

Détection de changements entre deux images satellitaires

Filipe Lauer, Júlia Togashi

Novembre 2020

1 Introduction

Synthetic Aperture Radar (SAR) c'est une technique qui permet de capturer des images de jour comme de nuit dans différentes conditions météorologiques avec un peu de bruit responsable pour devenir difficile sa interprétation par des humains [3].

Détecter le changement entre deux images SAR dans le même terrain mais en moments différents, est très important pour vérifier quelques nuances qui sont produit pendent le temps. Cette détection peut-être applicable dans différents domaines et applications, comme détecter le damage fait par un tremblement de terre [1], le changement de glace de mer pendent des années [4] ou l'évolution des zones urbaines [2].

Dans ce projet, nous vérifions un algorithme avec différents lois de distributions à priori pour trouver un bon seuil qui permet détecter le changement entre deux images. D'abord, nous implémentons la méthode du seuil d'erreur minimum avec la distribution normale proposé par [5] en python et nous le testons dans des images synthétiques pour vérifier s'il marche bien. Après, nous implémentons aussi l'algorithme de seuil d'erreur minimum avec les 3 trois lois proposés par [6] et nous testons tous les méthodes implémenté pour détecter le changement d'un rivière dans une image SAR originale et débruité. Enfin, nous comparons le seuil trouvé par les différents méthodes dans les deux cas, avec les images originales et débruités.

Pour évaluer la performance entre les méthodes et les images originales et débruités, nous faisons deux types d'analyse. La première c'est une analyse qualitatif, où nous regardons l'image binarisé pour le seuil et aussi l'image binarisé multiplié par un masque binaire du vérité terrain. La deuxième c'est une analyse quantitatif, où nous utilisons un masque binaire du vérité terrain de changement pour évaluer l'image binarisé trouvé après l'application du seuil. Après ces deux analyses, nous avons trouvé que les résultats ne sont pas beaucoup mieux par rapport aux méthodes, mais oui par rapport à débruitage des images.

Le reste de cet article est organisé comme suit. Dans la section 2, nous présentons la première partie du projet, où nous implémentons l'algorithme pour les quatre lois et testons lequel avec la loi normal dans les données synthétiques et aussi nous montrons l'intuition de l'algorithme pour débruiter les images. Dans la section 3, nous montrons les données réels originales et débruités et les expériences que nous avons fait avec ces données. Dans la section 4 nous présentons et discutons les résultats obtenus. Enfin, dans la section 5 nous parlerons sur l'apprentissage que nous avons eu avec le projet.

2 Description des méthodes

Dans cette section nous parlerons sur les méthodes utilisés dans le projet. Nous commençons par la méthode de seuil d'erreur minimum proposé par [5], après nous parlerons sur quelques généralisations de cette méthode, proposé par [6]. Au fin, nous faisons une brève discussion sur l'algorithme pour débruiter des images SAR proposé par [3].

2.1 Le seuil d'erreur minimum

Kittler and Illingworth (1985) [5] présentent une méthode pour trouver le seuil d'erreur minimum pour classer deux images de façon automatique. La méthode utilise l'histogramme en niveaux de gris de l'image et suppose une distribution gaussienne.

Donc nous pouvons considérer l'histogramme comme un estimateur de la densité de probabilité de deux distributions à niveau de gris distinctes. Les distributions $p(g|i)$ ont la moyenne μ_i , écart type σ_i et probabilité à priori P_i .

Nous pouvons dire que pour une $p(g|i)$ et P_i donné, il y a τ qui satisfait la relation ci-dessous. Pour trouver le seuil d'erreur minimum, nous devons trouver le bon τ .

$$P_1 p(g|1) \leq P_2 p(g|2) \begin{cases} g \leq \tau \\ g > \tau \end{cases} \quad (1)$$

Ayant l'histogramme gris, nous pouvons trouver τ , en lui augmentent de manière interactive T, jusqu'à ce que nous trouvions une bonne valeur qui minimise une fonction de coût. Cette méthode proposée par [5] approxime les paramètres de la loi normale d'une manière peu coûteuse en calcul numérique. Nous décrivons les formules impliquées ci-dessous.

$$P_i(T) = \sum_{g=a}^b h(g) \quad (2)$$

$$\mu_i(T) = \frac{\sum_{g=a}^b h(g)g}{P_i(T)} \quad (3)$$

$$\sigma_i^2(T) = \frac{\sum_{g=a}^b \{g - \mu_i(T)\}^2 h(g)}{P_i(T)} \quad (4)$$

Où:

$$a = \begin{cases} 0 & i = 1 \\ T + 1 & i = 2 \end{cases} \quad (5)$$

Et:

$$b = \begin{cases} T & i = 1 \\ n & i = 2 \end{cases} \quad (6)$$

Alors, nous cherchons T qui minimise la fonction de coût $J(\tau) = \min_T J(T)$. A partir du développement mathématique que nous pouvons trouver dans [5], nous arrivons finalement à la formule :

$$J(T) = 1 + 2[P_1(T) \log \sigma_1(T) + P_2(T) \log \sigma_2(T)] - 2[P_1(T) \log P_1(T) + P_2(T) \log P_2(T)] \quad (7)$$

D'abord, nous avons implémenté la méthode dans la langage python. Nous cherchons à reproduire les résultats des seuils illustrés dans [5] pour confirmer le bon fonctionnement de notre algorithme.

Pour cela, nous générons des données synthétiques distribuées selon la loi normale grâce à une fonction de la bibliothèque numpy.

