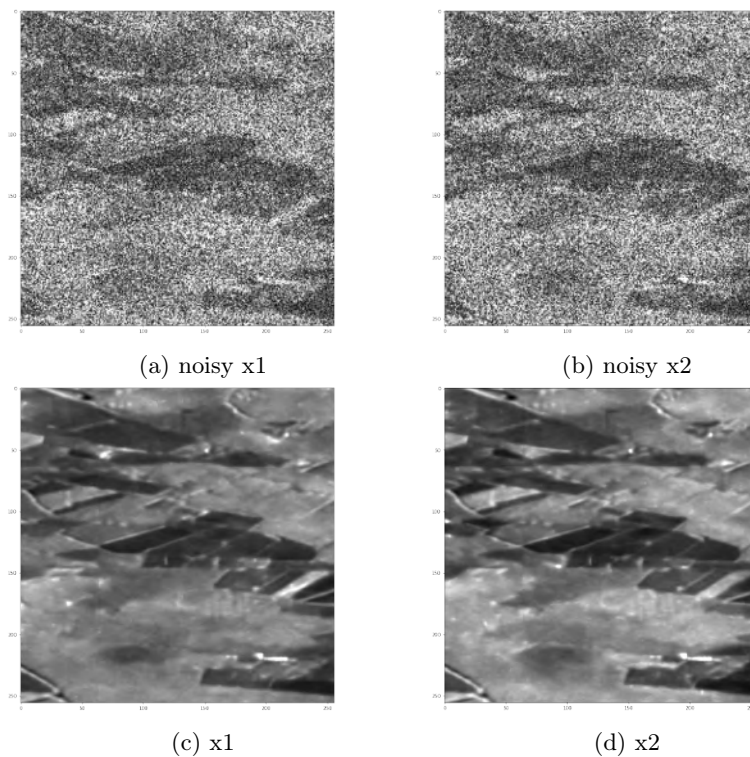


FIGURE 35 – SC_MAP x2x1

Regardons les résultats, avec les nouvelles notations x_1 corresponds à l'image du temps t_1 avec inversion, donc à l'image du temps t_2 sans inversion, de même pour les autres images :



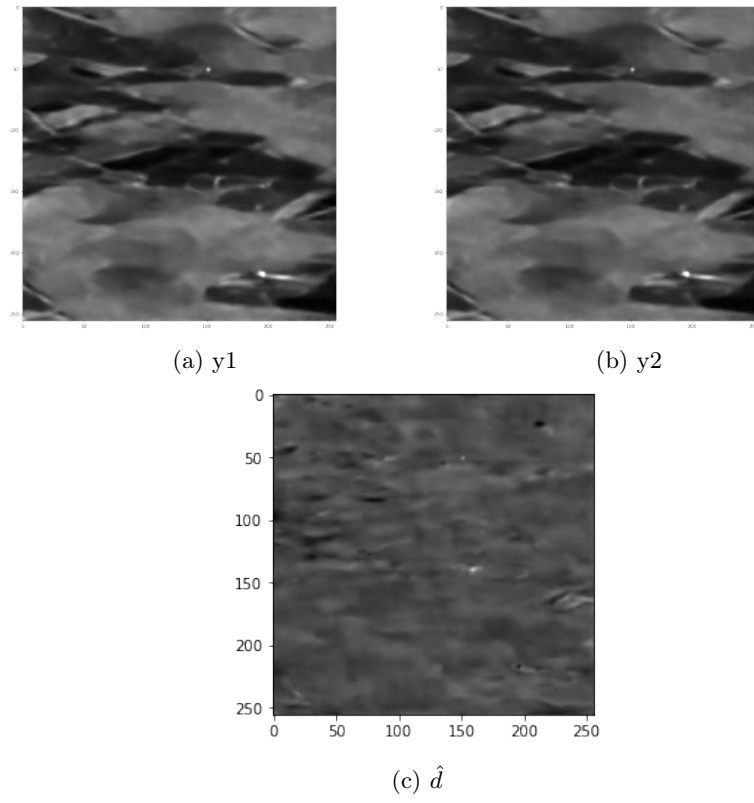
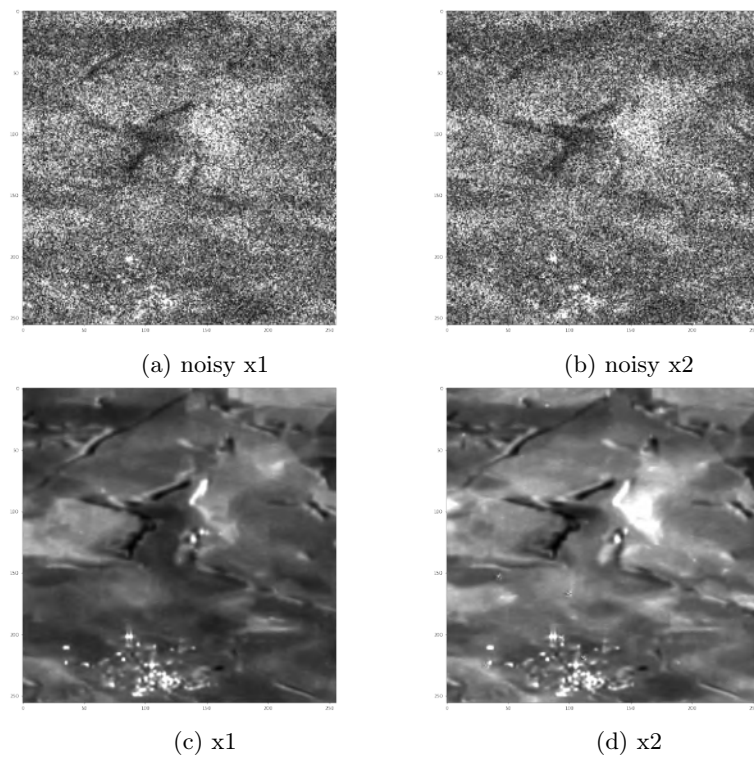


