Buried Object Classification from GPR Data by using Second Order Deep Learning Models

Douba Jafuno, Ammar Mian, Guillaume Ginolhac, Nickolas Stelzenmuller

▶ **To cite this version:**

Douba Jafuno, Ammar Mian, Guillaume Ginolhac, Nickolas Stelzenmuller. Buried Object Classification from GPR Data by using Second Order Deep Learning Models. International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, Jul 2024, Athenes, Greece. hal-04770269

HAL Id: hal-04770269

https://hal.univ-grenoble-alpes.fr/hal-04770269v1

Submitted on 6 Nov 2024

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.

Buried Object Classification from GPR Data by using Second Order Deep Learning Models

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

[1]Savoie Mont Blanc University, LISTIC, F-74000 Annecy, France
[2]Géolithe, Grenoble, France

ABSTRACT

We propose a new pipeline to classify Ground Penetrating Radar (GPR) images, based upon the use of second-orders statistics computed over convolutional features of a ResNet-like architecture. In particular, the architecture consists in an end-to-end training phase with backpropagation from convolutional filters from layers adapted to Symmetric Positive Definite (SPD) matrices. The developed approach is tested and compared to a shallow network given in the GPR literature and a deep Computer Vision model like ResNet. We show that we outperform these methods when the number of training data is small and when some of them are mislabelled.

Index Terms— GPR, Classification of buried object, covariance matrices, CNN

1. INTRODUCTION

Ground Penetrating Radar (GPR) is a system that provides an image of the ground. The transmitted wave penetrates the ground and is reflected by buried objects. As the GPR moves along an axis, objects are viewed from different positions, and the resulting image of an object looks like hyperbola. The final image, called a B-scan in the GPR community, is noisy and the signal-to-noise ratio for objects is generally low. For this reason, much of the work in this field is devoted to improving the quality of the B-scan and are based on image processing tools [1, 2, 3]. This enhancement step is often necessary to detect and locate the various hyperbolas. In this paper, we assume that these steps are done. The final task is then to classify the various objects detected. The different methods developed for GPR data are based on the shape of the hyperbola. Next deep learning models [4, 5, 6] have been built to classify the objects from this shape of hyperbola. However, these models are shallow like in [4] where only 3 layers are used. In this case, they do not benefit from the rich representation of large Computer-Vision (CV) models such as MobileNetV2, ResNet or Alexnet. On the opposite these latter depend on a large amount of unknown parameters which makes models difficult to train for applications with a small amount of labelled data like in the GPR scenario. We propose in this paper to use the CV models while trying to keep the number of learned parameters smaller.

In fact, in [7], we show that it is possible to estimate a covariance matrix from the outputs of the first layers of a CV model. This covariance matrix measures the correlation between the different filters and appears to be a better feature to classify than the simple hyperbola image. In this previous paper, we used classical machine learning methods (SVM and Minimum Distance Mean) that was not sufficient to achieve correct classification rates in particular when the number classes increase. In this paper, we therefore propose an end-to-end training model using backpropagation from CNN filters to specific layers adapted to the covariance matrices. These layers come from the literature of networks built for Symmetric Positive Definite (SPD) matrices [8, 9, 10, 11].

We test our approach in a real dataset provided by Geolithe and labelled by geophysical experts. These simulations show that we outperform a shallow GPR model [4] and computer-vision models like ResNet when the number of training data is small and when some of them are mislabelled.

2. GPR DATA

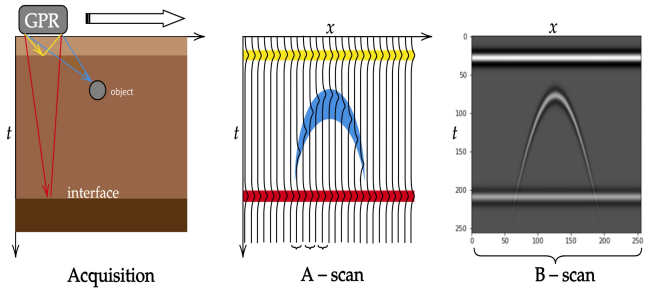
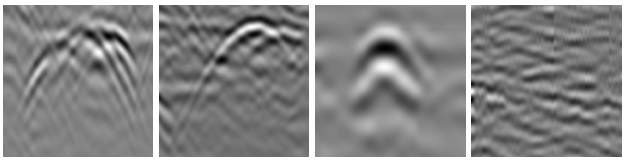


Fig. 1: Illustration of GPR acquisition principle.

