RPCANet: Deep Unfolding RPCA Based Infrared Small Target Detection

Fengyi Wu\textsuperscript{1,*}, Tianfang Zhang\textsuperscript{1,*}, Lei Li\textsuperscript{2}, Yian Huang\textsuperscript{1}, Zhenming Peng\textsuperscript{1†}

\textsuperscript{1}University of Electronic Science and Technology of China, Chengdu, China
\textsuperscript{2}University of Copenhagen, Denmark

fengyiwu@std.uestc.edu.cn; sparkcarleton@gmail.com; lilei@di.ku.dk
huangyian@std.uestc.edu.cn; zmpeng@uestc.edu.cn

Abstract

Deep learning (DL) networks have achieved remarkable performance in infrared small target detection (ISTD). However, these structures exhibit a deficiency in interpretability and are widely regarded as black boxes, as they disregard domain knowledge in ISTD. To alleviate this issue, this work proposes an interpretable deep network for detecting infrared dim targets, dubbed RPCANet. Specifically, our approach formulates the ISTD task as sparse target extraction, low-rank background estimation, and image reconstruction in a relaxed Robust Principle Component Analysis (RPCA) model. By unfolding the iterative optimization updating steps into a deep-learning framework, time-consuming and complex matrix calculations are replaced by theory-guided neural networks. RPCANet detects targets with clear interpretability and preserves the intrinsic image feature, instead of directly transforming the detection task into a matrix decomposition problem. Extensive experiments substantiate the effectiveness of our deep unfolding framework and demonstrate its trustworthy results, surpassing baseline methods in both qualitative and quantitative evaluations. Our source code is available at https://github.com/fengyiwu98/RPCANet.

1. Introduction

Research on infrared small target detection (ISTD) plays a crucial role in both civilian and military applications, such as maritime rescue [23], early-warning systems [6], and reconnaissance activities [51]. However, targets usually appear as point targets in just a few pixels due to the inherent extensive imaging distance in infrared detection, leading to limited shape and texture information. Further, background clutter and imaging system noise interfere with distinguishing infrared targets from complex noise, making ISTD a challenging academic topic.

In recent years, ISTD techniques have been advanced, falling into model-driven and data-driven categories. The model-driven framework includes filter-based [2,9,14], human vision system (HVS)-based [5,17,36], and optimization-based methods [6,10,35]. These methods provide theoretical and physical foundations and interpretable detection results, but their efficacy often hinges on finely tuned parameters, which may not suffice when the environment changes.

Data-driven methods for detecting small infrared targets have been enabled by the evolution of Artificial Neural Networks (ANNs). However, general deep-learning-based object detection techniques [11,20,25] have shown limited...
performance on ISTD due to the significant difference in shape and texture information between visible light and infrared. Therefore, recent studies have formulated ISTD as a segmentation problem [7, 8, 19, 47], which can better capture the discriminative and salient features of small targets. However, common segmentation ISTD methods [7, 8] that adopt ImageNet [27] backbones may limit the infrared data representation [38]. Moreover, the multiple downsampling operations in mainstream networks can cause target loss due to the small target size. The reliability of these networks is questionable as they are designed without ISTD domain knowledge.

Taking into account the benefits and drawbacks of both model and data-driven ISTD techniques, a pertinent question arises: Can we devise a balanced ISTD method that integrates the strengths of data-driven and model-driven methods to yield more reliable ISTD results? Deep unfolding network (DUN), also known as algorithm unrolling, is an emerging technique that bridges the gap between iterative algorithms and neural networks. This technique has attracted a great deal of attention across multiple fields [26, 29, 33, 42]. DUN constructs a network by unrolling the iterative solving algorithm of an existing model at the iteration level. Its hyperparameters are subsequently updated in a network-based fashion. By establishing systematic and precise connections between iterative algorithms and neural networks, DUNs exhibit considerable potential for building more interpretable networks [1].

Interestingly, optimization-based ISTD schemes, as an extended application of image completion, involve optimization steps such as model formulation (as shown in Fig. 1 (a), low-rank background and sparse target) and iterative solving methods [10, 37], which align with the requirement of DUNs. However, complex matrix operations hinder scholars from delving into DUN-based ISTD. Some studies in medical image restoration [16] and video separation [4] have modeled images using robust principle component analysis (RPCA). But directly learning parameters in singular value thresholding (SVT) and soft thresholding (ST) overlooks the inherent correlation of images and complicates the empirical selection of regularization parameters.