FIGURE 37 – Image d'évaluation



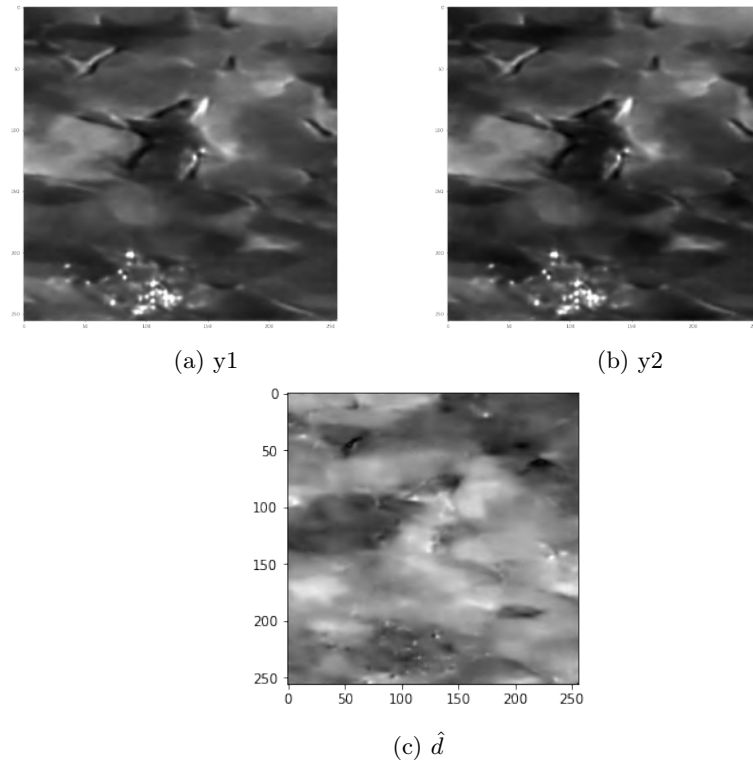
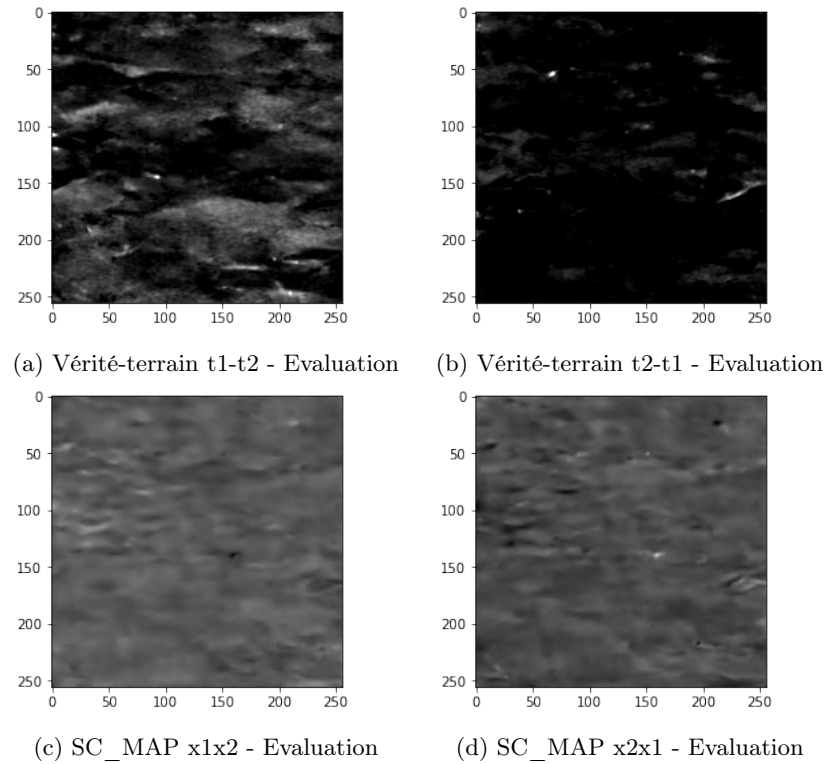


FIGURE 39 – Image de test

On peut voir que la prédiction de la carte de changements n'est pas la même entre SC_MAP $x_1 \times x_2$ et SC_MAP $x_2 \times x_1$, cela signifie que le réseau n'est pas invariant aux permutations temporelles :