The principle of GPR is to move a RADAR consisting of a transmitter and a receiver along a straight line. The emitted signal is a Ricker-type wavelet whose main characteristic is its frequency. Depending on its frequency, the signal can penetrate the ground to a certain depth. The principle is illustrated on the left of Fig. 1. The GPR output is called an

A-scan and consists of the different acquisitions for each position. Most of the time, we prefer to build an image, called a B-scan, from the A-scan image. In this image, the shape of a simple object resembles a hyperbola, thanks to the movement of the GPR and the propagation of the electromagnetic wave. An example of a B-scan image with one object and two layers is shown on the right of Fig. 1.

In this article, we assume that the hyperbolas of objects are detected and localized, and that the main objective is to classify them by studying a thumbnail image of the corresponding hyperbola. Before presenting the proposed approach in the next section, we will introduce the dataset used in this work and some ideas on how the shape signal of the hyperbola informs us about the object present.



(a) (b) (c) (d)

Fig. 2: Examples of GPR images (after histogram correction thanks to GPRpy[1]) for different buried objects. (a) Wooden shelter, (b) Metallic, (c) Non-Metallic, (d) Empty.

Geolithe supplied 732 B-scans with an average size of $(4000, 800)$ pixels. Different parameters for GPR and acquisition were used, such as radar frequency (200MHz or 350MHz for Geolithe), radar elevation (0cm, 25cm, 50cm, 75cm, 100cm, 150cm), and soil type (wet sand, sand, gravel, dry gravel). The different hyperbolas of each B-scan are then detected, located and an adapted pre-processing is used to create a miniature image for each of them. The resulting thumbnails are resized to $(60, 112)$ pixels as in [4]. The final database therefore consists of 1,584 thumbnails divided into four categories: Metal, Non-metal, Wooden Shelter and Empty. An example for each category is shown in Fig. 2. The wooden shelter is a huge structure set into the ground, with an empty interior. This explains why the resulting image is made up of several hyperbolas. We also note that the image of a metallic object is quite different from that of a non-metallic object. The empty class is normally composed entirely of noise.

3. CLASSIFICATION MODEL

The model used in this work is presented in Figure 3, where we show both the type of layers used and their output shapes. The architecture proposed rely on the following components:

ResNet layers: The first layers correspond to ResNet convolutional layers with shortcut connections [12] which have been shown to allow a rich feature representation space across a large area of applications. We choose ResNet-34 to have fewer weights than larger model (we want to keep it efficient). As shown in Figure 3, we decide to only keep the first few L layers since we only keep the feature learning part and not the classification one. The initial image size is taken from [4], to have a baseline of comparison and we sample it to the ResNet image input size by using image interpolation. We use a non trained ResNet model that we train from scratch because the features learned in computer vision models appears to be not sufficient in regards to separating the different class, at least with only the first few layers. In this paper, we use $L = 8$.

[1]see `https://github.com/NSGeophysics/GPRPy`.

	d	d_1	d_2	d_3	d_4
SRCNet	256	235	217	179	128
RCNet	64	58	54	44	32

Table 1: Output dimensions of SPDNet layers.

Covariance pooling layer: Then a covariance pooling layer is added, that allows to reduce the dimensionality of the learned features and while preserving correlation information between them compared to first-order pooling approaches [9]. Two scenarios (RCNet and SRCNet), where we either stack the output of each layers or we only keep the last one. In the SRCNet approach, we benefit from a more diverse set of convolutional features, while in the RCNet we have fewer diversity but a reduced number of overall parameters in the following layers. To do the estimation, we rely on the Fast MPN-Cov framework [10] without the regularization part.

SPDnet layers: After estimating the correlation structure between the features, we want to reduce the dimensionality since maybe not all the information of the covariance is useful for the classification task. To do that, we employ layers form the SPDnet [8] architecture which defines two main layers that allow to work on SPD matrices:

- BiMap Layer (to generate more compact and discriminative SPD matrices):

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

  where $\boldsymbol{X}_{k-1}$ is the input SPD matrix of the k-th layer, $\boldsymbol{W}_k \in \mathbb{R}_*^{d_k \times d_{k-1}}, (d_k < d_{k-1})$ is the orthonormal transformation matrix (connection weights), and $\boldsymbol{X}_k \in \mathbb{R}^{d_k \times d_k}$ is the resulting matrix.

- ReEig Layer (to regularize the output of the BiMap, inspired by ReLU):

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

  where $\max(\epsilon \mathbf{I}, \mathbf{\Sigma}_{k-1})$ is a diagonal matrix with corrected eigenvalues to stay on the SPD manifold.

The dimensions used in this paper are reporting in table 1.

