Détection d'anomalies en SAR basée sur les réseaux génératifs

Max MUZEAU[1, 2], Chengfang REN[1], Sébastien ANGELLIAUME[3], Mihai DATCU[4,5], Jean-Philippe OVARLEZ[1,2]

[1]CentraleSupélec SONDRA, Université Paris-Saclay, 3 rue Joliot Curie, 91190 Gif-sur-Yvette, France,

[2]DEMR, ONERA, Université Paris-Saclay, 8 Chemin de la Hunière, 91123 Palaiseau, France,

[3]DEMR, ONERA, F-13661 Salon cedex Air, France,

[4]German Aerospace Center (DLR), 82234 Weßling, Germany,

[5]Politehnica University of Bucharest (UPB), 060042 Bucharest, Romania

max.muzeau@centralesupelec.fr, chengfang.ren@centralesupelec.fr,
sebastien.angelliaume@onera.fr, Mihai.Datcu@dlr.de, jean-philippe.ovarlez@onera.fr

Résumé – Ce papier propose une méthode de détection d'anomalies pour l'imagerie SAR basée sur l'apprentissage profond. Elle ne nécessite pas de vérité terrain des anomalies, ce qui répond à un problème récurrent en télédétection : le manque de données annotées pour entraîner les réseaux de neurones. Le modèle proposé combine un *adversarial autoencoder* suivi d'un détecteur statistique de changement basé sur la matrice de covariance. Une étape de *despeckling* est préalablement effectuée, ce qui permet de filtrer le bruit de speckle et d'améliorer significativement les performances de détection.

Abstract – This paper proposes an anomaly detection method for SAR imagery based on deep learning. It does not require ground truth of anomalies, which addresses a recurrent problem in remote sensing: the lack of labeled data to train neural networks. The proposed model combines an adversarial autoencoder followed by a statistical change detector based on the covariance matrix. A despeckling step is first performed, which allows to filter the speckle noise and to significantly improve the detection performances.

1 Introduction

La détection d'anomalies est un sujet fondamental en traitement d'image. Etudié dans de nombreux domaines comme l'imagerie médicale [1], la vidéo [2], l'imagerie hyperspectrale [3] et dans notre cas, l'imagerie radar à synthèse d'ouverture (SAR) [4], la disponibilité de telles données s'étant fortement accrue ces dernières années suite à la mise en orbite de nombreux satellites tels que TerraSAR-X et Sentinel. Même si les données sont maintenant massivement disponibles, le manque d'annotation de celles-ci demeure un problème crucial pour l'utilisation d'algorithmes supervisés. De plus, le nombre de zones anormales est très largement inférieur au nombre de zones normales, ce qui rend un entraînement supervisé encore plus complexe. Dans ce contexte, l'emploi d'algorithme non supervisé est généralement préféré, parmi lesquels un des plus utilisé est le détecteur de Reed-Xiaoli [5]. Récemment, de nombreux travaux se sont basés sur les réseaux de neurones profonds, principalement grâce à des autoencodeurs et des GAN [6, 7] (voir § 2.2.1 et 2.2.2). Ils permettent de réduire fortement la taille des données d'entrée tout en préservant l'information. Leur utilisation permet de plus de ne pas reconstruire les zones anormales en sortie du décodeur (voir § 2.2.3). Certains auteurs ont appliqué ces algorithmes pour l'imagerie SAR [8, 4], sans pour autant s'attacher à la problématique du bruit de speckle, qui pourtant accroît fortement la difficulté [9].

Un des objectif de ce papier est de répondre à ce problème grâce à l'ajout d'une étape de *despeckling*. Une fois l'effet du bruit minimisé, un *adversarial autoencodeur* est utilisé pour obtenir une image sans anomalie. Finalement, une approche de détection de changement entre images est utilisée. L'originalité réside alors dans l'utilisation conjointe d'une méthode de détection de changement usuelle et d'autoencodeurs pour la détection d'anomalies.

2 Méthode proposée

2.1 Filtrage du speckle

Les images SAR sont impactées par un bruit de *speckle*. Il est dû aux nombreux rétro-diffuseurs qui se somment de manière cohérente au sein d'une même cellule de résolution. Un modèle de ce bruit est proposé par Goodman [9]. Pour réduire son effet sur les images SAR, le réseau de neurones SAR2SAR [10] pré-entraîné est utilisé sur les images en intensité.

Ces images subissent généralement une pondération pour réduire les lobes secondaire des fortes cibles ou ayant nécessité un sur-échantillonnage pour garantir des pixels carré, mais la contrepartie est la présence d'une corrélation spatiale entre pixels. Un des avantages des méthodes de *despeckling* par Deep Learning par rapport aux méthodes statistiques est la prise en compte de cette corrélation.