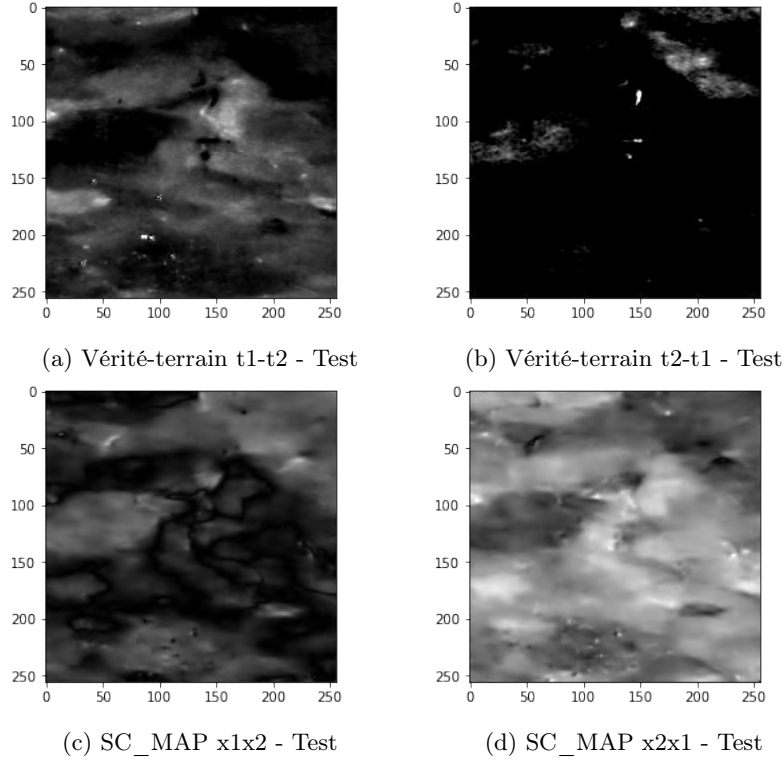


FIGURE 41

Les résultats des cartes de changements me semblent étonnant, la carte de changements x1x2 semble visuellement plus proche de la vérité-terrain t2t1, de même pour la carte prédite x2x1 qui semble plus proche de la vérité-terrain t1t2. J'ai vérifié si je n'ai pas inversé des dates dans mon code, je n'ai trouvé aucun biais ni erreur qui pourrait fournir cette possible inversion. Si cela est une erreur, je n'ai pas réussi à l'identifier.

Dans tous les cas, nous pouvons voir que le réseau n'est effectivement pas invariant aux permutations temporelles, ce qui est normal étant donné que les vérité-terrains ne le sont pas elles-mêmes : $x1-x2 \neq x2-x1$. Plutôt que de chercher à prédire une carte pour entre les séries x1x2 et x2x1, on va plutôt chercher à prédire une carte de changements unique entre ces 2 piles et donc invariante aux permutations temporelles.

6.5 Invariance du réseau aux permutations temporelles

Pour que le réseau soit invariant aux permutations temporelles, on veut minimiser la différence entre les prédictions de la carte de changements \hat{d}_1 du réseau SC_MAP pour les entrées t_1 et t_2 et de la carte \hat{d}_2 pour les entrées t_2 et t_1 :

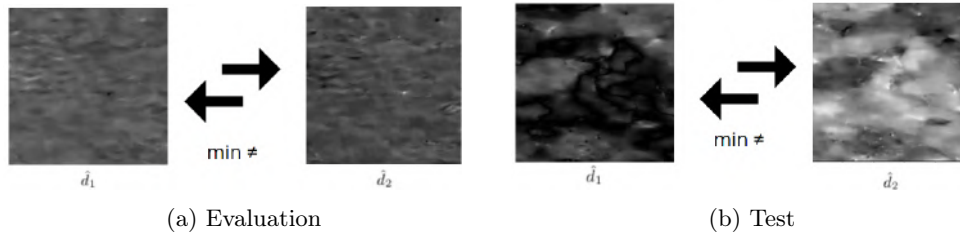


FIGURE 42

Pour cela, nous allons utiliser un troisième terme dans la fonction de coût minimisant la différence absolue entre ces 2 prédictions $|\hat{d}_1 - \hat{d}_2|$, la puissance de ce terme sera contrôlée par l'hyperparamètre γ .

$$Loss_3 = \gamma \left(\frac{1}{batch_size} \sum_{i=0}^{N-1} |\hat{d}_{1i} - \hat{d}_{2i}| \right)$$

La fonction de coût finale sera de la sorte : $Loss = Loss_1 + Loss_2 + \gamma Loss_3$ tel que :

$$Loss = \left(\frac{1}{batch_size} \sum_{i=0}^{N-1} |X_{1i} - Y_{1i} + cn| \right) + \left(\frac{1}{batch_size} \sum_{i=0}^{N-1} |\hat{d}_1 - Y_{1i} + X_{2i}| \right) + \gamma \left(\frac{1}{batch_size} \sum_{i=0}^{N-1} |\hat{d}_{1i} - \hat{d}_{2i}| \right)$$

Pour cela, nous allons utiliser le fonctionnement des réseaux siamois, c'est-à-dire que l'on fera 2 apprentissages avec des poids partagés :