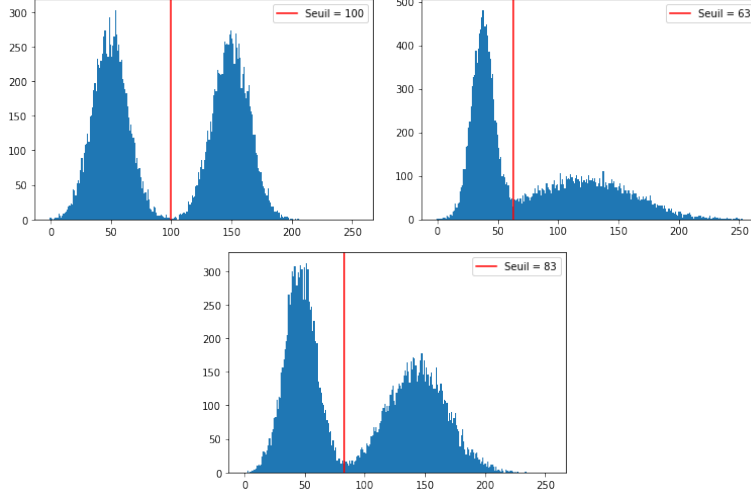


Figure 1: Reproduction des exemples de [5]. L'image en haut gauche sont les lois normales $\mu_1, \sigma_1 = (50, 15)$; $\mu_2, \sigma_2 = (150, 15)$. L'image en haut droite $\mu_1, \sigma_1 = (38, 9)$; $\mu_2, \sigma_2 = (121, 44)$. Enfin l'image en bas $\mu_1, \sigma_1 = (47, 13)$; $\mu_2, \sigma_2 = (144, 25)$. On peut observer la valeur des seuils trouvés pour chaque cas dans les images.

Comme nous pouvons observer, nous avons obtenu des résultats expérimentaux satisfaisants pour le seuil des distributions proposées. Nous remarquons aussi que dans les exemples deux et trois, la méthode trouve le seuil qui minimise l'erreur de classification entre une distribution et l'autre.

2.2 Généralisation du seuil d'erreur minimum

Moser et Serpico (2006) [6] ont généralisé le algorithme de seuil d'erreur minimum pour trouver le changement entre deux images SAR (qui ne suivent pas les distributions gaussiennes) de manière non supervisé.

I_0 et I_1 étant deux images SAR de même taille acquises à partir de la même zone géographique aux instants $t_1 > t_0$. Nous avons que $I_j = \{r_{j1}, r_{j2}, \dots, r_{jn}\}$, où N est égale au nombre de pixels et j des variables aléatoires ($j = 0, 1$). Cette méthode pour trouver le changement utilise le rapport entre les deux images $R = \{u_1, u_2, \dots, u_n\}$ où $u_k = r_{0k}/r_{1k}$.

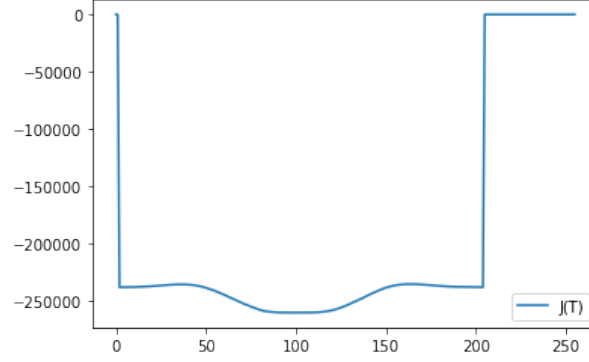


Figure 2: Exemple de la fonction $J(T)$ trouvée pour le premier exemple.

Premièrement, ils proposent une nouvelle formulation pour la fonction de coût $J(T)$, qui prend en compte la densité de probabilité. Cela permet à la méthode d'être généralisée pour les distributions outre la loi normale.

$$J(\tau) = - \sum_{i=0}^1 \left[P_{i\tau} \log(P_{i\tau}) + \sum_{u \in R_{i\tau}} h(u) \log p_i(u|\hat{\theta}_{i\tau}) \right] \quad (8)$$

$$\text{Où: } R_{0\tau} = \{0, 1, \dots, \tau\} \quad \text{et} \quad R_{1\tau} = \{\tau + 1, \tau + 2, \dots, Z - 1\}$$

Il propose trois variantes de l'algorithme [5], utilisant des distributions Nakagami, Log-Normale et Weibull. Comme pour l'algorithme pour trouver le seuil d'erreur minimum, nous avons essayé d'implémenter ces trois variations avec python.

2.2.1 Loi Nakagami

Cette première modèle représente la densité de probabilité (décrit ci-dessous) standard pour les images SAR.

$$p_i(u|L_i, \gamma_i) = \frac{2\Gamma(2L_i)}{\Gamma^2(L_i)} \frac{\gamma_i^{L_i} u^{2L_i-1}}{(\gamma_i + u^2)^{2L_i}} \quad (9)$$

Où $L_i > 0$ et $\gamma_i > 0$ sont les paramètres de la loi Nakagami, et Γ c'est la fonction Gamma (nous avons utilisé la fonction de bibliothèque math du python pour l'implémentation).

À partir du développement mathématique décrit dans [5], nous pouvons estimer les paramètres à l'aide des formules suivantes :

$$2k_{1i} = \log \gamma_i \quad \text{et} \quad 2k_{2i} = \Psi(1, L_i) \quad (10)$$

Où la fonction $y = \psi(1, x)$ que nous devons inverser est la fonction trigamma. Nous pouvons résoudre le problème de façon numérique sous certaines conditions : Il faut notamment que la solution cherchée est pour x réel positif, qui c'est le cas. Avec $x > 0$, la fonction $y : x \Rightarrow \psi(1, x)$ est

monotone strictement décroissante, et en connaissant y , nous pouvons utiliser un solveur numérique pour trouver x tel que $\psi(1, x) = y$. Cela revient à chercher la valeur de x qui minimise la fonction $abs(f(x) = \psi(1, x) - y)$. Pour l'implémentation, nous avons utilisé la fonction `scipy.optimize.fsolve`.

