HAL
open science

Classification d'objets enfouis par un modèle du second d'ordre en utilisant des données GPR

Douba Jafuno, Ammar Mian, Guillaume Ginolhac, Nickolas Stelzenmuller

To cite this version:

Douba Jafuno, Ammar Mian, Guillaume Ginolhac, Nickolas Stelzenmuller. Classification d'objets enfouis par un modèle du second d'ordre en utilisant des données GPR. GRETSI, Aug 2023, Grenoble (38000), France. hal-04234436

HAL Id: hal-04234436

https://hal.science/hal-04234436v1

Submitted on 10 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Classification d'objets enfouis par un modèle du second d'ordre en utilisant des données GPR

Douba Jafuno[1], Ammar Mian[1], Guillaume Ginolhac[1], Nickolas Stelzenmuller[2]

[1]Université Savoie Mont-Blanc, LISTIC, Annecy, France

[2]Géolithe, Grenoble, France

douba.jafuno,ammar.mian,guillaume.ginolhac@univ-smb.fr,nickolas.stelzenmuller@geolithe.com

Résumé – Dans ce papier, nous considérons un problème de classification multi-classes des signaux de radar à pénétration de sol (GPR pour Ground Penetrating Radar). Ce type de système permet de fournir une image du sous-sol et en particulier des objets enfouis. A cause du mouvement du RADAR, la réponse des objets est hyperbolique. Nous proposons donc d'effectuer cette classification à partir d'imagettes de ces hyperboles. Pour ce faire, le modèle proposé utilise des matrices de covariances calculées en sortie de filtre d'un réseau peu profond. L'estimation est effectuée avec un pipeline comprenant une étape de backpropagation et des couches convolutives adaptées aux matrices de covariances sont ensuite ajoutées. Ces dernières permettent d'effectuer une réduction de dimension avant d'effectuer la classification en fin de réseau. Différentes couches basées sur des concepts de géométrie riemannienne gèrent les spécificités des matrices de covariances dans la classification. L'approche proposée permet alors d'effectuer une classification automatique des objets enfouis avec les données étiquetées disponibles. Nous testons notre approche sur un jeu de données réelles fourni par la société Géolithe.

Abstract – In this paper, we consider a multi-class classification problem of ground penetrating radar (GPR) signals. This type of system provides an image of the underground, particularly of buried objects. Because of the movement of the RADAR, the response of the objects is hyperbolic. We therefore propose to perform this classification from images of these hyperbolas. The proposed model uses covariance matrices computed from the filter outputs of a shallow network. The estimation is performed with a pipeline including a backpropagation step and convolutional layers adapted to the covariance matrices. The latter allows for a reduction of the data space. Then different layers based on Riemannian geometry concepts allow considering the properties of the covariance matrices in the classification. The proposed approach then allows to perform an automatic classification of buried objects by training on available labeled data. We test our approach on a real dataset provided by the Geolithe company.

1 Introduction

Le GPR (GPR pour Ground Penetrating Radar) est un RADAR conçu pour imager le sous-sol [1]. En se déplaçant, il émet régulièrement une onde qui va se réfléchir sur des objets enfouis (un tuyau, une mine ou bien une cavité). À la fin de l'acquisition, on obtient alors une image du sous-sol, appelée radargramme, avec les différents objets. Dû au déplacement du RADAR, les objets ont une forme hyperbolique. La plupart des traitements consistent alors à débruiter l'image dans le but de détecter et localiser ces objets [2]. Après cette étape, il semble alors intéressant de rajouter une étape de classification pour déterminer la nature de l'objet. Cette étape est importante par exemple dans la prévention des risques pour éviter les chutes de rochers ou des glissements de terrain. La plupart des méthodes de classification en GPR se basent sur la forme de l'hyperbole en utilisant des méthodes classiques de machine learning [3] ou de deep learning [4, 5]. Néanmoins, l'imagette contenant l'hyperbole est souvent très bruitée et malheureusement donne assez peu d'informations sur l'objet. Pour avoir de bon résultat de classification, il est alors nécessaire d'avoir beaucoup de données labellisées et donc une étape d'entrainement importante et lourde en temps de calcul.