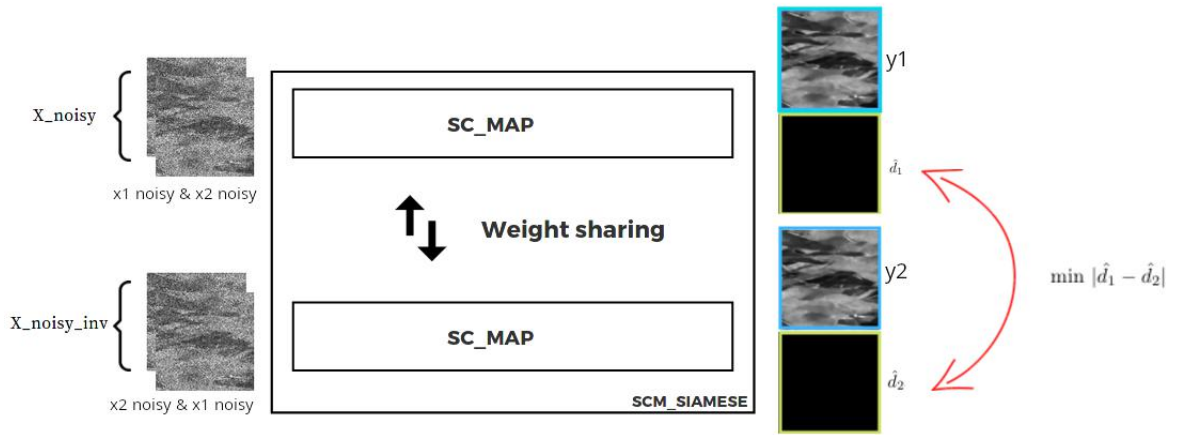


FIGURE 43 – SCM_SIAMESE

```
y1, d_estim1 = autoencoder(X_noisy)
y2, d_estim2 = autoencoder(X_noisy_inv)
Loss3 = np.abs(d_estim1 - d_estim2)
```

Nous appellerons ce réseau par la suite SCM_SIAMESE.

6.5.1 Optimisation de l'hyperparamètre γ

Nous allons maintenant nous intéresser à la valeur de l'hyperparamètre γ :

