



**HAL**  
open science

# Deep unrolling of Robust PCA and Convolutional Sparse Coding for stationary target localization in through wall radar imaging

Hugo Brehier, Arnaud Breloy, Chengfang Ren, Guillaume Ginolhac

► **To cite this version:**

Hugo Brehier, Arnaud Breloy, Chengfang Ren, Guillaume Ginolhac. Deep unrolling of Robust PCA and Convolutional Sparse Coding for stationary target localization in through wall radar imaging. EUSIPCO 2024, Aug 2024, Lyon, France. hal-04707971

**HAL Id: hal-04707971**

**<https://hal.science/hal-04707971v1>**

Submitted on 24 Sep 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.

# Deep unrolling of Robust PCA and Convolutional Sparse Coding for stationary target localization in through wall radar imaging

Hugo Brehier\*, Arnaud Breloy†, Chengfang Ren\*, Guillaume Ginolhac‡

\*SONDRA, CentraleSupélec, Université Paris-Saclay

†CEDRIC, Conservatoire National des Arts et Métiers

‡LISTIC, Université Savoie Mont-Blanc

**Abstract**—Through Wall Radar Imaging aims to see through walls using electromagnetic waves. Low rank and sparse decomposition methods have been effective in processing returns in order to distinguish the wall response from the interior scene. However, they rely on model assumptions that can be a poor approximation of the actual physics. In the meantime, data-driven methods based on Deep Learning can provide an improvement regarding to this limitation. We thus propose a new unrolled network inspired by Robust PCA and Convolutional Sparse Coding which proves to be competitive and especially efficient in scarce data regimes.

**Index Terms**—Through Wall Radar Imaging, Localization, RPCA, CSC, Deep Unrolling.

## I. INTRODUCTION

Through Wall Radar Imaging (TWRI) [1] is a field of research aiming at imaging scenes obstructed to the naked eye by walls using electromagnetic waves penetrative properties. It may be used in the context of surveillance or monitoring [2]. We focus here on the detection and localization of stationary targets.

In recent years, methods using the optimization framework of low rank and sparse decomposition methods, i.e. Robust PCA (RPCA), have shown their effectiveness on TWRI localization [3], [4]. However, the signal model remains a simplified approximation of the actual undergoing physics: dispersive walls, anisotropic targets, clutter, etc., are not considered. With this in mind, Deep Learning (DL) methods have recently been successfully used on TWRI [5], [6] by directly learning more complex models from the data. These models require many labeled samples in order to extract a meaningful model which can be limiting in practice.

A possible remedy is to rely on more structured neural network architectures such as unrolled ones, in order to combine the best of both worlds. Unrolled networks are constructed by converting a fixed number of iterations of an optimization algorithm into the layers of a neural network. The first unrolled network was developed in the case of sparse coding via the Iterative Shrinkage Thresholding Algorithm (ISTA) algorithm and coined Learned ISTA (LISTA) [7]. It emphasized the gain in runtime of obtaining the same sparse codes as the underlying optimization algorithm.

Subsequently, other works showed the improved performance of such unrolling methods by departing from the

original optimization algorithm [8]. They benefit from their structure by having fewer parameters, better efficiency w.r.t. data samples and more generalization power than generic deep networks. Already, works have emerged on Convolutional Sparse Coding (CSC) [9], [10] and RPCA [11]–[13].

This paper proposes a new unrolled network sourced from the underlying physical model of TWRI used in RPCA which we mix with CSC [14]. This bridges the gap between generic data-driven DL models and optimization methods. It allows the method to be interpretable and more efficient in using data samples, which is important in TWRI where measurements are scarce.

Section II presents the signal model as well as RPCA and CSC optimization frameworks. Section III presents the proposed deep unrolling method while Section IV studies its performance on simulated measurements.