Thus, to design an ISTD network that addresses the above issues and balances efficiency and interpretability, we develop a deep architecture called Robust Principle Component Analysis Network (RPCANet), as shown in Fig. 1 (d). Rather than applying ST on deep features, this framework uses neural layers to approximate the sparsity constraint function within a target extraction module (TEM). In addition, we design a background extraction module (BEM) to mimic the proximal function with convolution layers, thus approximating the background and eliminating steps such as SVT. An image reconstruction module (IRM) is also introduced to iteratively combine the target and background outputs. In summary, our work contributes to the field of ISTD by:

- RPCANet is a learnable deep network architecture derived from a relaxed RPCA model in Fig. 1 (b) and (c), combining the accuracy of data-driven networks with the interpretability of model-driven ISTD approaches.
- Our TEM and BEM approximate the target and background for segmentation. Nonlinear proximal mapping problems for targets are handled effectively, while complex matrix calculations in background estimation are replaced with neural layers. Moreover, the proposed IRM combines the background and targets, aiming to reconstruct the image.
- Experimental results on multiple datasets demonstrate the effectiveness of RPCANet compared to state-of-the-art baselines. Visualization of learned characteristics provides comprehensive and reliable results.

2. Related Work

2.1. Optimization-based ISTD

Numerous ISTD methods have been developed over the past decades. Optimization-based methods are dominant in the model-driven category, compared to traditional filter-based [2, 9, 14] and HVS-based [5, 17, 36] schemes. The infrared patch image (IPI) model by Gao et al. [10] introduces the method of low-rank and sparse decomposition within the stable RPCA model [52]. This matrix-based category has been enriched by works in single [44, 49] and multi [35] subspaces. Based on the hypothesis that tensors leverage correlation information better than matrices [12], Dai et al. proposed a reweighted infrared patch tensor (RIPT) model for ISTD [6]. Various strategies [18, 37, 45] based on tensors have been applied to both single and multi-frame infrared sequences. The optimization-based methods model targets and backgrounds in a mathematically interpretable manner within RPCA. However, they have limitations in robustness, parameter tuning, and efficiency [46, 48]. Therefore, our goal is to develop a model that preserves interpretability while improving robustness and efficiency.

2.2. Deep Learning-based ISTD

In contrast to model-driven ones, convolutional neural networks (CNNs) are data-driven to learn the non-linear mappings between the original images and masks and are flexible to complicated scenarios. Wang et al. proposed a conditional GAN (MDvsFA-cGAN) [34] to reduce false alarms and missed detections in ISTD. Dai et al. presented an asymmetric contextual modulation (ACM) method [7] to enhance feature fusion. Zhang et al. [47] extracted contextual information of the target in the deep layer, based
on ACM. Li et al. [19] designed the DNA-Net with an enhanced receptive field to prevent small targets from disappearing. Wu et al. [38] integrated residual u-blocks into the network, which preserves feature resolution. Zhang et al. [46] merged an edge block in a U-Net framework to enhance edge features, considering target shapes vary. Ying et al. [40] updated labels with single-point supervision, based on effective models [7, 8, 19]. However, most data-driven methods tailor the segmentation network, which is a black box, instead of incorporating classic ISTD algorithms with field knowledge. Only a few studies have tried this approach [8, 15]. Therefore, we aim to find a solution that balances interpretability and effectiveness.

2.3. Deep Unfolding Networks

Deep unfolding networks (DUNs) solve iterative optimization problems via neural networks and have been widely used in image processing. Gregor and LeCun [13] first proposed the Learned IST A (LIST A) model, which learns the parameters of the iterative shrinkage thresholding algorithm (ISTA). Zhang et al. [41, 42] improved it by introducing IST A-Nets, which incorporate neural layers into the update steps. Yang et al. [39] developed ADMM-CSNet, which is based on unfolding the alternating direction multiplier method (ADMM) for MRI-oriented image reconstruction. Borgerding et al. [3] applied the approximate message passing algorithm (AMP) to learn sparse linear inverse problems, and Zhang et al. [50] extended it to compressive sensing. Other works also use DUNs for optimization models, such as low-rank representation [48] and RPCA [4, 28]. These DUNs frameworks blend the strengths of model and data-driven methods, achieving high and robust performance. Given the solid knowledge of optimization and deep learning-based methods, a progression towards DUN-based ISTD is a logical next step.