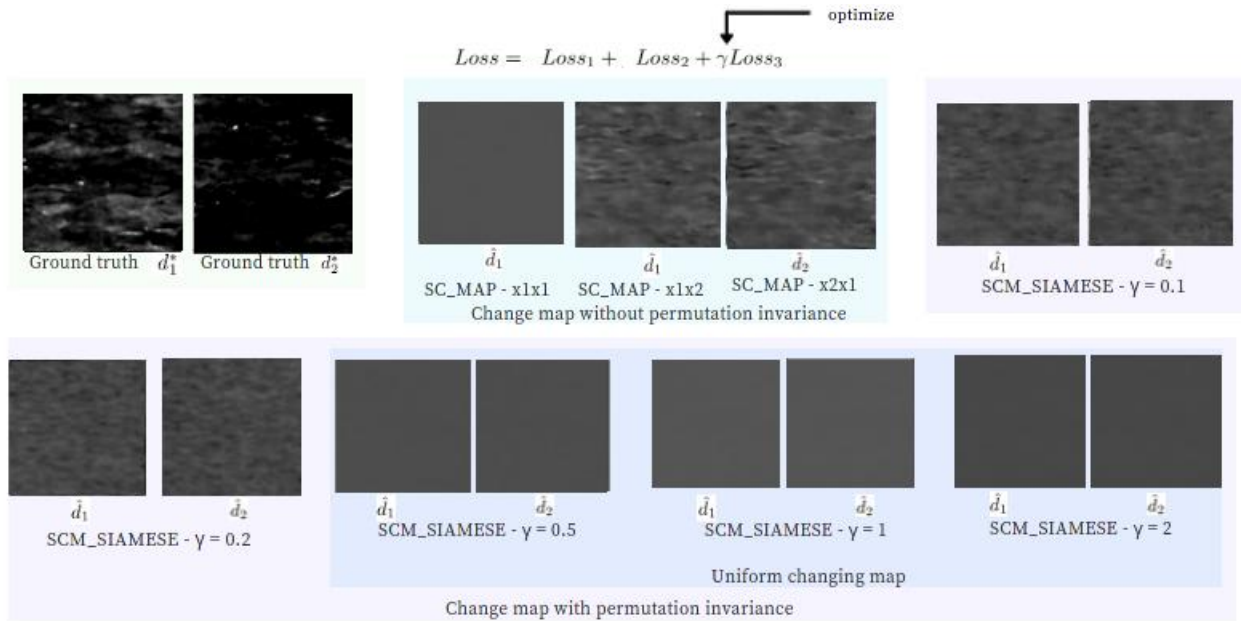


FIGURE 44 – Evaluation

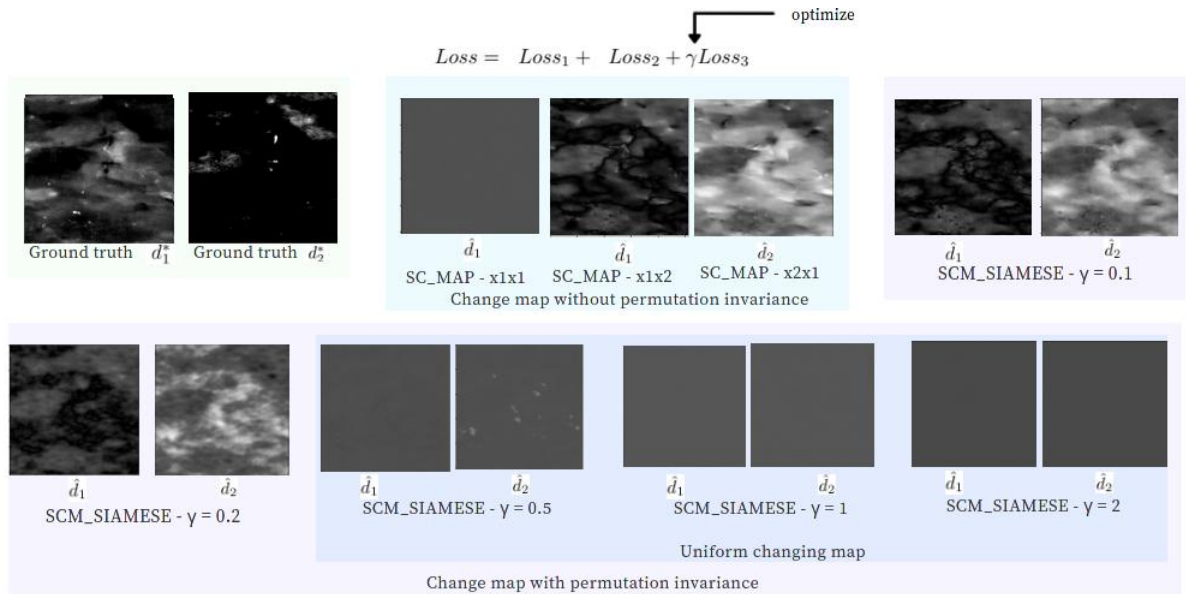


FIGURE 45 – Test

- Prédiction des cartes de changements sans invariances aux permutations - *SC_MAP*
On peut voir que lorsque l'on donne 2 fois le même temps t_1 nous avons bien comme expliqué précédemment une carte de changements uniforme. On peut voir aussi que le réseau n'est pas invariant aux permutations temporelles entre la pile $x1x2$ et $x2x1$ la carte prédite est différente.
- Prédiction des cartes de changements avec invariances aux permutations - *SCM_SIAMESE*
Maintenant, lorsque l'on regarde le comportement du réseau avec l'invariance face aux permutations

temporelles, on peut voir que selon le poids qui est donné au troisième terme de la fonction de coût par γ , que plus celui-ci est fort, plus les cartes tendent à être similaires. Cependant, elles tendent à être similaires vers un unique niveau de gris, ce qui ne nous intéresse pas puisqu'aucun changement n'est prédit alors qu'il y en a en réalité. Le poids est alors trop fort et le réseau préfère converger vers une uniformité pour favoriser la minimisation de la $Loss_3$ au dépit de la $Loss_2$ qui cherche une carte de changements réaliste et proche de la vérité-terrain.

Le réseau semble avoir du mal à converger vers une invariance de carte de changements face aux permutations. Nous devons alors nous concentrer sur des faibles valeurs de γ pour éviter cette uniformité. Pour cela, regardons visuellement les débruitages associés :

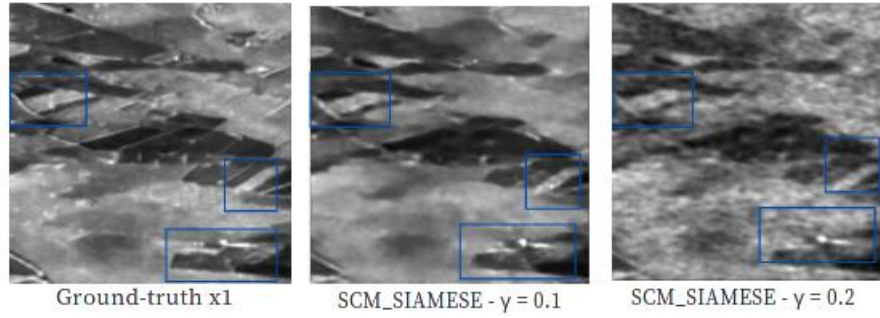


FIGURE 46 – Evaluation

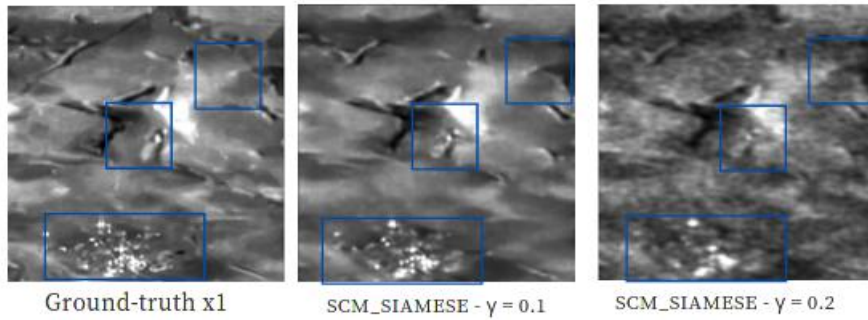


FIGURE 47 – Test

On peut voir que $\gamma = 0.2$ produit un très mauvais débruitage, cela provient probablement du fait que le réseau essaye de produire des cartes de changements égales, on semble être à la limite où le réseau préfère tendre vers une uniformité des cartes, la limite où le réseau détruit les informations de la carte et des estimations, pour prédire une carte qui n'est pas réaliste et qui n'a pas encore convergé vers l'uniformité. Cela dégrade fortement les performances du réseau.