Enfin, les estimateurs $\hat{k}_{1i\tau}$ et $\hat{k}_{2i\tau}$ sont donne par :

$$\hat{k}_{1i\tau} = \frac{\sum_{u \in R_{i\tau}} h(u) \log u}{\sum_{u \in R_{i\tau}} h(u)} \quad \text{et} \quad \hat{k}_{2i\tau} = \frac{\sum_{u \in R_{i\tau}} h(u) (\log u - \hat{k}_{1i\tau})^2}{\sum_{u \in R_{i\tau}} h(u)} \quad (11)$$

2.2.2 Loi Log-normale

La loi log-normale, qui est aussi acceptée pour représenter la densité de probabilité des images SAR, va nous donner des résultats similaires que si nous appliquons l'algorithme de seuil d'erreur minimum sur le log du rapport des images.

En remplaçant la formule, nous trouvons une fonction $J(t)$ très proche de la méthode standard, comme nous pouvons le voir ci-dessous.

$$J(\tau) = \sum_{i=0}^1 \hat{P}_{i\tau} \left(\frac{1}{2} \log \hat{k}_{2i\tau} - \log \hat{P}_{i\tau} \right) \quad (12)$$

2.2.3 Loi Weibull

La loi de Weibull a la densité de probabilité décrite ci-dessous ensemble comment estimer vos paramètres $\eta > 0$ et $\lambda > 0$ (les estimateurs de $\hat{k}_{1,2i}$ sont les mêmes pour la loi Nakagami).

$$p_i(u|\eta_i, \lambda_i) = \eta_i \lambda_i^{\eta_i} \frac{u^{\eta_i-1}}{(\lambda_i^{\eta_i} + u^{\eta_i})^2} \quad (13)$$

$$k_{1i} = \log \lambda_i \quad \text{et} \quad k_{2i} = \frac{2\Psi(1, 1)}{\eta_i^2} \quad (14)$$

Les auteurs montrent que les résultats avec les trois distributions sont similaires et que seulement l'application de l'algorithme de seuillage n'a pas de très bon résultat à cause du bruit. Pour l'améliorer dans ce cas, il faut débruiter les images et après trouver le seuil. Ce sont des résultats que nous allons vérifier au cours du projet.

2.3 Débruitage SAR2SAR

Dalasso, Denis et Tupin (2020) [3] présentent le SAR2SAR, un algorithme auto supervisé de deep learning pour débruiter les images SAR. L'algorithme a trois principales composantes, un passage au logarithme de l'image, un bloque de débruitage composé par le bruit résiduel obtenu pour une réseau U-net [7] et l'image passée au log et après l'application de l'exponentielle au fin du bloque de débruitage.

L'algorithme a trois étapes pour être formé. D'abord, il est formé dans les images composé par un bruit synthétique. Après, il y a un fine-tuning avec des images SAR réelles et enfin il est raffiné à l'aide de réflectivités SAR pré-estimées. Avec ces trois parties de la formé, les auteurs montrent que leur algorithme surpasse les algorithmes précédents dans la plupart des cas.

Comme est posé par [6], la meilleur approche pour détecter le changement entre deux images SAR c'est d'avant la débruiter. Avec cet algorithme, nous pourrons tester si l'application de cette débruitage vraiment améliore le résultat et combien est-il meilleur. Dans cette partie, nous n'avons pas implémenté l'algorithme, nous avons seulement utilisé les images débruités par lui.

3 Travaux réalisés sur les images SAR

Dans cette section nous montrerons les expériences qui nous avons fait dans le projet. D'abord, nous montrerons les donnés utilisés, après les expériences faites avec les donnés originales et débruités. Enfin, nous parlerons sur les méthodes qui nous utiliserons pour évaluer la qualité du seuil trouvé par chaque algorithme.

3.1 Données utilisé

Dans ce projet, nous avons utilisé deux images en différents dates d'un terrain avec une rivière. Nous pouvons regarder dans les images qu'il y a un changement dans la rivière entre les deux dates. Nous regardons aussi qu'il y a quelques petits changements dans le terrain mais ils ne sont pas très importants pour nous. L'objectif c'est seulement trouver le changement dans la rivière.

Sur les images dans les deux dates, nous avons une version originale, c'est-à-dire avec du bruit, une version débruité en utilisant l'algorithme proposé par [3] et le vérité terrain (VT), qui c'est une image binaire avec que de la rivière. Ci-dessous nous pouvons regarder ces images.

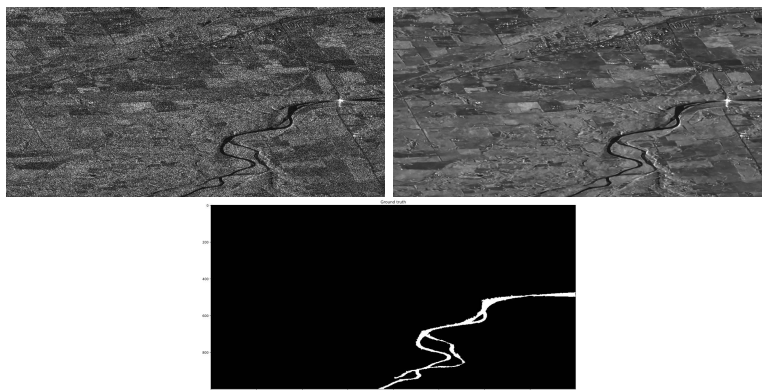


Figure 3: Exemple de l'image SAR bruité, débruité et du vérité terrain

3.2 Expériences

Pour détecter le changement entre les images SAR, nous avons utilisé l'algorithme implémentés dans l'histogramme du rapport entre les deux images $I_r = \frac{I_2}{I_1}$, où I_2 est l'image plus récent et I_1 est l'image plus ancienne.