## II. RPCA AND CSC MODELS FOR TWRI

### A. Frequency model and RPCA methods

A widely used model for 2D TWRI [15] consists of a stepped-frequency radar acquiring an  $M$  frequency signal (usually in  $L$ -band) at  $N$  positions (usually a few dozen) along the wall to penetrate with the displacement axis being parallel to the wall. The returned signal for the  $m^{\text{th}}$  frequency and  $n^{\text{th}}$  position is expressed as follows:

$$y_f(m, n) = \sum_{k=1}^K \sigma_w^{(k)} \exp(-j\omega_m \tau_w^{(k)}) + \sum_{i=1}^R \sum_{p=1}^P \sigma_p^{(i)} \exp(-j\omega_m \tau_{p,n}^{(i)}) \quad (1)$$

where  $P$  denote the number of targets,  $K$  the number of reverberations in the wall and  $R$  the number of multipaths. Moreover,  $\sigma_w^{(k)}$  and  $\tau_w^{(k)}$  stand for the complex overall reflectivity and round-trip delay for the wall returns associated with the  $k^{\text{th}}$  reverberation, while  $\sigma_p^{(i)}$  and  $\tau_{p,n}^{(i)}$  denote those related to the  $p^{\text{th}}$  target,  $i^{\text{th}}$  multipath, and  $n^{\text{th}}$  radar position.

We discretize these returns across a grid with dimensions  $(N_x, N_z)$  that cover the scene yielding a dictionary denoted  $\Psi_f$ . For the  $i^{\text{th}}$  multipath scheme and the  $n^{\text{th}}$  transceiver position, the  $(n_x, n_z, i)^{\text{th}}$  column of this dictionary describes

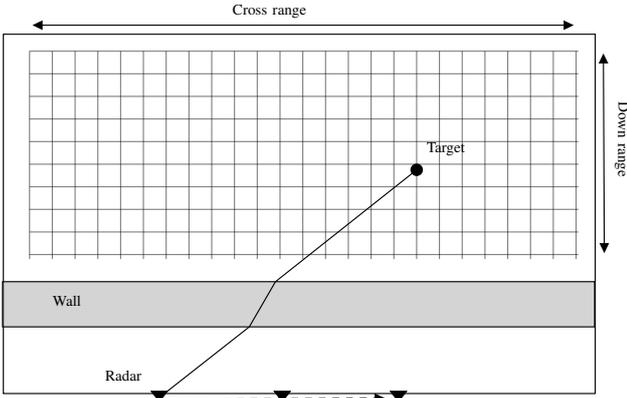


Fig. 1: Scene description

the return from a point target located at the  $(n_x, n_z)^{th}$  pixel via the  $i^{th}$  multipath. Its associated delay  $\tau_{(n_x, n_z), n}^{(i)}$  can in turn be obtained by geometric considerations [15]. For example, for the direct path, we may use Snell's law as the signal undergoes two refractions through a homogeneous wall (see Figure 1) whose angle we can retrieve via some root-finding method. To factorize this into an overall model for all returns, we make the assumption that both the target and wall reflectivities are non-dispersive and anisotropic. This gives the data matrix of radar returns  $\mathbf{Y}_f \in \mathbb{C}^{M \times N}$  as:

$$\mathbf{Y}_f = \mathbf{L}_f + \Psi_f (\mathbf{I}_N \otimes \text{vec}(\mathbf{R}_f)) \quad (2)$$

where  $\otimes$  denotes the Kronecker product. Here:

- $\mathbf{L}_f \in \mathbb{C}^{M \times N}$ , the matrix of front wall returns, which exhibits a low-rank structure since wall returns are invariant along the displacement axis.
- $\mathbf{R}_f \in \mathbb{C}^{N_x N_z \times R}$  containing the amplitude of scene returns across the  $R$  multipaths on a grid of size  $N_x \times N_z$ , associated with the dictionary  $\Psi_f \in \mathbb{C}^{M \times N_x N_z R N}$  which contains the expected returns at those positions of point targets.  $\mathbf{R}_f$  is sparse due to the presence of few targets. This sparsity is structured (by row) since all multipaths represent the same underlying scene and should activate together.