Lorsque $\gamma = 0.1$ on voit cependant un bon débruitage avec une conservation de structures et de détails intéressants. Même si les cartes ne sont pas aussi similaires que ce que l'on aurait pensé, elles le sont déjà un peu plus que sans la prise en compte de la $Loss_3$. Nous allons maintenant comparer l'ensemble des méthodes entre elles et voir si l'invariance aux changements temporels permet au réseau d'améliorer ses performances.

6.6 Comparaison des approches implémentées

6.6.1 Evaluation quantitative

Voici une évaluation quantitative basé sur 2 indices :

- PSNR - Peak Signal to Noise Ratio

C'est un terme désignant le rapport entre la puissance maximale possible d'un signal et la puissance du bruit de corruption qui affecte la fidélité de sa représentation.

$$\text{PSNR}(x, y) = 20 \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right)$$

- SSIM - Structural Similarity Index for Measuring image quality

C'est un modèle basé sur la perception qui considère la dégradation de l'image comme une modification perçue de l'information structurelle, tout en incorporant d'importants phénomènes perceptifs, notamment les termes de masquage de luminance et de masquage de contraste. L'information structurelle est l'idée que les pixels ont de fortes interdépendances, surtout lorsqu'ils sont proches dans l'espace. Ces dépendances sont porteuses d'informations importantes sur la structure des objets de la scène visuelle.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Où μ_x , μ_y , σ_x , σ_y et σ_{xy} sont les moyennes locales, les écarts types et la covariance croisée pour les images x et y .

$$C1 = (0.01 * L)^2$$

$$C2 = (0.03 * L)^2$$

Où L est la range dynamique.

Ensemble d'évaluation

evaluation	Réseau de base	Réseau de base x1x1	SC_MAP x1x1	SC_MAP x1x2	SC_MAP x2x1
Mairais1_10 PSNR	31.549	33.067	32.963	32.792	28.587
Mairais1_10 SSIM	0.862	0.888	0.888	0.884	0.878

evaluation	SCM_SIAMOIS - $\gamma = 0$	SCM_SIAMOIS - $\gamma = 0.1$	SCM_SIAMOIS - $\gamma = 0.2$
Mairais1_10 PSNR	32.417	32.625	30.348
Mairais1_10 SSIM	0.877	0.873	0.824

evaluation	SCM_SIAMOIS - $\gamma = 0.5$	SCM_SIAMOIS - $\gamma = 1$	SCM_SIAMOIS - $\gamma = 2$
Mairais1_10 PSNR	29.508	32.019	32.830
Mairais1_10 SSIM	0.857	0.861	0.880

Ensemble de Test

test	Réseau de base	Réseau de base x1x1	SC_MAP x1x1	SC_MAP x1x2	SC_MAP x2x1
Limagne1_10 PSNR	28.577	30.481	30.391	29.576	31.262
Limagne1_10 SSIM	0.840	0.876	0.875	0.859	0.890

test	SCM_SIAMOIS - $\gamma = 0$	SCM_SIAMOIS - $\gamma = 0.1$	SCM_SIAMOIS - $\gamma = 0.2$
Limagne1_10 PSNR	29.426	28.894	25.559
Limagne1_10 SSIM	0.854	0.846	0.790

test	SCM_SIAMOIS - $\gamma = 0.5$	SCM_SIAMOIS - $\gamma = 1$	SCM_SIAMOIS - $\gamma = 2$
Limagne1_10 PSNR	23.996	27.349	25.734
Limagne1_10 SSIM	0.802	0.812	0.847

6.6.2 Evaluation visuelle

L'évaluation quantitative n'est pas suffisante à elle-même, il est intéressant de comparer visuellement les résultats :