Pour les images originales, il y a un bruit souvent impulsionnel, où l'intensité de chaque pixel est multiplié par le bruit $I = A * b$, où A est l'intensité originale du pixel et b est le bruit. Avec ce bruit impulsif, c'est difficile de faire un bon rapport entre les deux images, donc il faut les passer

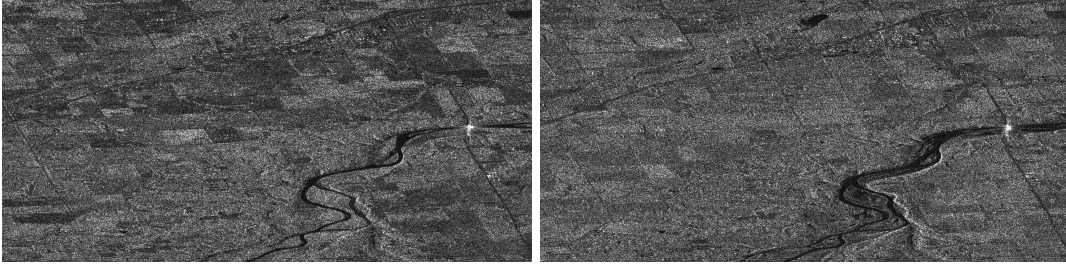


Figure 4: Exemple d'image SAR pour deux dates différentes

au log pour transformer le bruit multiplicatif en un bruit additif $\log A * b = \log A + \log b$. Pour les images débruitées en utilisant l'algorithme proposé par [3], nous considérons que les images n'ont pas du bruit, donc nous pouvons faire le rapport normalement.

Avec le rapport entre les deux images I_r , nous utilisons l'algorithme pour trouver le seuil qui meilleur détecte le changement entre les images avec les quatre lois implémentés: la loi Normal, la loi Nakagami, la loi Log-normal et la loi Weibull.

Après trouver le seuil, nous proposons deux masques différents pour faire une évaluation qualitative et quantitative de la qualité du seuil en détecter les changements de la rivière entre les deux images.

- Le premier masque a été obtenu à partir d'une dilatation (méthode de morphologie mathématique vu en cours) sur le plus grand véritable terrain de la rivière. Le résultat est une image binaire contenant la rivière à n'importe quelle date de l'image SAR. Nous avons utilisé ce masque pour limiter la détection de changement à la région de la rivière (il est où la plupart du changement va se passer), et ainsi, regarder mieux les résultats obtenus.
- Le deuxième masque a été obtenu à partir du changement entre le véritable terrain (VT) des deux images que nous voulons détecter le changement (implémentation avec la fonction `np.logical_xor`). Comme nous l'avons déjà décrit, la plus grosse partie de changement est concentrée sur la rivière, donc quand nous faisons la différence entre les VTs, nous allons obtenir le véritable changement de l'image. Ainsi, ce masque va être utilisé pour comparer les performances entre les méthodes appliquées.

Pour évaluer qualitativement le seuil trouvé par l'algorithme dans chaque loi et pour l'image originale et débruitée, nous convolutons le masque obtenu pour l'opération d'ouverture avec l'image binarisée après appliquer le seuil. Ainsi, nous pouvons regarder si la rivière a été bien détectée ou pas.

Pour évaluer quantitativement le seuil trouvé, nous utilisons quatre métriques: accuracy, précision, recall and F1-score. L'accuracy est une métrique que capture le nombre total de bonnes prédictions par rapport à toutes les prédictions 15.

$$Accuracy = \frac{TP + TN}{TN + TP + FN + FP} \quad (15)$$

La précision est une métrique que capture le nombre de bonnes prédictions de la classe positif par rapport à toutes les prédictions qui disent que le pixel est de la classe positif 16.

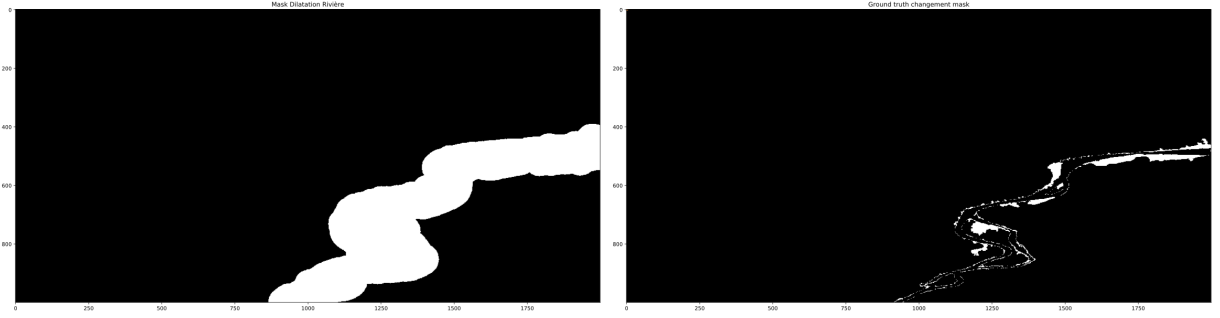


Figure 5: A gauche, le masque obtenu par la dilatation de la rivière; à droite, le masque de le vérité terrain des zones de changements de la rivière

$$Precision = \frac{TP}{TP + FP} \quad (16)$$

Le recall est une métrique que capture le nombre de bonnes prédictions de la classe positif par rapport à tous les vérités pixels de classe positif 17.

$$Recall = \frac{TP}{TP + FN} \quad (17)$$

Le F1'score est une métrique que rejoindre la précision et le recall en faisant la moyenne harmonique entre ces deux métriques. Dans cette métrique, le nombre de bonnes prédictions de la classe positif par rapport au nombre total de prédictions sur la classe positif et aussi par rapport au nombre total de vérités positifs sont important 18.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (18)$$

Pour chaque métrique, nous avons utilisé comme le VT le masque obtenu par l'opération de XOR entre les deux VT et comme la prédiction l'image binarisé après l'application du seuil.