3. Deep Unfolding RPCA Network

3.1. Problem Formulation

For an infrared image \( D \), the physical model separates it into low-rank background \( B \) and sparse target \( T \) as:

\[
    D = B + T ,
\]

where \( D, B, T \subseteq \mathbb{R}^{m \times n} \). In a general RPCA [52] manner, we aim to recover the low-rank background \( B \) that could pair the given \( D \) by constraining sparse target \( T \) and usually transform the detection problem into:

\[
    \min_{B,T} \text{rank}(B) + \lambda \|T\|_0 \quad \text{s.t.} \quad D = B + T ,
\]

where \( \lambda \) indicates a positive trade-off parameter, and \( \| \cdot \|_0 \) denotes the \( l_0 \)-norm as the number of nonzero entries.

However, solving (2) is NP-hard since the rank function and \( l_0 \)-norm are both non-convex and discontinuous. Considering this, models like IPI [10] individually replace them with the nuclear norm (\( \| \cdot \|_* \), the sum of singular values in a matrix) and \( l_1 \)-norm (\( \| \cdot \|_1 \)) through principal component pursuit (PCP) and reformulate (2) as:

\[
    \min_{B,T} \|B\|_* + \lambda \|T\|_1 \quad \text{s.t.} \quad D = B + T .
\]

In complex infrared scenarios, the background and target may vary in complexity and sparsity, and a single norm or rank function may not capture the practical constraints [49]. Thus, we use \( \mathcal{R}(B) \) and \( S(T) \) to constrain the prior knowledge of the background and target images, respectively:

\[
    \min_{B,T} \mathcal{R}(B) + \lambda S(T) \quad \text{s.t.} \quad D = B + T .
\]

Moreover, to simplify the complexity of updating variables due to the augmented Lagrange multipliers [6], we adopt a simpler and more intuitive \( l_2 \)-norm to transform the constrained problem into an unconstrained one [30] as:

\[
    \mathcal{L}(B, T) = \mathcal{R}(B) + \lambda S(T) + \frac{\mu}{2} \| D - B - T \|_F^2 ,
\]

where \( \mu \) is a penalty coefficient, and \( \| \cdot \|_F \) indicates the Frobenius norm (F-norm). For a matrix \( X \), its F-norm equals \( \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |X_{ij}|^2} \). Based on (5), we can optimize the background and target individually in an iterative scheme.

3.2. Model Iterative Solving

Updating \( B^* \): To update the background, we draw the sub-problem as:

\[
    B^* = \arg\min_B \mathcal{R}(B) + \frac{\mu}{2} \| B + T - D \|_F^2 .
\]

As (3) illustrates, former optimization-based ISTD approaches usually set \( \mathcal{R}(B) \) as \( \|B\|_* \), then degenerates (6) into the sum of the nuclear norm and \( l_2 \) norm, and this problem has an analytical solution as:

\[
    B^* = \mathcal{D}_\mu(D - T) ,
\]

where \( \mathcal{D}_\mu(\cdot) \) denotes the SVT operator [21] with the threshold of \( \mu \). However, solving (7) involves SVD and functions on its singular values. In DUNs, we emulate this using neural networks, which means we must perform SVD on each neural tensor in each forward propagation. This poses challenges for time consumption and accuracy. [4] proposes an initialization method based on the best rank-r approximation SVD, but still faces precision issues. [48] introduces a technique with different rank constraints for the submatrices, which improves interpretability but depends on manual tuning and ignores the intrinsic image properties when simulating matrix computation in neural layers.

Thus, instead of adopting the nuclear norm and solving it with complex SVDs, we degrade it to a constrain function \( \mathcal{R}(B) \) in this study, and introduce a proximal operator \( \text{prox}_\mu(\cdot) \) to approximate the closed-form solution for the background, which is formulated as:

\[
    B^* = \text{prox}_\mu(D - T) .
\]
We use convolutional layers to approximate proximal functions and solve the optimization problem. This method eliminates the need for complex matrix operations and leverages the nonlinear capabilities of neural networks to extract deep features from images in a data-driven way. And we will illustrate the detailed construction in Section 3.3.