Then TWRI aims to recover  $\mathbf{R}_f$  via the measurements  $\mathbf{Y}_f$ . We may tackle this via the RPCA framework [16]:

$$\begin{aligned} \min_{\mathbf{L}_f, \mathbf{R}_f} \quad & \mu \|\mathbf{L}_f\|_* + \lambda \|\mathbf{R}_f\|_{2,1} \\ \text{s.t.} \quad & \mathbf{Y}_f = \mathbf{L}_f + \Psi_f (\mathbf{I}_N \otimes \text{vec}(\mathbf{R}_f)) \end{aligned} \quad (3)$$

$\|\cdot\|_*$  denotes the nuclear norm, which is the sum of singular values and is the convex envelope of the rank.  $\|\cdot\|_{2,1}$  denotes the  $\ell_{2,1}$  norm, a structured sparse norm [17] across multipath. It may be solved via the Alternating Directions Method of Multipliers (ADMM), as in [16]. This was extended to handle heterogeneous noise in a method coined Huber-Kronecker RPCA (HKRPCA) [4] and to a riemannian framework [18].

### B. Image domain and CSC

In the previous section, we showed a matrix dictionary on the raw frequency signal matrix. Unfortunately, it results

in heavy operations in memory storage as well as computation time and may not work well with dispersive walls and anisotropic scattering. We investigate a convolutional method for lighter computations. This implies working in the *image* domain, where targets returns are localized and can be matched to convolutional kernels. We use the Back-Projection (BP) method [1, Section 3.5.1] in order to transform the raw data into an image. The features of the BP image allow us to extract meaningful and well-grounded representations of the data. We adopt the CSC framework as *the image retains the low rank and sparse properties* of the signal matrix after BP. Our CSC model on the image  $\mathbf{Y} \in \mathbb{R}^{P_x \times P_z}$ , resulting from BP on  $\mathbf{Y}_f$ , may then be written as:

$$\begin{aligned} \min_{\mathbf{L}, \{\mathbf{R}_k\}_{k=1}^K} \quad & \mu \|\mathbf{L}\|_* + \lambda \sum_{k=1}^K \|\mathbf{R}_k\|_1 \\ \text{s.t.} \quad & \mathbf{Y} = \mathbf{L} + \sum_{k=1}^K \Psi_k * \mathbf{R}_k \end{aligned} \quad (4)$$

where  $*$  denotes a convolution. Moreover,  $\{\Psi_k\}_{k=1}^K$  is a collection of  $K$  convolutional filters and  $\{\mathbf{R}_k\}_{k=1}^K$  is a collection of sparse activation maps (the same size as the input BP image  $\mathbf{Y}$ ) that we aim to retrieve. Along with  $\mathbf{L}$ , they decompose the BP image in low rank and sparse components.

In our case, we do not know the filters, so we opt to learn them. We may achieve this via a kind of convolutional neural network (CNN). There is in fact a close connection between CNN and CSC, as evidenced in [19]. Taking this into account, we will link the aforementioned optimization method to a deep learning one via the framework known as deep *unrolling* [8], which unrolls a fixed number of iterations of an iterative optimization algorithm and transpose them to layer of a network.

## III. DEEP UNROLLING METHODS

### A. Source algorithm: a composite PGD for CSC/RPCA

We thus need an iterative optimization method to unroll into a network. We may rewrite the CSC model (4) as a classical sparse coding problem in Lagrangian form via vectorization and the use of a concatenation of Toeplitz matrices denoted  $\Psi_c$  which contains shifted replicas of the filters  $\{\Psi_k\}_{k=1}^K$ . Adding a low rank component  $\mathbf{L}$ , this gives a mix between CSC and RPCA:

$$\min_{\mathbf{L}, \mathbf{R}_c} \frac{1}{2} \|\text{vec}(\mathbf{Y} - \mathbf{L}) - \Psi_c \text{vec}(\mathbf{R}_c)\|_2^2 + \lambda \|\mathbf{R}_c\|_1 + \mu \|\mathbf{L}\|_* \quad (5)$$

with  $\mathbf{R}_c \triangleq [\mathbf{R}_1, \dots, \mathbf{R}_K]$  the concatenation of the  $K$  sparse activation maps. This optimization program may be compacted using a composite variable [11]:

$$\min_{\mathbf{z}} f(\mathbf{z}) + \frac{1}{2} \|\text{vec}(\mathbf{Y}) - \mathbf{K}\mathbf{z}\|_2^2 \quad (6)$$