Pour gérer des données fortement bruitées, il est connu que les matrices de covariances sont des caractéristiques plus pertinentes que le signal ou l'image brute pour obtenir de bons résultats de classification. Dans ce cadre il est aussi intéressant d'intégrer la géométrie Riemannienne pour tenir compte des propriétés spécifiques des matrices de covariances. Cette approche a montré sa pertinence pour la classification de signaux EEG [6] mais a aussi obtenu des résultats préliminaires intéressants en classification GPR [7]. Dans ce dernier travail, les covariances sont construites à partir des sorties de filtres de convolution d'un réseau CNN (Covolutional Neural Network) en s'inspirant de [8]. Cette étape permet d'obtenir des matrices avec une grande richesse sur l'imagette. Les résultats de [7] sont plutôt encourageants pour une classification considérant seulement deux classes mais les performances se dégradent en multi-classes. On propose

donc dans ce papier un nouveau *framework* basé sur des couches de convolution adaptées aux matrices de covariances. Le principe de base est de garder un réseau léger avec peu de couches grâce l'utilisation des matrices de covariance au lieu des simples images. Plus particulièrement, le *framework* proposé dans dans ce papier est basé sur la combinaison de deux pipelines proposées dans [9, 10] et [11]. L'approche proposée sera testée sur un jeu de données réelles fourni par la société Géolithe et nous nous intéressons plus particulièrement à la classification multi-classes.

2 Principe du GPR

Le radar à pénétration de sol (GPR pour Ground Penetrating Radar) est un RADAR dont l'onde émis est un Ricker comme illustré sur la Figure 1. Cette onde se propage dans le sol et se réfléchit sur un éventuel objet. Pour avoir une image du sol, le radar se déplace le long d'un axe et l'émission-réception est effectué à intervalles régulier. C'est pourquoi l'image d'un objet a une forme hyperbolique. Cette image est appelée radargramme et un exemple en est donné en figure 2. En fonction du sol, des couches traversées et bien sur des propriétés électromagnétiques et géométriques, le signal réfléchi aura une forme particulière. Notre but est donc de classifier ces objets en se servant d'imagettes contenant l'hyperbole leur correspondant. On supposera que les hyperboles sont détectées et isolées à partir du radargramme (une petite explication du pré-traitement sera donnée en début de partie 4).

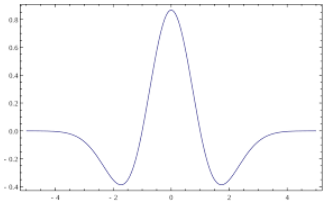

FIG. 1: Exemple d'un signal Ricker.

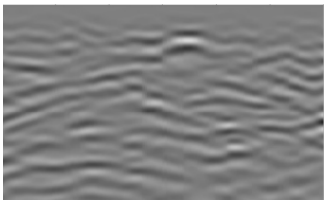

FIG. 2: Exemple d'un radargramme GPR.

3 Méthodes de Classification

Les différents pipelines utilisés dans ce papier sont illustrés sur la Figure 3. Tout d'abord, nous construisons les matrices de covariances. Après avoir pré-traité nos imagettes d'hyperbole en nuance de gris et en les redimensionnant à la taille $(N_x, N_y) = (60, 120)$, nous utilisons les L premières sorties de filtres de convolution d'un CNN classique (ici MobileNetV2) dans le but d'obtenir un tenseur $\mathbf{T}$ de taille (N_x, N_y, d) (d est le nombre total de filtre en sortie des L premières couches). Classiquement comme dans [7, 8], des estimateurs classiques de matrices de covariances [12] sont utilisés pour effectuer cette étape. Dans le but d'avoir une étape de backpropagation, nous proposons d'utiliser à la place de ces estimateurs, l'approche proposée dans [9,10], appelée MPN-COV. La matrice de covariance obtenue peut alors être de très grande dimension et il semble être intéressant de réduire l'espace des données pour obtenir de meilleures performances. On propose donc de rajouter des couches de convolutions adaptées aux matrices de covariances en utilisant le framework proposé dans [11], dénoté SPDNet. Nous donnons dans la suite des détails sur ces deux pipelines.