4 Résultats

Pour évaluer les résultats nous avons séparé entre des résultats qualitatifs et des résultats quantitatifs. Pour les résultats qualitatifs, nous pourrons regarder le changement trouvé pour chaque méthode dans les données originales et débruités. Pour les résultats quantitatifs, nous pourrons comparer en utilisant des métriques proposés les différentes méthodes pour les données originales et aussi débruités.

4.1 Analyse qualitatif et quantitatif

Pour évaluer qualitativement les résultats de l'algorithme de seuil d'erreur minimum avec chaque loi, nous présentons deux images. L'image binarisé après l'application du seuil et l'image binarisé multiplié par le masque dilaté.

Pour évaluer quantitativement les résultats de l'algorithme de seuil d'erreur minimum avec chaque loi, nous avons comparé l'image binarisé pour le seuil trouvé (et aussi l'image binarisé multiplié par le masque dilaté) et le masque XOR. Pour faire cette comparaison, nous avons utilisé les quatre métriques précédemment proposés, l'accuracy, la précision, le recall et le F1-score.

Pour la bonne détection de la rivière, les métriques les plus importantes sont la précision, le recall et le F1-score. La précision nous montre si les pixels qui l'algorithme a prédit comme un pixel de changement de la rivière a été classifié correctement. Le recall nous montre combien de la rivière a été bien détecté par l'algorithme et la F1-score nous donne un score générale qu'utilise les deux métriques. L'accuracy nous donne une évaluation global, pas que de la rivière.

4.1.1 Données originales

Avec les données originales, c'est-à-dire avec du bruit, nous avons les évalué de deux façons différents, avec le rapport des images et avec le log du rapport des images. Pour le rapport entre les images nous l'évaluons avec l'algorithme de seuil d'erreur minimum avec les trois lois généralisés, et pour le log du rapport nous évaluons avec l'algorithme avec la loi normale. Nous avons déjà expliqué que le log du rapport entre les images est important pour diminuer l'influence du bruit impulsif dans l'image.

Ci-dessous nous pouvons regarder l'histogramme du rapport des images originales avec les seuils trouvés par l'algorithme de seuil d'erreur minimum avec les trois lois généralisés utilisés.

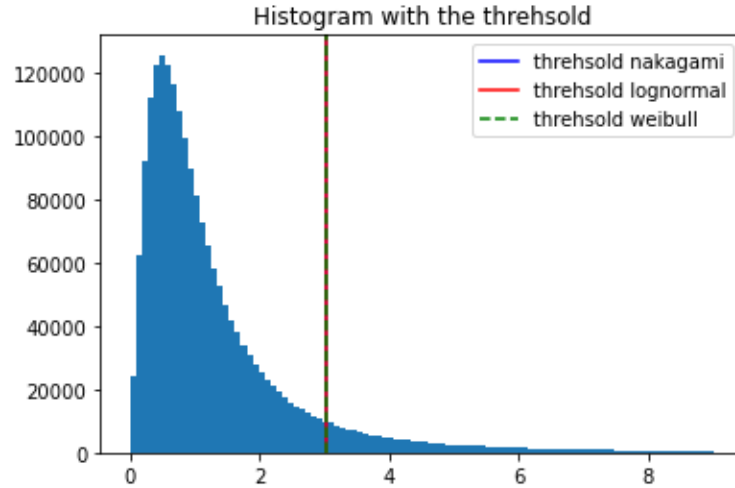


Figure 6: Histogramme du rapport des images originales avec les seuils trouvés

Avec cet histogramme, nous pouvons regarder que les trois seuils trouvés sont égaux. Cela est un résultat acceptable étant donné que la distribution de l'histogramme ne suit pas chacune des quatre lois. Ci-dessous nous pouvons regarder l'image binarisé après l'utilisation des trois lois et aussi sa multiplication par le masque dilaté.

Comme le seuil trouvé est le même pour les trois lois, nous avons que l'image binarisé est la même dans les trois cas. Cela est valide aussi pour les métriques d'évaluation, comme nous pouvons regarder dans la table 1 ci-dessous.

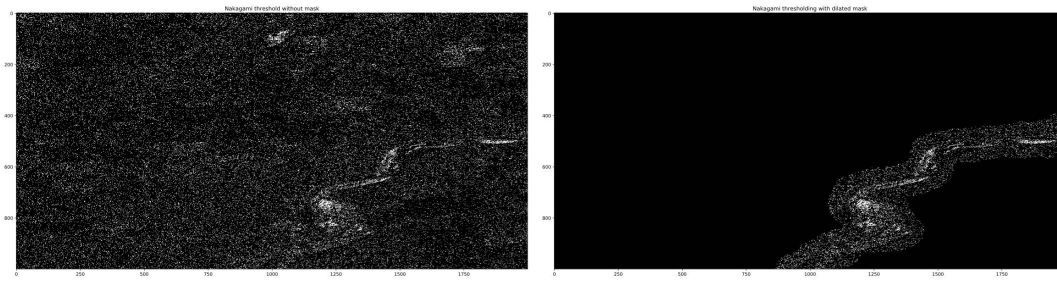


Figure 7: En gauche l'image originale binarisé multiplié par le masque dilaté, en droite l'image binarisé.