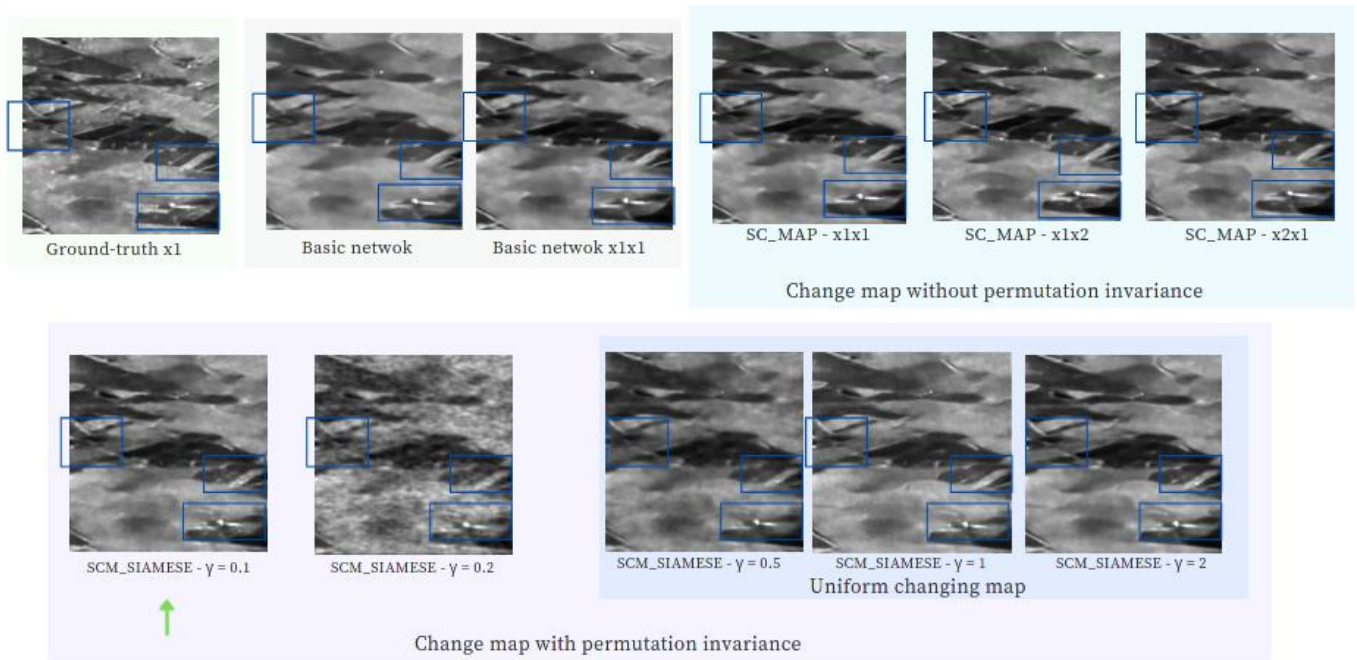


FIGURE 48 – Evaluation

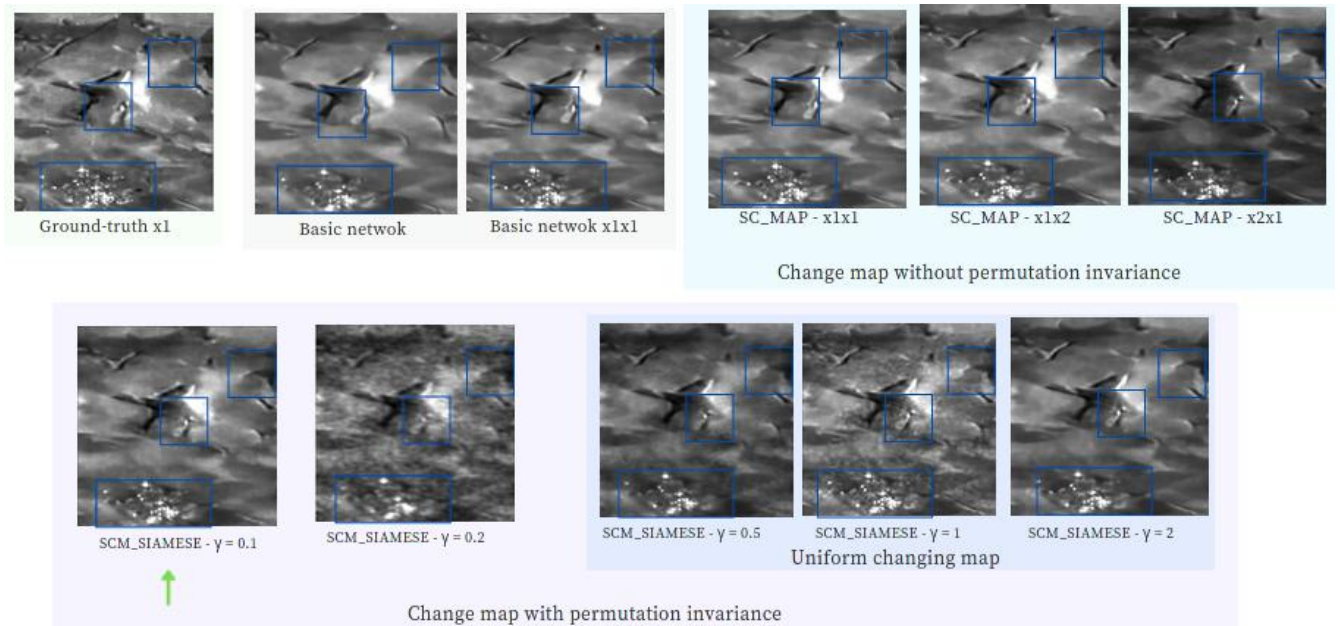


FIGURE 49 – Test

Finalement, on peut voir que le meilleur résultat semble être le réseau *SCM_SIAMESE* avec $\gamma = 0.1$ c'est-à-dire le réseau qui prends en compte les changements temporels avec une invariances face aux permutations. Bien que celui-ci n'est pas les meilleurs indices PSNR et SSIM, on peut surtout remarquer que le débruitage contient de nombreuses structures qui ne sont pas visibles sur les autres méthodes.

Regardons le ratio entre la vérité-terrain x_1 au temps t_1 et l'estimation y_1 pour le réseau de base et $SCM_SIAMESE$ avec $\gamma = 0.1$:

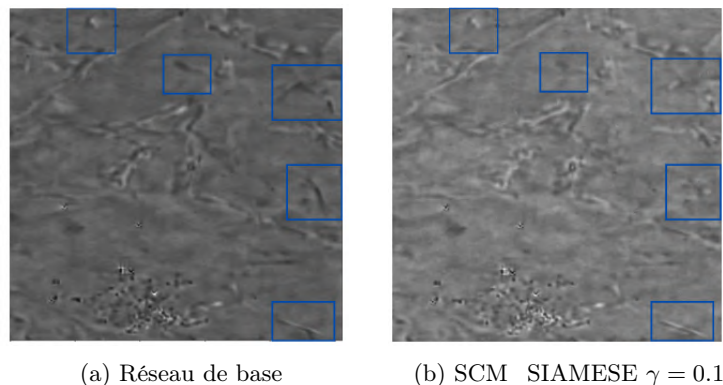


FIGURE 50 – Images de test $\frac{y_1}{x_1}$

On peut en effet voir moins de structures résiduelles dans le ratio.