MPN-COV [9, 10]

MPN-COV est un modèle qui calcule dans un premier temps une estimation de la covariance sur les données centrées de notre tenseur $\mathbf{T}$ (cette couche est appelée CovPool). Ensuite une étape de régularisation à partir des valeurs propres est rajoutée. L'algorithme de newton-schulz est mis en oeuvre au lieu de la SVD pour obtenir une approximation moins coutêuse en temps de calcul. On dénomme cette approche Fast MPN-COV. Les détails de la backpropagation sont donnés dans les papiers [9, 10]. Les calculs sont basés sur les dérivations de [13].

SPDNet [11]

SPDNet est un modèle qui pour $k \geq 1$ prend en entrée une matrice $\boldsymbol{X}_{k-1} \in Sym^{+}_{d_{k-1}}$ symmétrique définie postive de taille d_{k-1} (variété SPD pour Symmetric Positive Definite). On va réduire la taille de l'espace par plusieurs couches de convolution BiMap et une couche de régularisation ReEig basée sur une EVD. Ces 2 étapes peuvent être répétées plusieurs fois. Finalement nous avons une couche de LogEig pour faire des mesures entre les matrices de covariance dans un espace commun (plan tangent à la variété riemannienne des matrices SPD pris à l'identité). Les inconnues du problème sont les matrices de convolution $\boldsymbol{W}_k \in \mathbb{R}_*^{d_k \times d_{k-1}}$ qui sont rang faible et appartiennent à une variété de Stiefel. Voici quelques détails sur les différentes couches. Tout d'abord on a la couche(pour générer des matrices SPD plus compactes et plus discriminantes), en partant de $\mathbf{X_0} = \mathbf{X}$ et pour $k \geq 1$:

$$\mathbf{X}_k = f_b^{(k)}(\mathbf{X}_{k-1}; \mathbf{W}_k) = \mathbf{W}_k \mathbf{X}_{k-1} \mathbf{W}_k^T$$

où $\boldsymbol{X}_{k-1}$ est alors la matrice SPD d'entrée de la k-ième couche, $\boldsymbol{W}_k \in \mathbb{R}_*^{d_k \times d_{k-1}}, (d_k < d_{k-1})$ est la matrice de transformation (poids des connexions), $\boldsymbol{X}_k \in \mathbb{R}^{d_k \times d_k}$ est la matrice résultante. Nous avons ensuite la couche ReEig Layer (pour améliorer les performances discriminatives et qui s'inspire de ReLU). Nous calculons d'abord l'EVD (Décomposition en valeur propres) de $\mathbf{X}_{k-1} = \mathbf{U}_{k-1} \boldsymbol{\Sigma}_{k-1} \mathbf{U}_{k-1}^T$, et ensuite

$$\mathbf{X}_k = f^r(\mathbf{X}_{k-1}) = \mathbf{U}_{k-1} \max(\epsilon \mathbf{I}, \boldsymbol{\Sigma}_{k-1}) \mathbf{U}_{k-1}^T,$$

$\max(\epsilon \mathbf{I}, \boldsymbol{\Sigma}_{k-1})$ est une matrice diagonale dont les termes seront les valeurs propres corrigées pour rester dans la