Table 1: Métrique par loi pour le rapport des données originales

Métrique	Loi Nakagami	Loi Log-noramle	Loi Weibull
Accuracy	0.901	0.901	0.901
Précision	0.034	0.034	0.034
Recall	0.295	0.295	0.295
F1-score	0.0610	0.0610	0.0610

Avec cette table, nous pouvons regarder que l'algorithme trouve un seuil avec une très bonne accuracy mais pas très bonne précision, recall et F1-score. Avec le recall, nous avons que presque 30% du changement de la rivière a été bien détecté, et avec la précision nous avons que seulement 3.4% des zones que l'algorithme a dit qui est une zone de changement de la rivière sont vraiment corrects. La bonne accuracy veut dire que les zones qui ne sont pas de changement sont bien prédisé pour l'algorithme. Cela veut dire que l'algorithme ne trouve pas bien la zone de changement de la rivière mais trouve bien où elle n'a pas changé.

Pour le log du rapport entre les deux images, nous utilisons l'algorithme de seuil d'erreur minimum avec la loi normale. Nous utilisons cette loi parce que nous espérons que l'histogramme du log du rapport suive une loi normale. Ci-dessous nous pouvons regarder son histogramme avec le seuil trouvé.

Avec cet histogramme, nous pouvons regarder que le seuil trouvé est au demi des données, donc nous espérons que il détectera une grande partie de la zone de changement de la rivière (bon recall) mais il détectera aussi beaucoup de zones de faux changement (mauvaise précision et accuracy). Ci-dessous nous pouvons regarder l'image binarisé après l'utilisation de l'algorithme avec la loi normale et aussi sa multiplication par le masque dilaté.

Comme le seuil trouvé est le même pour les trois lois, nous avons que l'image binarisé est la même dans le trois cas. Cela c'est valide aussi pour les métriques d'évaluation, comme nous pouvons regarder dans la table 2 ci-dessous.

Avec ces métriques, nous avons pu observer qu'est-ce que nous avons dit avant. Comme l'algorithme a détecté beaucoup de zones de changement, il a mieux trouvé la zone de changement de la rivière, donc nous avons un meilleur recall par rapport au recall obtenu dans la table 2. Au même temps, la précision et l'accuracy ont eu des résultats beaucoup pires que les résultats présentés dans la table 1.

Comme nous avons une métrique mieux pour chaque approche (rapport et log du rapport) dans

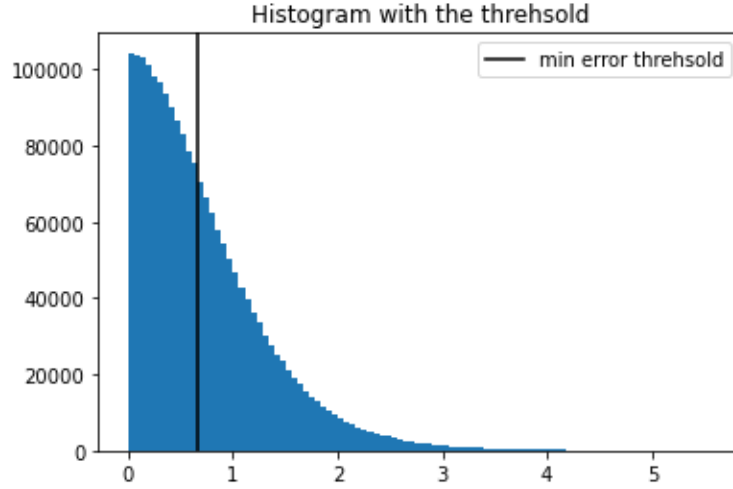


Figure 8: Histogramme du log du rapport des images originales avec le seuil trouvé

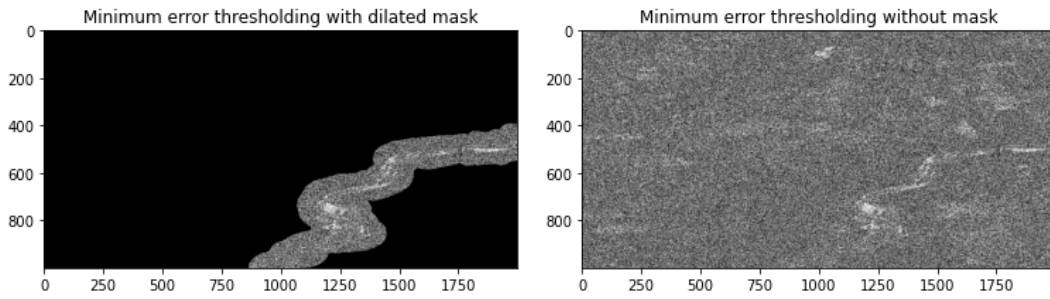


Figure 9: En gauche l'image originale binarisé, en droite l'image binarisé multiplié par le masque dilaté.

Table 2: Métrique pour le log du rapport des données originales

Métrique	Loi Normale
Accuracy	0.558
Précision	0.013
Recall	0.566
F1-score	0.027

le sujet de bien détecter le changement de la rivière, nous pouvons utiliser la métrique F1-score pour dire quel approche a eu le meilleur résultat. La métrique F1-score, comme nous avons dit précédemment, est la moyenne harmonique entre la précision et le recall, donc elle nous donne une meilleur évaluation sur la détection de la rivière. Nous avons un meilleur F1-score pour la première approche, en utilisant l'algorithme de seuil d'erreur minimum généralisé dans le rapport entre les

deux images.