where  $\mathbf{z} = [\text{vec}(\mathbf{L})^T, \text{vec}(\mathbf{R}_c)^T]^T$ ,  $\mathbf{K} = [\mathbf{I}, \Psi_c]$  and  $f(\mathbf{z}) = \mu \|\mathbf{L}\|_* + \lambda \|\mathbf{R}_c\|_1$  whose proximal [20] is separable in its components and thus computable as  $\text{prox}_{\|\cdot\|_*}(\mathbf{L})^T$ ,  $\text{prox}_{\|\cdot\|_1}(\mathbf{R}_c)^T$ . The gradient of the differentiable part is

readily known as  $\mathbf{K}^H(\text{vec}(\mathbf{Y}) - \mathbf{K}\mathbf{z})$ . We may then apply a step of Proximal Gradient Descent (PGD) [20], re-separate the components of  $\mathbf{z}$  and return to the original convolutional form, giving at iteration  $(n + 1)$ :

$$\begin{aligned}\mathbf{L}^{(n+1)} &= \text{prox}_{\mu\|\cdot\|_*}(\mathbf{L}^{(n)} - \alpha(\mathbf{Y} - \mathbf{L}^{(n)} - \sum_k \Psi_k * \mathbf{R}_k^{(n)})) \\ \mathbf{R}_c^{(n+1)} &= \text{prox}_{\lambda\|\cdot\|_1}(\mathbf{R}_c^{(n)} - \alpha(\{\Psi_k\} * (\mathbf{Y} - \mathbf{L}^{(n)} - \sum_k \Psi_k * \mathbf{R}_k^{(n)})))\end{aligned}\quad (7)$$

where  $\alpha$  is some step-size and we define  $\{\Psi_k\} * \mathbf{X} \triangleq [\text{flip}(\Psi_1) * \mathbf{X}, \dots, \text{flip}(\Psi_K) * \mathbf{X}]$  where flip reverses entries along both dimensions [9].

### B. Proposed network: LCRPCA

We are ready to propose an unrolling of the optimization scheme outlined in (7) which we will call Learned Convolutional Robust PCA (LCRPCA). This is similar in spirit to Corona [11] which considers unrolling RPCA algorithms. However, we include a genuine dictionary (with several kernels/filters per dictionary) and stay closer to the source algorithm (by not decoupling the dictionaries in the  $\mathbf{L}$  and  $\mathbf{R}$  steps). The  $(n + 1)$ <sup>th</sup> layer of our proposed method is:

$$\begin{aligned}\mathbf{L}^{(n+1)} &= \text{prox}_{\mu\|\cdot\|_*}(\mathbf{L}^{(n)} + w_0(\mathbf{Y} - \mathbf{L}^{(n)} - \sum_k \mathbf{W}_{1,k}^{(n)} * \mathbf{R}_k^{(n)})) \\ \mathbf{R}_c^{(n+1)} &= \text{prox}_{\lambda\|\cdot\|_1}(\mathbf{R}_c^{(n)} + \{\mathbf{W}_{2,k}^{(n)}\} * (\mathbf{Y} - \mathbf{L}^{(n)} - \sum_k \mathbf{W}_{1,k}^{(n)} * \mathbf{R}_k^{(n)}))\end{aligned}\quad (8)$$

where we define  $\{\mathbf{W}_{2,k}\} * \mathbf{X} \triangleq [\mathbf{W}_{2,1} * \mathbf{X}, \dots, \mathbf{W}_{2,K} * \mathbf{X}]$ . The filters  $\{\mathbf{W}_{2,k}\}$  are introduced to allow for some more flexible encoding-decoding structure than Eq. (7) suggests. This is summed up graphically in Figure 2.

At the end of the network, the image  $\mathbf{Y}$  is reconstructed and the detection map  $\mathbf{R}_d$  is obtained via the overall sparse component on which we apply the sigmoid operator, denoted  $\sigma$ , to get a (soft) binary map. This allows to capture precisely the shape of the targets via the pattern inscribed in the filters:

$$\mathbf{Y} = \mathbf{L} + \sum_k \mathbf{W}_{1,k} * \mathbf{R}_k, \quad \mathbf{R}_d = \sigma\left(\sum_k \mathbf{W}_{1,k} * \mathbf{R}_k\right) \quad (9)$$

The network will learn on training data the set of filters  $\{\mathbf{W}_{1,k}\}$  and  $\{\mathbf{W}_{2,k}\}$  as well as the scalars  $w_0$  and  $\mu, \lambda$ .