**Updating \( T^* \):** Similar with (6), the sub-problem of optimizing the target is written as:

\[
T^* = \arg \min_T \lambda S(T) + \frac{\mu}{2} \| T + B - D \|_F^2 .
\]  

(9)

As discussed in Section 3.1, common optimization approaches impose an \( l_1 \) norm constraint on the sparse target image. However, this poses the challenge of mapping the soft thresholding into the neural network [42]. Moreover, sparse constraints often vary depending on the detection scenarios change [49]. Thus, we aim to derive a simpler and more intuitive representation for the closed-form solution of (9). To handle the above issues, we consider conducting Taylor expansion of \( S(T) \). For a function \( f(t) \), with its Lipschitz continuous gradient function \( \nabla f(t) \), then \( f(t) \) can be Taylor approximated at the a fix point \( t_0 \) by:

\[
\hat{f}(t, t_0) \approx \frac{L}{2}||t - t_0 + \frac{1}{L}f(t_0)||^2 + C .
\]  

(10)

where \( L \) is a constant, \( C = -\frac{L^2}{2}||\nabla f(t_0)||^2 + f(t_0) \) (please refer to the supplementary material for detail deductions). Based on this, we approximate \( S(T) \) at the last \( T^{k-1} \) as:

\[
\hat{S}(T, T^{k-1}) \approx \frac{L_S}{2}||T - T^{k-1} + \frac{1}{L_S}\nabla S(T^{k-1})||^2 + C_s ,
\]  

(11)

where \( L_s \) is the Lipschitz constant of \( S(T) \) and \( C_s = \frac{1}{2L^2}||\nabla S(T^{k-1})||^2 + S(T^{k-1}) \) represents as a constant. And the updating formula of the target matrix can be substituted as:

\[
T^* = \arg \min_T \lambda \hat{S}(T, T^{k-1}) + \frac{\mu}{2} \| T + B - D \|_F^2 .
\]  

(12)

which simply involves the sum of two \( l_2 \) norms, rather than a traditional \( l_1 \) norm constraint optimization problem, and therefore does not require the use of conventional algorithms or simulating soft thresholding [42]. By taking the derivative of the equation and equating it to zero, a closed-form solution for updating \( T \) at the \( k \)-th step can be derived:

\[
T^k = \frac{\lambda L_S}{\lambda L_S + \mu} T^{k-1} + \frac{\mu}{\lambda L_S + \mu} (D^{k-1} - B) - \frac{\lambda}{\lambda L_S + \mu} \nabla S(T^{k-1}) ,
\]  

(13)

where all three coefficients are constant values. By assigning each of them a learnable vector, the final equation for updating the target matrix can be reformulated:

\[
T^k = \gamma T^{k-1} + (1 - \gamma)(D^{k-1} - B) - \epsilon \nabla S(T^{k-1}) ,
\]  

(14)

where \( \gamma = \frac{\lambda L_S}{\lambda L_S + \mu} , \epsilon = \frac{\lambda}{\lambda L_S + \mu} \). We learn the function \( \nabla S \) end-to-end without complex matrix operations such as soft thresholding, satisfying the Lipschitz continuity assumption. And the updating equation for reconstructed \( D^k \) is:

\[
D^k = B^k + T^k .
\]  

(15)

**3.3. RPCANet Framework**

This section describes the overall architecture of the network and module design of RPCANet, based on the optimization equations in Section 3.2. As shown in Fig. 2, the input to the network is an infrared image \( X \in \mathbb{R}^{H \times W} \) with targets, where \( H \) and \( W \) are the image height and width, and the update parameters are initialized as \( D^0 = X \) and \( T^0 = 0 \). These parameters are then passed through \( K \) decomposition stages, each corresponding to an iterative matrix low-rank sparse decomposition process, to simulate the update operation of multiple iterations in model-driven approaches.

In detail, the updated parameters \( D^{k-1} \) and \( T^{k-1} \) are fed into the \( k \)-th decomposition stage, where \( k \in \{1, \ldots, K\} \). The background \( B^k \), target \( T^k \), and reconstructed result \( D^k \) for the current stage are estimated by BEM, TEM, and IRM, respectively. Typically, \( B^k \) represents the latent variable of the current decomposition stage and is not involved in the parameter transfer between stages.
Network Design shows. Here, in RPCANet: background extraction module (BEM), target extraction module (TEM), and image reconstruction module (IRM).