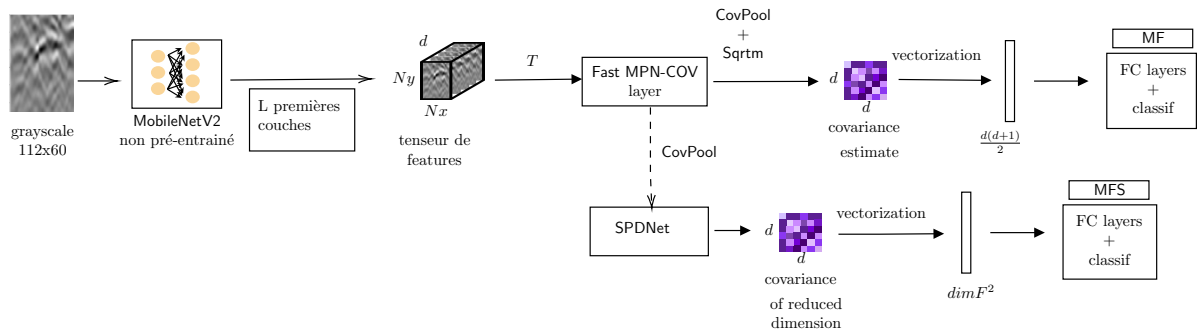

FIG. 3: Chaine de traitement pour la classification. Le Modèle MF considère MobileNetV2 suivi des couches de Fast MPN-COV tandis que le Modèle MFS considère MobileNetV2 suivi d'une couche de Fast MPN-COV et de couches de SPDNet.

variété SPD. Puis après avoir éventuellement répété les couches BiMap et ReEig, nous utilisons la couche LogEig Layer qui est aussi basée sur l'EVD de la matrice $\mathbf{X}_{k-1}$:

$$\mathbf{X}_k = f^l(\mathbf{X}_{k-1}) = \log(\mathbf{X}_{k-1}) = \mathbf{U}_{k-1}\log(\mathbf{\Sigma}_{k-1})\mathbf{U}_{k-1}^T$$

où $\log(\mathbf{\Sigma}_{k-1})$ est la matrice diagonale des logarithmes des valeurs propres. Finalement, des couches *Fully-Connected* et de classification sont ajoutées à la fin ce qui permet d'avoir un framework complet pour la classification multi-classes de nos imagettes d'hyperboles. Les étapes de la backpropagation sont données dans [11, 14].

4 Expériences et analyses

Choix et description du Datasets

Nous disposons de 732 fichiers fournis par la société Géolithe correspondant à des radargrammes de taille moyenne de 4000 x 800 pixels ainsi que toutes leur information d'acquisition comme : la fréquence du radar (200MHz ou 350MHz pour Geolithe), l'élévation du radar (0cm, 25cm, 50cm, 75cm, 100cm, 150cm) et le type de sol (sable humide, sable, grave, grave sèche). Pour chaque radargramme nous avons un masque de même taille associé divisé en plusieurs sous-masques (isolant mieux les hyperboles) qui est labellisé à deux niveaux : la catégorie de chaque hyperbole (objet, réseau discontinuité et vide, un exemple est donné à la figure 4) et la sous-catégorie associée (fusil en bois etc). Cependant au départ nous avons une répartition déséquilibrée de nos catégories avec respectivement 2593 objets, 585 réseaux, 74 discontinuités et 3654 vides. Une nouvelle répartition du jeu de données à donc été construite. Nous le réduisons à quatre nouvelles catégories : abri en bois, objets métalliques, objets non métalliques et vide (aucun objet dans l'imagette). La composition des catégories objets métalliques et objets non métalliques a été faite à partir des sous catégories comme montré sur la Figure 5. Aléatoirement, nous avons construit un jeu de données à partir des 4 élévations: 25, 50, 75 et 100 et des deux fréquences de façon à ce que chaque classe contienne 396 imagettes réparties de façon égale.

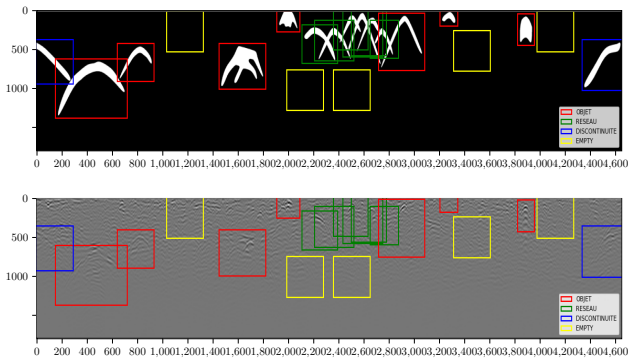

FIG. 4: Image du masque (haut) associée au radargramme (bas). Chaque hyperbole est labellisée. L'ensemble des rectangles est ensuite projeté sur le radargramme.