As for all RPCA methods, we aim at finding components  $\mathbf{L}$  and  $\mathbf{R}$  that faithfully reconstruct the data. We do not consider ground truths for  $\mathbf{L}$  which would necessitate the empty scene. Additionally, a *crucial* point is that we have access to the ground truth of the scene on the train data, which we leverage during training to enhance the detection map of the network outputs.

Thus, the loss  $\mathcal{L}$  used during training is a weighted sum of two components: a reconstruction loss in the form of an Euclidean distance and a detection loss as seen in [5] in the form of the Cross-Entropy plus the Dice Coefficient (also known as F1-score). It is used to cope with class

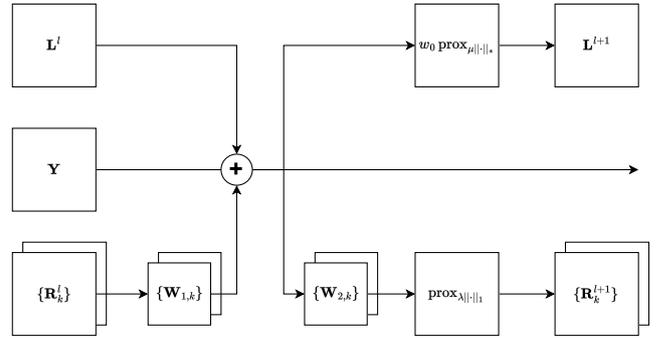


Fig. 2: LCRPCA architecture at  $l^{\text{th}}$  layer

imbalance as the target class contains much fewer pixels than the background one:

$$\mathcal{L}(\hat{\mathbf{Y}}, \hat{\mathbf{R}}_d) = \frac{\|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2 + \text{CE}(\hat{\mathbf{R}}_d, \mathbf{R}_d)}{P_x P_z} + (1 - \text{DC}(\hat{\mathbf{R}}_d, \mathbf{R}_d)) \quad (10)$$

where CE is the cross-entropy loss and DC is the Dice coefficient. Here,  $(\hat{\mathbf{Y}}, \hat{\mathbf{R}}_d)$  is the network output while  $(\mathbf{Y}, \mathbf{R}_d)$  is the reference.

## IV. SIMULATION STUDY

### A. Setting

The dataset for training is collected via GprMax [21] simulations. We consider metallic cylinders varying in radius (5 – 10 cm) in number (1 – 3 targets in one scene) and in position. We generate 330 different scenes with the same dispersive wall via the multi-pole Debye model [22]. Indeed, dispersivity [1, Section 2.2] is a limitation of the model in Section II which assumes that the target and walls are non-dispersive [15, Section 3.1]. The data-driven structure of deep neural networks may be able to alleviate this phenomenon.

Each scene is then added with 10 different draws of a heterogeneous student-t noise (10 - 30 dB and 2.5-5 d.f.). We then have a dataset of 3300 noised returns. The Train/Validation/Test dataset sizes are respectively 2400,800 and 100. We use a learning rate of 0.001 for 30 epochs and the Adam optimizer. Finally, we initialize (i.e. we feed the first layer of the network) with  $\hat{\mathbf{L}} = \mathbf{Y}$  and  $\hat{\mathbf{R}}_d = \{\mathbf{0}\}^{64}$ .

In the following sections, our method LCRPCA is used with  $K = 64$  filters, 6 layers of size  $2 \times (7 \times 7)$ ,  $2 \times (5 \times 5)$ ,  $2 \times (3 \times 3)$  which implies that filters are not shared across layers. The maximum rank of the low-rank component is fixed to 5 in order to use randomized SVD, which stabilizes the gradient when backpropagating the loss through the proximal of the nuclear norm. Indeed, it avoids zero or repeated singular values which make the gradient non-finite. For comparison, we use HKRPCA and Corona which we adapted to our 2D setup as well as the method of [5] that relies on a U-Net with attention.