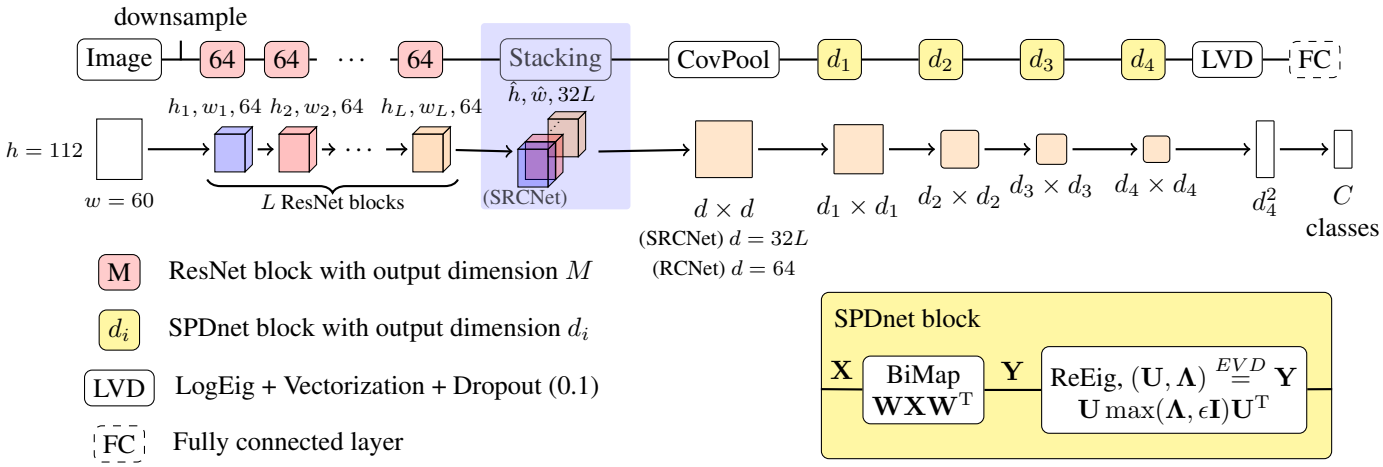


Fig. 3: Illustration of the architectures used in this paper. In *RCNet* (Residual Covariance Network), we take only the last output of the ResNet blocks while in *SRCNet* (Stacked Residual Covariance Network), we stack the first 32 outputs (to save memory space) of the outputs features by interpolating them to a common size of 38×20. With RCNet, we have $h_L = 54$ and $w_L = 30$ for the first 3 layers and $h_L = 28$ and $w_L = 15$ for the others.

Final layers: Finally, we want to classify the lower-dimensional discriminative SPD matrix. To be able to apply traditional Fully Connected (FC) layers, we first need to transform the SPD matrix into a feature in euclidean space, which is done by the LogEig operator which takes the matrix logarithm of the input. Then, we vectorize the obtained matrix and add an additional dropout mechanism to avoid overfitting.

Back-propagation Steps: Since the main steps are matrix operations, the backward is not classical like in most of deep learning models. In particular, the gradients of the operations based on SVD have been firstly derived in [13]. A more stable formula is given in [11] and will be used. The BiMap layer needs the computation of a Gradient Riemannian on the Stiefel manifold and the details are given in [8].

4. NUMERICAL RESULTS

To compare our approach on the dataset previously presented, we consider the model developed in [4] which is a shallow convolutional network with only 3 layers. Since our approach is a constructs on the ResNet architecture, we also compare with the *ResNet-34* model in two different setups: *i)* fine-tuning, in which we take pre-trained weights on the ImageNet dataset as initialization and fine-tune on our dataset, *ii).* re-training, in which we initialize the weights randomly.

We separate the dataset into a training set (1108 images), a validation set (238 images) and a testing set (238 images) which will used for accuracy metrics. We consider two scenarios on the training part: *i).* varying the percentage of training data used, to reflect the low-availability of labelled data in GPR scenarios, *ii).* with a varying level of mislabelled data, to study robustness of the models over likely mislabelling. For both scenarios, we train over 100 different random generator seeds to assess variability of the learning and with the following parameters: a batch-size of 64, a learning rate of 10^{-2} and we stop the training when training accuracy hits a plateau over 10 previous iterations.