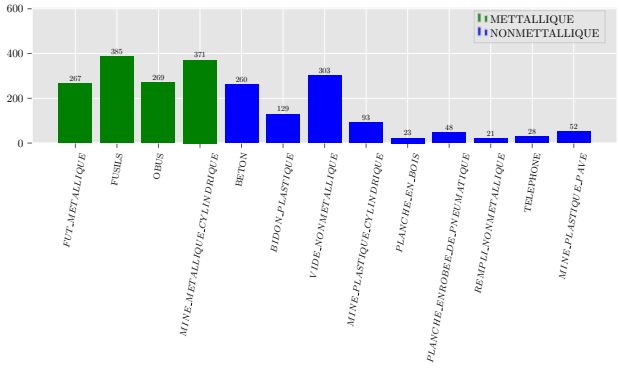

FIG. 5: Repartitions des deux catégories Métallique et Non Métallique en fonction des sous catégories.

Pré-traitement

On explique brièvement le pré-traitement pour isoler chaque hyperbole dans une imagette à partir du masque 4. A partir de cette image, nous utilisons l'algorithme DBSCAN pour regrouper chaque hyperbole de chaque catégorie. En connaissant ainsi le nombre de clusters, le centroïde de chaque hyperbole est identifié à l'aide d'un algorithme du K-Means. Il est ensuite facile de construire un rectangle, délimité par chaque point extrême de ce cluster, autour de l'hyperbole. Dans certains cas, les hyperboles

se chevauchent (par exemple les rectangles verts sur la figure 4) et il est alors nécessaire d'effectuer une érosion pour séparer les hyperboles avant d'appliquer l'algorithme DBSCAN. Finalement nous projetons les positions de chaque rectangle du masque sur le radargramme afin de récupérer nos imagettes à classer qui sont ensuite redimensionnées pour qu'elles soient adaptées aux entrées de nos modèles présentés en Figure 3.

Classification et résultats

Nous avons séparé nos données en trois ensembles de données : données d'entraînement, données de validation et données de test avec des ratios de 80% 20% et 20% respectivement. Sur le principe du *K-folding*, nous interchangeons les ensembles d'entrainement et de validation afin de trouver les meilleurs hyperparamètres en cross-validation. Le meilleur modèle ainsi obtenu est testé sur les données de test. Nous avons testé plusieurs valeurs de L pour la construction du tenseur de données qui servira ensuite dans le pipeline MPN-COV. Les valeurs de L choisies sont 4, 6 et 8 ce qui conduit à des tailles de matrices $d \times d$ avec $d = 128$, 192, 224. Il faut noter que ces valeurs de L sont faibles comparé à l'entièreté du réseau MobileNetV2 qui est un réseau très profond (2 millions de paramètres). Pour SPDNet, nous avons utilisé SPDNet-3BiRe[1] dans lequel les coucheset ReEig se répètent trois fois. Les tailles des matrices de transformation pour $L = 4$ la dimension est de $128102, 10272, 7260$, pour $L = 6$ la dimension est de $192 \times 161, 161 \times 114, 114 \times 95$ et pour $L = 8$ la dimension est de $224 \times 170, 170 \times 120, 120 \times 100$. Les résultats sont affichés dans le Tableau 1. Le modèle MS correspond à l'utilisant de la pipeline Fast MPN-COV seul alors que le modèle MFS est la combinaison de Fast MPN-COV et de SPDNet. On note que ce dernier modèle apporte un gain non négligeable (autour de 4/5%) qui se rapproche du résultat avec MobileNetV2 complet (86 %) mais avec un nombre beaucoup moins important de paramètres (40-50k par rapport à 2M). Les performances s'améliorent avec L. En classification binaire, nous avons noté que le gain de performance était très faible pour $L > 8$ alors que le temps de calcul commençait à être important. Nous ferons cette analyse pour la classification multi-classe dans un futur papier.