**Background Estimation Module (BEM):** As shown in Fig. 3, we adopt BEM to estimate the background. In (8), proximal operator prox\(_\mu(\cdot)\) is to-be-decided. Here, inspired by former DUN-based works [29, 32, 33, 50] and in the spirits of flexibility [41], we adopt a residual structure proxNet(\(\cdot\)) to simulate this operator. As (16) formulates:

\[
B^k = \text{proxNet}(D^{k-1} - T^{k-1}) = D^{k-1} - T^{k-1} + F^k(D^{k-1} - T^{k-1}),
\]

(16)

where \(F^k(\cdot)\) indicates the 3 \(\times\) 3 convolution group in the structure as Fig. 3 shows. Here, \(F^k(\cdot)\) consists of \(l_B\) middle layers [\(\text{Conv} + BN + \text{ReLU}\)], and two convolution layers: feature extraction layer [\(\text{Conv} + BN + \text{ReLU}\)] and image reconstruction layer [\(\text{Conv}\)], respectively. Here, BN stands for batch normalization and ReLU for rectified linear unit [22]. All of them are in the stride of 1 and the padding of 1, and we set \(C = 32\) and \(l_B = 3\) in this study.

**Target Extraction Module (TEM):** This module takes the updated background \(B^k\), target \(T^{k-1}\), and reconstruction result \(D^{k-1}\) as inputs, as shown in Fig. 3. Moreover, we set \(\gamma = 0.5\) to evenly treat the three inputs and rewrite (14) as:

\[
T^k = T^{k-1} + D^{k-1} - B^k - \varepsilon\nabla S(T^{k-1}).
\]

(17)

We assign the parameter learning task to \(\varepsilon\) as \(\varepsilon^k\), which is a learnable scalar independent of each reconstruction stage and does not share parameters. As to the Lipschitz continuous gradient function \(\nabla S\), [31] finds that a single-layer CNN consisting of a convolution layer and a ReLU activation function is Lipschitz continuous, which also holds for multiple stacked layers. Therefore, to avoid complex network design, we adopt simple convolution layers and ReLU to simulate the function \(\nabla S\) as shown in the orange box in Fig. 3, in accordance with the assumption in Section 3.2. We also introduce the difference of last \(D^{k-1}\) and updated \(B^k\) to enhance successive target feature, and the update function of \(T^k\) is written as:

\[
T^k = T^{k-1} + D^{k-1} - B^k - \varepsilon^k G^k(T^{k-1} + D^{k-1} - B^k).
\]

(18)

Specifically, \(G^k(\cdot)\) comprises an initial convolution layer, \(l_T\) of middle layers, and a reconstruction layer. Since the Lipschitz continuous property of the statistical operation BN is not clear [31], we omit the BN from the convolution block.

**Image Reconstruction Module (IRM):** To align with the RPCA process that maps the target and background into a restored image, we devise an IRM that converts the decomposition task into an image reconstruction task with a neural network \(M^k(\cdot)\), as presented in Fig. 3:

\[
D^k = M^k(B^k + T^k).
\]

(19)

Instead of applying residual blocks or other complex networks in the reconstruction module [32, 33], we employ a simple and decent CNN architecture [43] in learning image features and mapping the decomposed background and target effectively. Similar to \(F^k(\cdot)\), \(M^k(\cdot)\) has three types of convolution layer with \(l_D\) middle layer, where \(l_D = 3\).

The corresponding updating operations in optimization-based and our DUN-based frameworks are shown in Fig. 4. To sum up, we propose an end-to-end training framework for the ISTD task named RPCANet. Its trainable parameters set \(\Theta\) includes the convolutional network parameters in BEM, TEM, and IRM in each decomposition stage, and the independent trainable scalar \(\varepsilon\) in TEM, which collects as \(\Theta = \{\Theta^k_{\text{BEM}}, \Theta^k_{\text{TEM}}, \Theta^k_{\text{IRM}}, \varepsilon^k\}_{k=1}^K\), where \(K\) is the number of overall decomposition stages.