### B. Visualization

We show the imaging results where target positions are highlighted with red circles. In Figure 3, we show on the

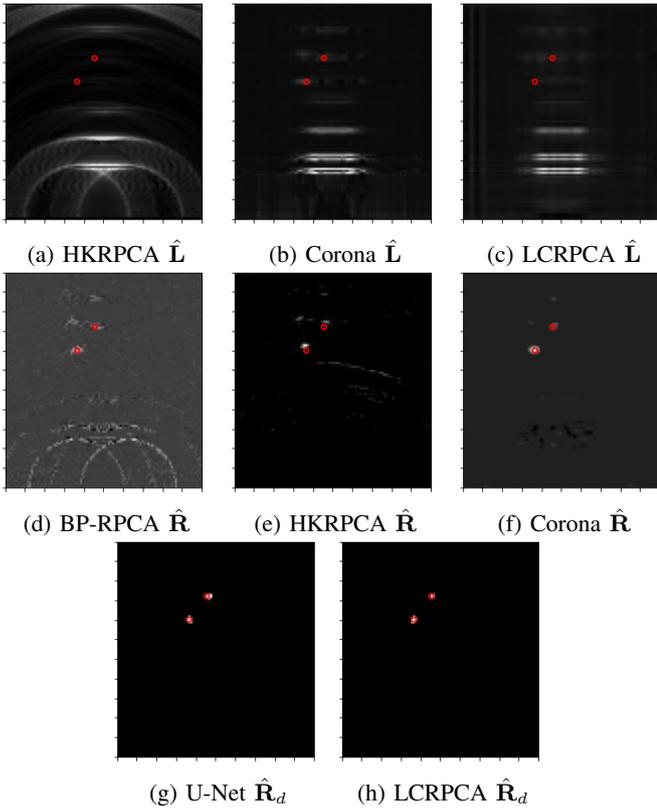


Fig. 3: Sample results. First row: low rank components. Second row: sparse components. Third row: detection maps

first row the low rank components  $\mathbf{L}$  which are supposed to capture wall returns (the U-Net model does not have a low rank component and is thus not shown). We see that the one of HKRPCA (3a) is more complex, with some sprawl on the side. The one of Corona (3b) has more ghosts while the one of LCRPCA (3c) is cleaner. We then have sparse components on the second row, which are supposed to capture target returns. The one resulting of RPCA on the BP image, which is the input of the DL methods, is in Figure 3d. The one of HKRPCA may be seen in Figure 3e, where we see some ghosts behind the true targets. The dispersive wall also has the effect of defocusing the detection. The sparse component of Corona (3f) is cleaner but retains a small ghost. Finally, in the last row (Figures 3g and 3h) we have the detection maps of LCRPCA and U-Net which are quite similar.

Then, we may look at some components of the network with the scalars learned in Figure 4. We see how allowing them to vary can bring adaptivity.

### C. Performance comparison

We move on to the quantitative evaluation with Receiver Operating Characteristic (ROC) and Precision-Recall curves. The legends are enhanced with the Area Under the Curve (AUC) of the ROC and the Average Precision (AP). We see the same ordering of the methods in the two graphs 5 and 6, with the U-Net followed by LCRPCA then Corona and RPCA

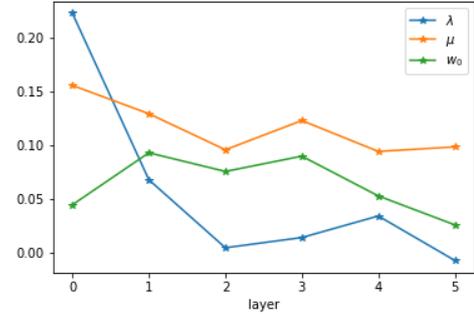


Fig. 4:  $\lambda, \mu, w_0$  across layers

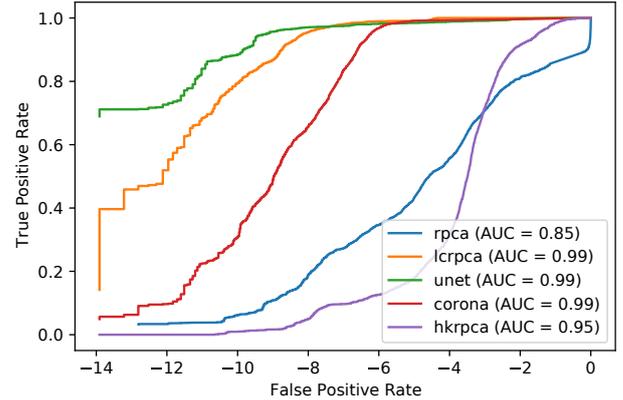


Fig. 5: ROC (FPR in log scale) with full training data

methods last. We then compute the same metrics but with 10% of the training data in Figures 7 and 8. The U-Net performance backs down to the level of LCRPCA and even falls off at the tail end of the curves. LCRPCA is better both in AUC and AP. This highlights the better efficiency of our method under a restricted training regime which is an interesting property in our application where data is scarce.