L'algorithme SAR2SAR est une adaptation pour le SAR de l'algorithme *noise2noise* [11]. Il répond à un problème majeur qui est l'absence d'images sans speckle comme référence. Dans [11], les auteurs montrent qu'il n'est pas nécessaire de

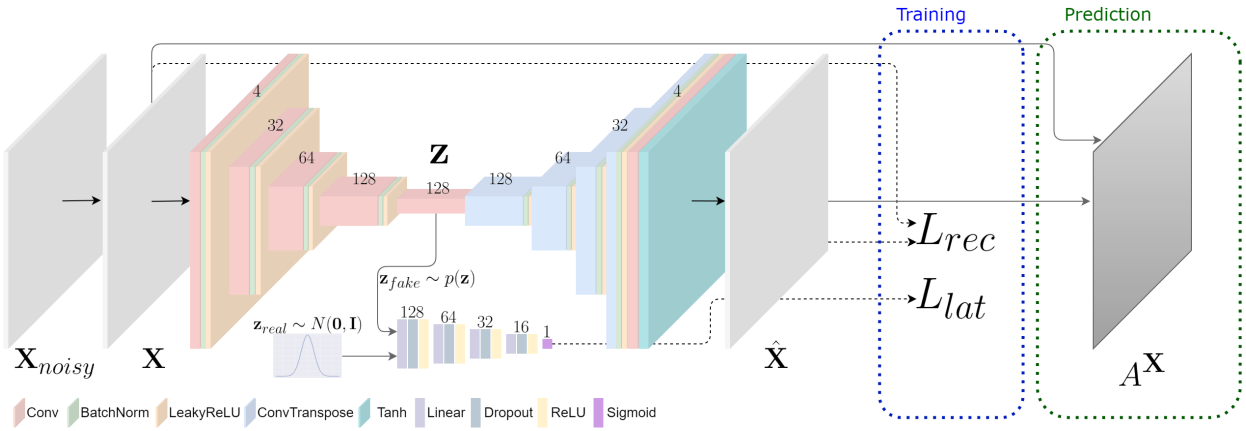

FIGURE 1 – Architecture utilisée pour détecter les anomalies

disposer d'une image sans bruit afin d'entraîner un réseau à partir de l'équation suivante :

$$\underset{\mathbf{Z}}{\operatorname{argmin}}\, \mathbf{E}_{\mathbf{Y}} \; k\mathbf{Z} \quad \mathbf{Y}k_F^2 \quad = \mathbf{E}_{\mathbf{Y}}[\mathbf{Y}] \tag{1}$$

où $\mathbf{Z}$ représente l'image en sortie du réseau et $\mathbf{Y}$ une image de la même zone avec une réalisation différente du bruit. D'après (1), minimiser l'erreur quadratique moyenne revient à prédire par le réseau la valeur moyenne des données bruitées. Il faut donc que le bruit soit centré et que les réalisations soient décorrélées. Dans le cas de données biaisées (ce qui est le cas des images SAR en log intensité), il convient de soustraire la moyenne du bruit au résultat s'il est additif. L'utilisation de cette méthodologie nécessite l'emploi de deux données de télédétection acquises sur la même zone mais avec un décalage temporel ou géométrique suffisant pour assurer deux réalisations de bruit différentes. Les apports de SAR2SAR concernent l'utilisation d'une fonction de coût adaptée à la distribution des images SAR ainsi qu'une étape de compensation des éventuels changements sur la baseline temporelle entre les acquisitions, ce qui est le cas pour la majorité des satellites d'observation de la Terre actuellement en orbite dans l'espace.

Cet algorithme fonctionne sur des images monovariées (mono canal). Dans le cas de données multivariées, il sera nécessaire de l'appliquer successivement sur chacun des canaux (polarimétrique par exemple) tout en ajustant la normalisation pour conserver une même dynamique entre les canaux. Une illustration est donnée en Fig. 2.

2.2 Reconstruction sans anomalies

Le processus de reconstruction requiert l'introduction des deux algorithmes suivants. On utilisera les notations $E(.)$, $D(.)$ et $D_c(.)$ pour définir respectivement l'encodage, le décodage et la sortie de discriminateur.