4. Network Training

4.1. Training Loss

Since we separate an ISTD task into the target segmentation and infrared image reconstruction, thus the loss func-
The former measures the target segmentation performance using SoftIoU [24], while the latter measures the infrared image reconstruction performance using the least squares error between the reconstructed image and the original image. The loss function is defined as follows:

\[
\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{segmentation}} + \tau \cdot \mathcal{L}_{\text{fidelity}}
\]

\[
= 1 - \frac{1}{N_t} \sum_{i=1}^{N_t} \frac{TP}{FP+TP+FN} + \tau \cdot \frac{N}{N_t} \sum_{i=1}^{N_t} \left\|D^K_i - D^K\right\|^2,
\]

where \(N_t\) and \(N\) are the total training number and total pixels per image. \(\tau\) is the regularization parameter and set to 0.01 in our experiment.

### 4.2. Implementation Details

We conduct experiments on three publicly available datasets: SIRST-Aug [47], NUDT-SIRST [19], and IRSTD-1K [46], with their split strategies, all images are normalized into 256 × 256. Besides, evaluation metrics are two target-level: probability of detection (\(P_d\)) and false alarm rate (\(F_a\)); two pixel-level: mean intersection over union (mIoU) and F-measure (\(F_1\)); and the receiver operating characteristic (ROC) curve and the area under ROC (AUC).

We train our model for 400 epochs in the PyTorch framework on each dataset using an Nvidia GeForce 3090 GPU. We use an Adam optimizer with an initial learning rate of \(10^{-4}\) in the poly policy, where the initial learning rate is multiplied by \((1 - \frac{\text{iter}}{\text{total\_iter}})^{0.9}\) and a batch size of 8.

### 5. Experiment

To confirm the fundamental process of the suggested network, we verify the model by visualizing mid-layers and conduct ablation studies. Next, we describe experiments on open-source datasets for performance evaluation.

#### 5.1. Model Verification

Fig. 5 presents the heatmaps of the intermediate update variable \(B^k\) and \(T^k\), where \(k \in \{2, 4, 6\}\). At lower stages, the network prioritizes low-level edge texture information in the background, leading to insufficient background reconstruction. As the decomposition deepens, the network acquires more comprehensive information, demonstrating its ability to learn from formula-guided training.

Likewise, \(T^k\) progressively learns the target’s location from disorganized high-frequency information, resulting in a sparse matrix as the segmented output. Fig. 5 (c) shows that false alarms in the initial stages are eliminated under mask supervision, showing the data-driven way corrects the detection result. These successive heatmaps verify the effectiveness of RPCANet in learning infrared variables according to training data while preserving the interpretability of RPCA model that facilitates the IST task.

#### 5.2. Ablation Studies

**Effects of Parameter \(K\) and \(l_\tau\):** Table 1 compares...
the effects of different reconstruction stage counts. Our method only reaches significant detection performance in two stages, validating the essential attributes of the suggested RPCANet. Additionally, we see that $K = 7$ has a worse ISTD performance than $K = 6$, which makes sense given that a higher K could hinder gradient propagation, based on this, we set $K$ to 6. Similarly, as shown in Table 2, the number of the convolutional layer $l_T$ faces the same manner, the performance of the network can be improved as $l_T$ raises, but the gain effect of too many reconstruction stages on the network performance is limited, thus, we take $l_T = 6$ in all our experiments.

### Studies of proxNet(·) and IRM

The emulation of proximal operators acts an essential role in background estimation [33, 41], we investigate two proxNet(·) constructions (residual block (RB) [32] and plain CNN) for proximal operators. Table 3 demonstrates that RB can yield some effects and a small parameter count, our network in Section 3.3 achieves better results with a minimal parameter increase. We also examine the influence of the IRM. As depicted in the third row of Table 3, RPCANet, guided by the reconstruction module, exhibits substantial improvements in four metrics, proving the module’s effectiveness.