Alors, nous pouvons conclure que pour ce cas utiliser appliquer le log avant faire du rapport entre les deux images n'a pas amélioré le résultat.

4.1.2 Données débruitées

Pour évaluer les données débruitées, nous pouvons utiliser seulement le rapport entre les deux images. Dans ce cas, le log du rapport n'a pas de sens parce nous espérons que n'y a pas de bruit multiplicatif à être transformé en bruit additif, en vrai nous espérons qu'il n'y ait aucun bruit. Alors, nous utilisons l'algorithme de seuil d'erreur minimum avec les quatre lois pour évaluer lequel trouve le meilleur seuil.

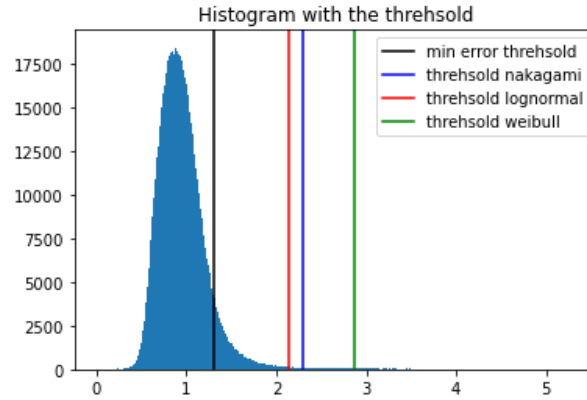


Figure 10: Histogramme du rapport des images débruitées avec les seuils trouvés

Nous pouvons regarder que pour chaque loi l'algorithme trouve un seuil différent. Ci-dessous nous pouvons regarder les images binarisées après l'utilisation de l'algorithme avec chacune des lois utilisés et aussi sa multiplication par le masque dilaté.

Nous pouvons regarder que l'algorithme avec la loi normale trouve un seuil plus petit que les seuils trouvés pour des autres lois, donc dans sa image binarisé il y a beaucoup plus de pixels préditsés comme zone de changement de la rivière. Ci-dessous nous pouvons analyser quantitativement cet effet dans la table 3.

Table 3: Métrique par loi pour le rapport des données debruités

Métrique	Loi Normale	Loi Nakagami	Loi Log-noramle	Loi Weibull
Accuracy	0.920	0.989	0.989	0.990
Précision	0.088	0.501	0.461	0.587
Recall	0.673	0.312	0.349	0.189
F1-score	0.155	0.384	0.397	0.286

Avec ces métriques, nous pouvons observer que vraiment l'algorithme avec la loi normale a bien détecté beaucoup plus du changement de la rivière (67%) contre seulement 35% que c'est le meilleur recall entre les autres trois lois. Cependant, pour trouver un seuil beaucoup plus petit et prédiser

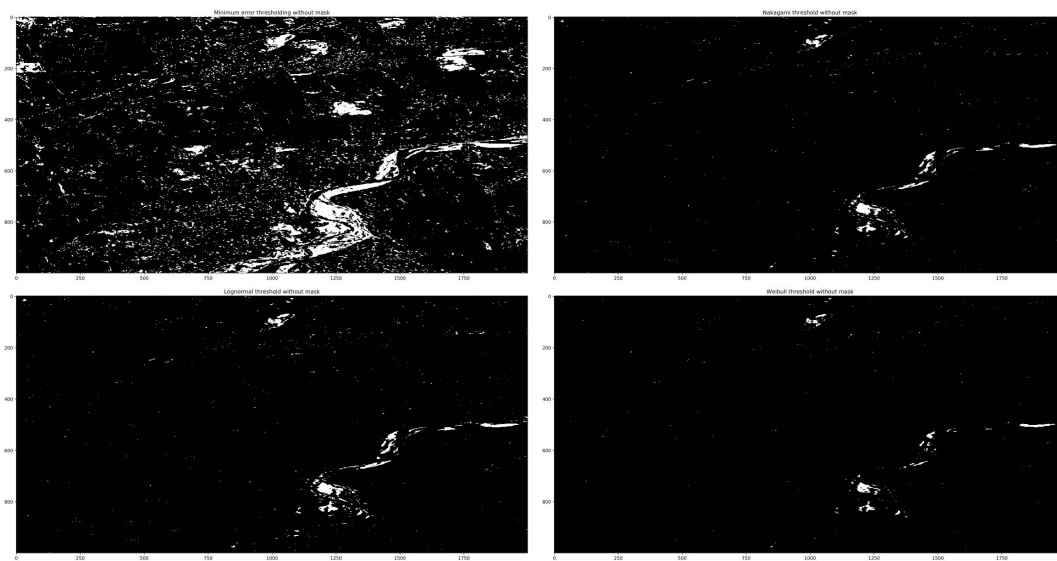


Figure 11: Image débruité binarisé après appliquer le seuil trouvé pour l'algorithme de seuil d'erreur minimum avec la loi Noramlle (haut gauche), la loi Nakagami (haut droite), la loi Log-normale (bas gauche) et la loi Weibull (bas droite).