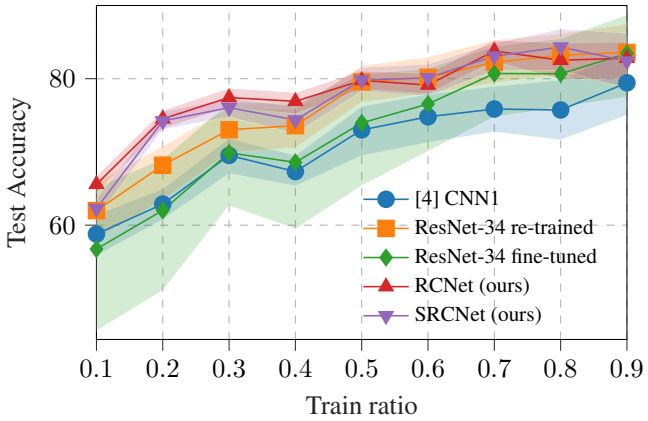


Fig. 4: Results of classification w.r.t to training dataset percentage over 100 different seeds. For each method, the line corresponds to the mean of accuracy over all the seeds, and the filled area corresponds to 5-th and 95-th quantiles.

The results of test accuracy w.r.t training ratio are shown in Figure 4. We can observe the following things: Ours proposed approaches (*SRCNet* and *RCNet*) both outperform the others approaches at any given training ratio and have lower variability over seeds. The *ResNet-34* approaches perform better than [4], which is understable given that they use considerably more parameters. The fine-tuning approach fares better in this case, which can be explained by the fact that the training starts with an already good discriminative capabilities between objects thanks to the pre-training. The results of test accuracy w.r.t mislabelling percentage are presented in Figure 5. We have comparable dynamic than the previous ex-

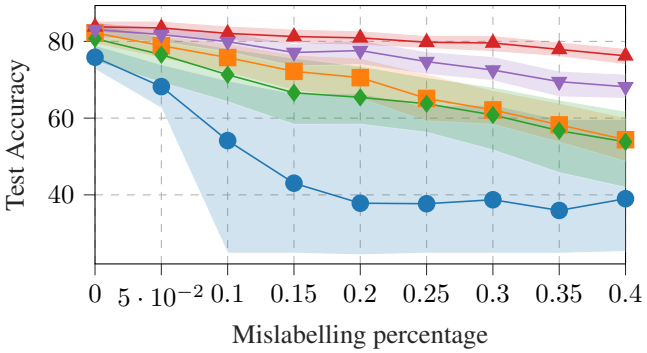


Fig. 5: Results of classification w.r.t training dataset mislabelling percentage. For each method, the line corresponds to the mean of accuracy over all the seeds, and the filled area corresponds to 5-th and 95-th quantiles. The models are plotted with same style as in Fig. 4.

periment with better performance and robustness of *RCNet*. The case of [4] is interesting as it show that shallow models have a lot of variability in the training phase as soon as there are a few mislabelled data. Again our approaches have the least variability over the seeds.

5. CONCLUSION

We have proposed a new architecture for GPR classification based upon covariance pooling and SPDnet architecture that outperforms both deep ResNet-34 models and shallow CNN approaches in terms of accuracy and robustness.

6. REFERENCES

[1] Guillaume Terrasse, Jean-Marie Nicolas, Emmanuel Trouvé, and Émeline Drouet, "Sparse decomposition of the GPR useful signal from hyperbola dictionary," in 2016 24th European Signal Processing Conference (EUSIPCO), 2016, pp. 2400–2404.

[2] Fabio Giovanneschi, Kumar Vijay Mishra, Maria Antonia Gonzalez-Huici, Yonina C. Eldar, and Joachim H. G. Ender, "Dictionary learning for adaptive GPR landmine classification," IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 12, pp. 10036–10055, 2019.

[3] Matthieu Gallet, Ammar Mian, Guillaume Ginolhac, Esa Ollila, and Nickolas Stelzenmuller, "New robust sparse convolutional coding inversion algorithm for ground penetrating radar images," IEEE Transactions on Geoscience and Remote Sensing, 2023.

[4] Maha Almaimani, Dalei Wu, Yu Liang, Li Yang, Dryver Huston, and Tian Xia, "Classifying GPR Images Using Convolutional Neural Networks," in Proceedings of the 11th EAI International Conference on Mobile Multimedia Communications, Brussels, BEL, 2018, MOBIMEDIA'18, p. 68–73, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[5] Haoqiu Zhou, Xuan Feng, Yan Zhang, Enhedelihai Nilot, Minghe Zhang, Zejun Dong, and Jiahui Qi, "Combination of Support Vector Machine and H-Alpha Decomposition for Subsurface Target Classification of GPR," in 2018 17th International Conference on Ground Penetrating Radar (GPR), June 2018, pp. 1–4, ISSN: 2474-3844.

[6] 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.

[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] 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.

[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] 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.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[13] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu, "Matrix backpropagation for deep networks with structured layers," in 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2965–2973.