We also show in Table I the Target to Clutter Ratio (TCR) of the methods. On the first row, we see when the DL methods are trained on 100% of the data where we see the same

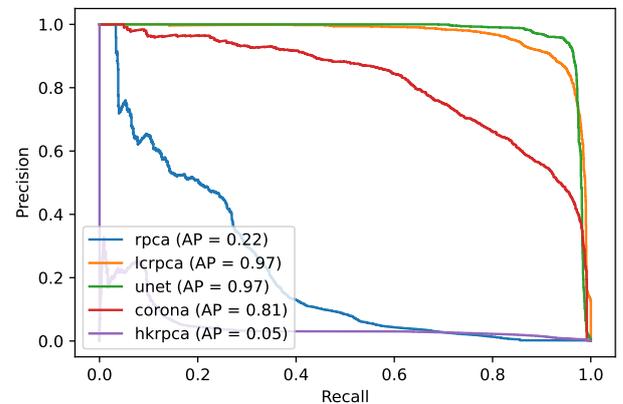


Fig. 6: Precision-Recall with full training data

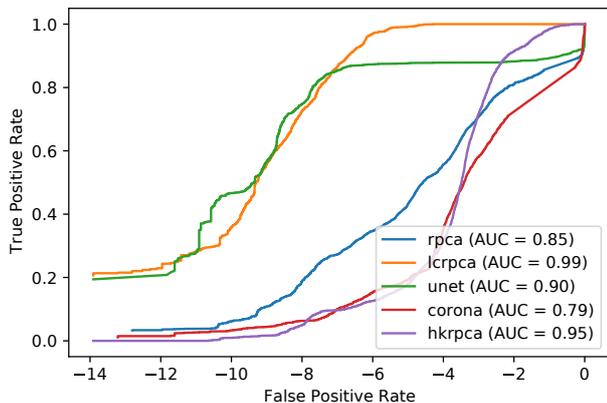


Fig. 7: ROC (FPR in log scale) with scarce training data

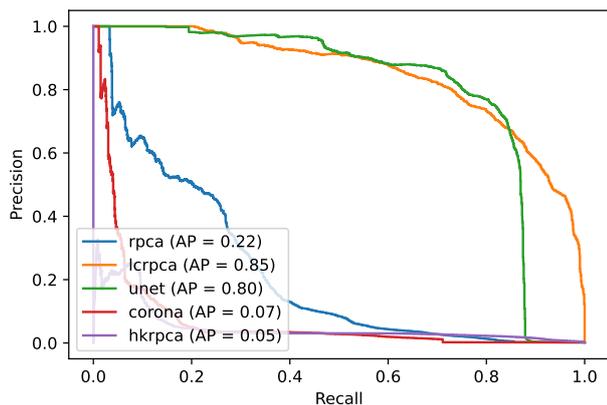


Fig. 8: Precision-Recall with scarce training data

TCR (dB)	HKR-PCA	BP-RPCA	Co-rona	U-Net	LCR-PCA
full training data	18.46	19.52	32.01	41.40	39.93
scarce training data	18.46	19.52	12.60	33.78	33.64

TABLE I: TCR with different trainings

ordering. The next row is with 10% of the data and shows how LCRPCA and the U-Net get closer. LCRPCA has lost 6dB in TCR while the U-Net has lost 8dB. Corona, the other DL method generating a low-rank component, totally collapses with a 20dB loss in TCR. Note that the U-Net has 8,650,474 trainable parameters while LCRPCA has 21,656.

## V. CONCLUSION

This paper presents a deep unrolling of RPCA mixed with CSC for TWRI localization of targets. It showed competitive performance, especially under restricted training data. There remain open questions about handling several unknown walls as well as cluttered environments, which we may hope to tackle by extending this method.