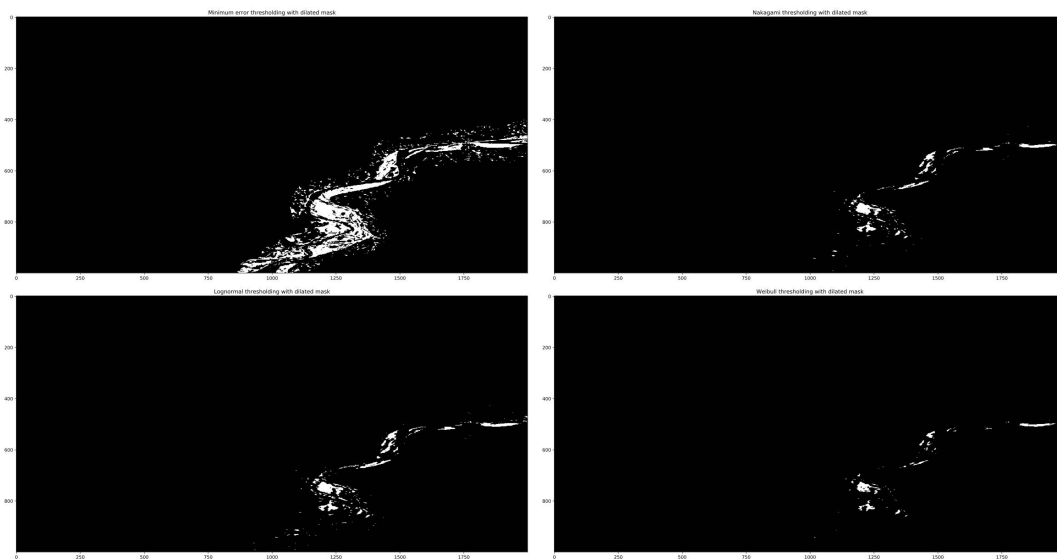


Figure 12: Image débruité binarisé et multiplié par le masque dilaté après appliquer le seuil trouvé pour l'algorithme de seuil d'erreur minimum avec la loi Noramlle (haut gauche), la loi Nakagami (haut droite), la loi Log-normale (bas gauche) et la loi Weibull (bas droite).

plusieurs pixels comme zone de changement, la précision de l'algorithme avec la loi normale est beaucoup pire que la précision des autres lois.

Pour évaluer globalement la bonne détection de la rivière, nous utilisons la métrique F1-score parce qu'elle donne la moyenne harmonique entre la précision et le recall. Dans ce métrique, nous avons que l'algorithme avec la loi log-normale a trouvé le meilleur seuil.

Pour l'accuracy, qui analyse globalement le résultat de l'algorithme, nous avons que les trois lois généralisés proposés par [6] ont presque le même résultat, qui est meilleur que le résultat obtenu pour la loi normale proposé par [5].

En restreignant le calcul des performances avec le masque dilaté, nous avons observé que nous obtenions des résultats plus satisfaisants (Table 4). Cela était attendu, car comme la plupart des changements temporels se produisent dans la rivière, le restreindre à cette zone nous donnera moins de faux positifs. Pour cette raison, nous pouvons observer un meilleur résultat principalement dans les métriques de précision et F1, pour toutes les lois.

Table 4: Métrique par loi pour le rapport des données débruitées avec masque

Metric	Minimum Erro	Loi Nakagami	Loi Log-noramle	Loi Weibull
Accuracy	0.974	0.991	0.991	0.991
Précision	0.248	0.666	0.625	0.761
Recall	0.673	0.312	0.349	0.189
F1-score	0.362	0.425	0.447	0.303

5 Conclusion

Dans ce projet nous avons pu regarder le fonctionnement d'un algorithme qu'essaye de trouver un seuil d'erreur minimum en utilisant 4 lois de distributions à priori, la loi Normale, la loi Nakagami, la loi Log-normale et la loi Weibull. Après, nous avons utilisé cet algorithme pour trouvé les zones de changement entre deux images SAR, plus spécifiquement, le changement d'une rivière dans une image météorologique dans deux dates différents. Nous avons regardé ce changement en utilisant l'image originale (avec du bruit) et aussi sa version débruité.

Dans [6], l'auteur argument que les trois lois de distribution proposés par lui n'améliorent pas de résultat, c'est qu'améliore vraiment le résultat c'est débruiter les images avec d'appliquer l'algorithme. Après les expériences que nous avons fait, nous avons regardé ce même résultat. Pour l'image originale, l'algorithme avec les trois lois trouvent le même seuil et n'ont pas un bon résultat, mais pour les données débruitées, ils trouvent pour chaque loi un seuil différent avec des métriques beaucoup mieux qu'ils ont eu avant.

Nous avons pu regarder aussi que l'algorithme de débruitage proposé par [3] donne un très bon résultat. Nous pouvons confirmer cela en observent les images réelles originales et débruitées et aussi par rapport aux résultats quantitatives obtenus. En plus, la grand amélioration quantitatif quand nous avons utilisé les données débruitées par rapport aux données originales montrent la qualité de la méthode de débruitage.

References

- [1] R. Anniballe, F. Noto, T. Scalia, C. Bignami, S. Stramondo, M. Chini, and N. Pierdicca. Earthquake damage mapping: An overall assessment of ground surveys and vhr image change detection after l’aquila 2009 earthquake. *Remote sensing of environment*, 210:166–178, 2018.
- [2] E. Colin Koeniguer and J.-M. Nicolas. Change detection based on the coefficient of variation in sar time-series of urban areas. *Remote Sensing*, 12(13):2089, 2020.
- [3] E. Dalsasso, L. Denis, and F. Tupin. SAR2SAR: a self-supervised despeckling algorithm for SAR images. *arXiv preprint arXiv:2006.15037*, 2020.
- [4] F. Gao, X. Wang, Y. Gao, J. Dong, and S. Wang. Sea ice change detection in sar images based on convolutional-wavelet neural networks. *IEEE Geoscience and Remote Sensing Letters*, 16(8):1240–1244, 2019.
- [5] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern recognition*, 19(1):41–47, 1986.
- [6] G. Moser and S. B. Serpico. Generalized minimum-error thresholding for unsupervised change detection from sar amplitude imagery. *IEEE Transactions on Geoscience and Remote sensing*, 44(10):2972–2982, 2006.
- [7] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.