2.2.1 Generative adversarial networks

Un *Generative Adversarial Network* (GAN) [12] est un algorithme d'apprentissage non supervisé ayant pour but initial la génération d'images de synthèse. Un générateur et un discriminateur sont entraînés conjointement, le premier devant générer une image la plus réaliste possible, à partir d'un vecteur source aléatoire de faible dimension distribué selon une loi multivariée choisie, et le deuxième devant vérifier la représentativité de cette image. Cette architecture est utilisée dans notre cas, non pas pour générer des images, mais pour l'encoder dans un espace latent de faible dimension et dont la loi du vecteur aléatoire est connue. (voir Fig. 1)

2.2.2 Adversarial Autoencoder

Un *adversarial autoencoder* [13] est composé d'un générateur (encodeur+décodeur) et d'un discriminateur dans l'espace latent. Le générateur se décompose en deux parties : l'encodeur qui va transformer une entrée $\mathbf{X}$ (ici, une image SAR multivariée, $\mathbf{X}$ 2 $\mathbf{R}^{h \quad w \quad c}$ de taille $h \quad w$ et de profondeur c, avec h et w le nombre de pixels en azimut et en distance possédant c canaux de polarisation) en un vecteur latent $\mathbf{z} = E(\mathbf{X})$ suivant une distribution *a posteriori* $\mathbf{z} \quad p(\mathbf{z})$. Le décodeur va, à partir de ce vecteur, faire une estimation de l'entrée $\hat{\mathbf{X}} = D(\mathbf{z})$. La différence avec un *autoencoder* est l'ajout d'une régularisation de l'espace latent. Un discriminateur (à la manière d'un GAN) contraint $\mathbf{z}$ à suivre une distribution gaussienne en déterminant si le vecteur d'entrée est issue d'une loi gaussienne centrée réduite ($\mathbf{z}_{real} \quad N(0; I)$) ou non ($\mathbf{z}_{fake} \quad p(\mathbf{z})$).

Le générateur et le discriminateur sont entraînés en deux phases successives :

1 - Erreur de reconstruction : cette phase permet au générateur de reconstruire une image avec une erreur pixel à pixel la plus faible. On minimise donc l'erreur de reconstruction L_{rec} [1] pour les anomalies, ces zones sont reconstruites moins fidèlement ce qui les rendent plus identifiables :

$$L_{rec} = \frac{1}{hwc} \sum_{i;j;k} \mathbf{X}_{i;j;k} \quad \hat{\mathbf{X}}_{i;j;k} \; {}_1 : \tag{2}$$

2 - Erreur de régularisation : cette étape permet de contrôler la distribution de l'espace latent et d'obtenir également une meilleure reconstruction [8, 13]. Pour cela, les poids des ré-

1. Une norme L^1 est ici privilégiée par rapport à une norme L^2 afin de ne pas pénaliser le réseau lorsque l'écart entre $\mathbf{X}$ et $\hat{\mathbf{X}}$ est élevé.

seaux E et D_c doivent être estimés de manière à obtenir :

$$\min_E \max_{D_c} L_{lat} \quad \mathbb{E}_{\mathbf{z}_{real} \quad N(\mathbf{0};\mathbf{I})} [\log(D_c(\mathbf{z}_{real})] + \mathbb{E}_{E(\mathbf{X}) \quad p(z)} [\log(1 \quad D_c(E(\mathbf{X}))] \; : \quad (3)$$

2.2.3 Méthode d'utilisation

En synthèse, le modèle de reconstruction ainsi exposé ne permettra *a priori* pas une reconstruction suffisamment exhaustive de l'ensemble des motifs en entrée. L'objectif étant de reconstruire uniquement ceux suffisamment présents dans les données, les motifs rares seront ainsi vus comme des "anomalies".

2.3 Détection de changement

Une fois l'étape de reconstruction effectuée, une méthode classique de détection de changements (comparaison des matrices de covariance entre chaque pixel $(k;l)$ des 2 images SAR $\mathbf{X}$ et $\hat{\mathbf{X}}$) est appliquée suivant la dimension c comme :

$$A^{\mathbf{X}}(k;l) = \quad \hat{}^{\mathbf{X}}_{k;l} \quad \hat{}^{\hat{\mathbf{X}}}_{k;l} \quad ^2_F \qquad (4)$$

où $\hat{}^{\mathbf{X}}_{k;l}$ (resp. $\hat{}^{\hat{\mathbf{X}}}_{k;l}$) est l'estimée de la covariance locale de l'image $\mathbf{X}$ (resp. de l'image $\hat{\mathbf{X}}$) dans une fenêtre carrée $B_{k;l}$ centrée sur le pixel $(k;l)$ au sens du Maximum de Vraisemblance pour une loi gaussienne, ce qui donne l'estimateur communément appelé Sample Covariance Matrix définie par :

$$\hat{}^{\mathbf{X}}_{k;l} = \frac{1}{jB_{k;l}j} \sum_{i;j \, 2 B_{k;l}} \mathbf{X}_{i;j} \quad \mu^{\mathbf{X}}_{k;l} \quad \mathbf{X}_{i;j} \quad \mu^{\mathbf{X}}_{k;l} \quad ^T ; \qquad (5)$$

avec $\mu^{\mathbf{X}}_{k;l} = jB_{k;l}j \quad ^1 \sum_{i;j \, 2 B_{k;l}} \mathbf{X}_{i;j}$:

3 Application pratique

3.1 Données SAR polarimétriques

Les données utilisées sont des images SAR acquises en bande X par le capteur aéroporté SETHI de l'ONERA sur la région de Nîmes-Garons en 2014. Ce sont des données en polarisation complète $(HH; HV; VH; VV)$, soit $c = 4$, l'angle d'incidence est de 45° en milieu de fauchée et les résolutions spatiales de 0.27 m en distance (géométrie slant) et 0.34 m en azimut. La zone imagée est principalement constituée de parcelles agricoles, soit des zones relativement homogènes sans anomalies. L'entraînement est réalisé de façon non supervisée, aucun tri n'a été effectué en pré-traitement pour exclure les anomalies, elles apparaissent avec très peu d'occurence. Aucune vérité terrain concernant les anomalies n'est donc nécessaire, ni une séparation des données de test. Le réseau s'entraîne en aveugle. Les images en log et normalisées entre 0 et 1 sont utilisées en entrée de générateur.

3.2 Architecture et paramètres

Nous avons mis en oeuvre la méthodologie avec et sans filtrage du speckle. Comme illustré Fig. 1, l'encodeur est constitué d'un enchaînement de couches de convolution et de normalisation par batch suivi d'un *leakyReLU* (pente de 0.1 pour les valeurs négatives). Pour le décodeur, l'architecture est la même, en replaçant la couche convolutive par une convolution transposée [14]. En suivant les recommandations de l'article DCGAN [15], on utilise un pas de 2 pour les opérations de convolution (noyau 4 4) afin de s'affranchir de l'étape de *max-pooling*. Il en est de même pour la convolution transposée qui donne une *feature map* de sortie deux fois plus large en hauteur et en largeur que celle en entrée. Les couches du discriminateur sont totalement connectées suivies d'un *dropout* (probabilité de 0.2) et d'un *ReLU*. On a une sigmoïde en sortie du discriminateur et une tangente hyperbolique en sortie du générateur.

Les images en entrée de l'encodeur sont découpées en patchs de taille 64 64 avec un recouvrement 50% grâce à une fenêtre glissante de pas 32, ce qui donne 116200 patches pour l'entraînement. Les poids sont optimisés via l'optimiseur ADAM [16], avec un *learning rate* de 10 ^{3} que ce soit pour L_{rec} (eq. 2) ou pour L_{lat} (eq. 3). Le modèle est implémenté avec Pytorch et la carte graphique utilisée est une GeForce RTX 3080 Ti 12G. L'entraînement dure 1h40 pour 20 époques. Pour la prédiction, les matrices de covariance nécessaires à la détection de changement sont calculées sur des fenêtres glissantes de taille 7x7.

3.3 Résultats

Chaque canal des images SAR polarimétriques est seuillé pour réduire l'impact des forts contributeurs sur la dynamique de l'image. Une fois la détection de changement éffectuée, le résultat est normalisé entre 0 et 1 et transformé en échelle log. On présente les résultats de l'algorithme de détection d'anomalies avec et sans l'étape de despeckling sur la Fig. 3.

Si l'algorithme de reconstruction est entraîné sur les images $\mathbf{X}_{noisy}$, l'*autoencoder* reconstruit des images sans le *speckle*. Ce phénomène peut s'expliquer de la même manière que l'algorithme *noise2noise* : de nombreux patches ont un fond similaire mais une réalisation différente du speckle. Une autre explication peut être la capacité de reconstruction du réseau qui n'est pas suffisante pour conserver le *speckle*. En comparant les matrices de covariance de $\mathbf{X}_{noisy}$ et de $\hat{\mathbf{X}}_{noisy}$, on retrouve ce bruit, rendant la détection d'anomalies plus difficile.

Au contraire, quand on entraîne le réseau sur les images $\mathbf{X}$ filtrée, il ne faut plus reconstruire le *speckle*, ce qui permet une meilleure reconstruction ainsi qu'une carte de détection $A^{\mathbf{X}}$ avec un taux de fausses alarmes réduit. De plus, l'estimation de la matrice de covariance avec l'eq. 5 est optimisée pour un bruit gaussien, alors que le logarithme de l'intensité du bruit est, lui, distribué selon un cas particulier de la loi de Fisher-Tippett $p(s) = e^s e^{\quad e^s}$. Comme attendu, les résultats semblent visuellement meilleurs lorsque l'étape de débruitage est préalablement appliquée.

Dans les extraits analysés Fig. 2, on observe des anomalies produites par des échos de forte énergie mais aussi par des points et des lignes qui répondent différemment selon les polarisations. La figure 3 montre que ces anomalies sont détectées dans la plupart des cas mais elles sont mieux mises en évidence avec la carte de détection $A^{\mathbf{X}}$. Sans *despeckling*,