L'introduction d'une carte de changements et l'invariance aux changements temporels semblent selon nos résultats augmenter les performances du réseau, comme dit précédemment la $Loss_3$ doit avoir un poids plus faible comparé aux 2 autres $Loss$ environ 10 fois moins d'importance, $\gamma = 0.1$ semble donc être une bonne valeur dans ces conditions d'expérimentations.

Remarque : La qualité d'image n'étant pas optimale sur un pdf après compression, je peux vous joindre les images des résultats.

7 Entraînement

- Les expériences ont été réalisées avec un GPU NVIDIA GeForce GTX Titan, GPU de la salle ppti de l'université (ppti-gpu-1).
- L'apprentissage se déroule sur 30 epochs avec un learning rate de 0,001. 6 (Lely), 2 (Marais 1) dates de taille 256 composent l'ensemble d'apprentissage. 1 lot de 8 patches est utilisé par image.
- L'évaluation se déroule sur 3 dates de la pile Marais1 (différentes de celles d'apprentissage), ["marais1_12", "marais1_11", "marais1_10"]
- Les tests se déroulent sur 2 dates de la pile limagne1 ["limagne1_11", "limagne1_10"].
- Pour l'évaluation et les tests avec duplication de l'informations seule la date "10" avec 2 simulations de bruit différents est utilisée.

Remarque :

Par soucis de temps d'apprentissage, j'ai effectué l'apprentissage sur très peu d'image par rapport aux articles de la bibliographie, il serait intéressant d'étendre les apprentissages à un ensemble d'apprentissage plus conséquent. Cependant, les résultats semblent déjà intéressants et comparables sur ce peu de donnée.

8 Conclusion

Finalement, on a pu voir que la duplication d'informations dans le réseau de base permet d'améliorer les performances de débruitage, les structures sont notamment mieux préservées. Par la suite, nous avons introduit la multi-temporalité à travers une série multi-temporelle de 2 images, avec la prédiction d'une carte de changements pour forcer le réseau à utiliser ces nouvelles informations. Nous avons pu voir que cela améliore les performances de débruitage par rapport au réseau de base, cependant, il n'est pas invariant aux permutations. Nous avons alors introduit l'invariance de la prédiction de la carte face aux permutations temporelles et nous obtenons de meilleures performance pour de petite valeur de γ .

Il faut faire attention à l'interprétabilité des résultats qui ne sont appris que sur un petit dataset, il serait intéressant de lancer les apprentissages sur un dataset plus important, ce que je n'ai pas eu le temps de faire dû à la longueur des apprentissages sur de grands datasets.

Une extension pourrait être de tester avec une série temporelle de 3 images pour voir si le réseau est capable de prendre en compte les nouvelles informations, afin de voir si il est intéressant d'augmenter la taille de la série temporelle.

9 Bibliographie

- [1] E. Dalsasso, X. Yang, L. Denis, F. Tupin, and W. Yang, “SAR Image Despeckling by Deep Neural Networks : from a Pre-Trained Model to an End-to-End Training Strategy,” *Remote Sensing*, vol. 12, no. 16, p. 2636, Jan. 2020, doi : 10.3390/rs12162636.
- [2] E. Dalsasso, L. Denis, and F. Tupin, “SAR2SAR : a semi-supervised despeckling algorithm for SAR images,” *IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing*, vol. 14, pp. 4321–4329, 2021, doi : 10.1109/JSTARS.2021.3071864.
- [3] W. Zhao, C.-A. Deledalle, L. Denis, H. Maître, J.-M. Nicolas, and F. Tupin, “Ratio-Based Multi-temporal SAR Images Denoising : RABASAR,” *IEEE Transactions on Geoscience and Remote Sensing*, 2019, doi : 10.1109/TGRS.2018.2885683.
- [4] E. Dalsasso, I. Meraoumia, L. Denis, and F. Tupin, “Exploiting multi-temporal information for improved speckle reduction of Sentinel-1 SAR images by deep learning,” *arXiv :2102.00682 [cs, eess]*, Feb. 2021, Accessed : Oct. 25, 2021. [Online]. Available : <http://arxiv.org/abs/2102.00682>
- [5] C.-A. Deledalle, L. Denis, S. Tabti, and F. Tupin, “MuLoG, or How to Apply Gaussian Denoisers to Multi-Channel SAR Speckle Reduction ?,” *IEEE Trans. on Image Process.*, vol. 26, no. 9, pp. 4389–4403, Sep. 2017, doi : 10.1109/TIP.2017.2713946.
- [6] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, “A tutorial on synthetic aperture radar,” *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 1, pp. 6–43, Mar. 2013, doi : 10.1109/MGRS.2013.2248301.
- [7] J. Lehtinen et al., “Noise2Noise : Learning Image Restoration without Clean Data,” *arXiv :1803.04189 [cs, stat]*, Oct. 2018, Accessed : Feb. 07, 2022. [Online]. Available : <http://arxiv.org/abs/1803.04189>