## REFERENCES

- [1] M. G. Amin, *Through-the-Wall Radar Imaging*. CRC Press, 2017.
- [2] K. Mu, T. H. Luan, L. Zhu, L. X. Cai, and L. Gao, "A survey of handy see-through wall technology," *IEEE Access*, vol. 8, pp. 82 951–82 971, 2020.
- [3] V. H. Tang, A. Bouzerdoum, and S. L. Phung, "Compressive radar imaging of stationary indoor targets with low-rank plus jointly sparse and total variation regularizations," *IEEE Transactions on Image Processing*, vol. 29, pp. 4598–4613, 2020.
- [4] H. Brehier, A. Breloy, C. Ren, and G. Ginolhac, "Through the wall radar imaging via Kronecker-structured Huber-type RPCA," *Signal Processing*, p. 109228, 2023.
- [5] H. Li, G. Cui, S. Guo, L. Kong, and X. Yang, "Human target detection based on FCN for through-the-wall radar imaging," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 9, pp. 1565–1569, 2021.
- [6] L. Qu, C. Wang, T. Yang, L. Zhang, and Y. Sun, "Enhanced through-the-wall radar imaging based on deep layer aggregation," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.
- [7] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *International Conference on Machine Learning*, 2010.
- [8] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.
- [9] H. Sreter and R. Giryes, "Learned convolutional sparse coding," in *2018 ICASSP*, 2018, pp. 2191–2195.
- [10] J. Xu, X. Deng, and M. Xu, "Revisiting convolutional sparse coding for image denoising: From a multi-scale perspective," *IEEE Signal Processing Letters*, vol. 29, pp. 1202–1206, 2022.
- [11] O. Solomon, R. Cohen, Y. Zhang, Y. Yang, Q. He, J. Luo, R. J. G. van Sloun, and Y. C. Eldar, "Deep unfolded robust PCA with application to clutter suppression in ultrasound," *IEEE Transactions on Medical Imaging*, vol. 39, no. 4, pp. 1051–1063, 2020.
- [12] H. Cai, J. Liu, and W. Yin, "Learned Robust PCA: A scalable deep unfolding approach for high-dimensional outlier detection," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 16 977–16 989.
- [13] E. Z. C. Tan, C. Chau, E. Soubies, and V. Y. F. Tan, "Deep Unrolling for Nonconvex Robust Principal Component Analysis," in *IEEE MLSP 2023*, 2023.
- [14] M. Gallet, A. Mian, G. Ginolhac, E. Ollila, and N. Stelzenmuller, "New Robust Sparse Convolutional Coding inversion algorithm for Ground Penetrating Radar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–14, 2023.
- [15] M. G. Amin and F. Ahmad, "Compressive sensing for through-the-wall radar imaging," *Journal of Electronic Imaging*, vol. 22, no. 3, pp. 1 – 22, 2013.
- [16] H. Brehier, A. Breloy, C. Ren, I. Hinojosa, and G. Ginolhac, "Robust PCA for Through-the-Wall Radar Imaging," in *2022 30th EUSIPCO*, 2022, pp. 2246–2250.
- [17] M. Kowalski, "Sparse regression using mixed norms," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 303–324, 2009.
- [18] H. Brehier, A. Breloy, C. Ren, and G. Ginolhac, "Through-the-wall radar imaging with wall clutter removal via riemannian optimization on the fixed-rank manifold," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 8596–8600.
- [19] V. Papyan, Y. Romano, and M. Elad, "Convolutional neural networks analyzed via convolutional sparse coding," *Journal of Machine Learning Research*, vol. 18, 07 2016.
- [20] H. Raguét, J. Fadili, and G. Peyré, "A generalized forward-backward splitting," *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1199–1226, 2013.
- [21] C. Warren, A. Giannopoulos, and I. Giannakis, "gprMax: Open source software to simulate electromagnetic wave propagation for ground penetrating radar," *Computer Physics Communications*, vol. 209, pp. 163–170, 2016.
- [22] S. S. Zhekov, O. Franek, and G. F. Pedersen, "Dielectric properties of common building materials for ultrawideband propagation studies," *IEEE Antennas and Propagation Magazine*, vol. 62, no. 1, pp. 72–81, 2020.