Modèle	$L = 4$ $d = 128$	$L = 6$ $d = 192$	$L = 8$ $d = 224$
MF	69%	72%	**75%**
MFS	73%	76%	**80%**

Tab. 1: Performances de classification sur l'ensemble test.

[1] Le développement des modèles a été effectué sous Python 3.9 avec les package pytorch et torchspdnet (`https://gitlab.lip6.fr/schwander/torchspdnet`, [14]).

5 Conclusion

Dans ce papier, nous avons proposé un *framework* de classification d'imagettes d'hyperboles provenant de données GPR, chaque hyperbole représentant un objet enfoui. Le modèle utilise des matrices de covariances calculées en sortie de filtre d'un réseau peu profond. L'estimation est effectuée avec un pipeline comprenant une étape de backpropagation et on utilise le modèle SPDNet [11] pour réduire la taille des données et améliorer les performances de classification.

References

[1] David J Daniels, *Ground Penetrating Radar*, IEE, 2004.

[2] Guillaume Terrasse, Jean-Marie Nicolas, Emmanuel Trouvé, and Émeline Drouet, "Sparse decomposition of the GPR useful signal from hyperbola dictionary," in *EUSIPCO*, 2016, pp. 2400–2404.

[3] Xisto L. Travassos, Sérgio L. Avila, and Nathan Ida, "Artificial Neural Networks and Machine Learning techniques applied to Ground Penetrating Radar: A review," *Applied Computing and Informatics*, vol. 17, no. 2, pp. 296–308, Jan. 2020, Publisher: Emerald Publishing Limited.

[4] Mostafa Elsaadouny, Jan Barowski, and Ilona Rolfes, "ConvNet Transfer Learning for GPR Images Classification," in *2020 German Microwave Conference (GeMiC)*, Mar. 2020, pp. 21–24, ISSN: 2167-8022.

[5] Zhi-Kang Ni, Jun Pan, Cheng Shi, Shengbo Ye, Di Zhao, and Guangyou Fang, "DL-Based Clutter Removal in Migrated GPR Data for Detection of Buried Target," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[6] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten, "Classification of covariance matrices using a riemannian-based kernel for BCI applications," *Neurocomputing*, vol. 112, pp. 172–178, 2013.

[7] Matthieu Gallet, Ammar Mian, Guillaume Ginolhac, and Nickolas Stelzenmuller, "Classification of GPR signals via covariance pooling on CNN features within a riemannian framework," in *IGARSS 2022*. IEEE, 2022, pp. 365–368.

[8] Sara Akodad, Lionel Bombrun, Junshi Xia, Yannick Berthoumieu, and Christian Germain, "Ensemble learning approaches based on covariance pooling of CNN features for high resolution remote sensing scene classification," *Remote Sensing*, vol. 12, no. 20, pp. 3292, 2020.

[9] Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo, "Is second-order information helpful for large-scale visual recognition?," in *Proceedings of the IEEE ICCV*, 2017, pp. 2070–2078.

[10] Peihua Li, Jiangtao Xie, Qilong Wang, and Zilin Gao, "Towards faster training of global covariance pooling networks by iterative matrix square root normalization," in *Proceedings of the IEEE CVPR*, 2018, pp. 947–955.

[11] Zhiwu Huang and Luc Van Gool, "A riemannian network for SPD matrix learning," in *Proceedings of the AAAI conference on artificial intelligence*, 2017, vol. 31.

[12] Olivier Ledoit and Michael Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *Journal of multivariate analysis*, vol. 88, no. 2, pp. 365–411, 2004.

[13] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu, "Training deep networks with structured layers by matrix backpropagation," *arXiv preprint arXiv:1509.07838*, 2015.

[14] Daniel Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, and Matthieu Cord, "Riemannian batch normalization for SPD neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.