### Studies of $\nabla S$ Simulation Network

We investigate the impact of incorporating $D^{k-1}$ and $B^k$ subtraction within $G^k(\cdot)$ for target feature enhancement. Table 4 compares two configurations. While using $T^{k-1}$ alone yields satisfactory results, incorporating additional information in (18) significantly improves mIoU and $F_1$. We can conclude that appropriate modifications to the network structure, guided by the model’s prior, can enhance detection performance.
Table 5. mIoU (%), F1 (%), Pd (%), Fd (10^-5), and runtime values of different methods performance on NUDT-SIRST [19], IRSTD-1K [46], and SIRST-Aug [47]. The second column records the parameter statistics for data-driven algorithms.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tophat [2]</td>
<td>-</td>
<td>22.23 36.37 96.19 76.27</td>
<td>5.35 10.15 68.73 86.50</td>
<td>16.70 28.62 95.05 11.77</td>
<td>0.011/-</td>
<td>-</td>
</tr>
<tr>
<td>MPCM [36]</td>
<td>-</td>
<td>9.26 16.95 70.58 32.74</td>
<td>14.87 25.89 68.73 6.51</td>
<td>19.76 33.00 93.40 3.14</td>
<td>0.0624/-</td>
<td>-</td>
</tr>
<tr>
<td>IPI [10]</td>
<td>-</td>
<td>34.62 51.43 92.38 7.54</td>
<td>18.64 31.42 78.01 11.10</td>
<td>21.93 35.87 80.06 1.62</td>
<td>3.0972/-</td>
<td>-</td>
</tr>
<tr>
<td>PSTNN [45]</td>
<td>-</td>
<td>25.46 40.58 78.52 7.95</td>
<td>16.38 28.15 69.07 7.65</td>
<td>13.83 24.30 59.97 7.65</td>
<td>0.1249/-</td>
<td>-</td>
</tr>
<tr>
<td>ACM [7]</td>
<td>0.398M</td>
<td>69.00 81.66 95.98 13.34</td>
<td>61.56 76.20 92.93 8.88</td>
<td>70.49 82.62 96.70 35.29</td>
<td>-0.0072</td>
<td>-</td>
</tr>
<tr>
<td>ALCNet [8]</td>
<td>0.427M</td>
<td>71.48 83.37 96.30 11.45</td>
<td>58.23 73.61 92.92 10.31</td>
<td>66.21 79.67 97.80 37.40</td>
<td>-0.0070</td>
<td>-</td>
</tr>
<tr>
<td>ISNet [46]</td>
<td>0.967M</td>
<td>87.51 93.34 97.35 3.37</td>
<td>55.29 71.21 94.61 14.19</td>
<td>70.51 82.71 97.66 31.57</td>
<td>-0.0132</td>
<td>-</td>
</tr>
<tr>
<td>AGPCNet [47]</td>
<td>12.360M</td>
<td>85.40 92.13 98.10 4.72</td>
<td>61.00 75.76 89.35 5.34</td>
<td>72.16 83.83 99.03 35.56</td>
<td>-0.0205</td>
<td>-</td>
</tr>
<tr>
<td>DNANet [19]</td>
<td>4.697M</td>
<td>83.94 91.27 98.52 6.21</td>
<td>60.51 75.40 91.07 5.43</td>
<td>69.58 82.06 96.14 27.26</td>
<td>-0.0250</td>
<td>-</td>
</tr>
<tr>
<td>UIUNet [38]</td>
<td>50.540M</td>
<td>88.71 94.01 91.43 1.89</td>
<td>63.06 77.35 93.60 6.57</td>
<td>71.80 83.59 98.35 28.29</td>
<td>-0.0261</td>
<td>-</td>
</tr>
<tr>
<td>Ours</td>
<td>0.680M</td>
<td>89.31 94.35 97.14 2.87</td>
<td>63.21 77.45 88.31 4.39</td>
<td>72.54 84.08 98.21 34.14</td>
<td>-0.0096</td>
<td>-</td>
</tr>
</tbody>
</table>

5.3. Comparison to State-of-the-art Methods

For model-driven methods, we select: filter-based Tophat [2], HVS-based MPCM [36], matrix optimization-based IPI [10], and tensor optimization-based PSTNN [45]. For data-driven algorithms, we conduct experiments on ACM [7], ALCNet [8], ISNet [46], AGPCNet [47], DNANet [19], and UIUNet [38].

**Qualitative Results:** Fig. 6 depicts visual results obtained from various algorithms applied to three datasets. RPCANet effectively generalizes complex scenarios, producing outputs with accurate target shapes and low false alarm rates. Compared to model-driven algorithms, our network excels at suppressing false alarms while preserving the sparse shape of infrared targets. Although some DL-based models can effectively reduce false alarms, they may struggle in small and complex situations, leading to missed detections. Algorithms like ISNet largely maintain the target shape; however, they occasionally encounter difficulties in multi-target scenarios. The visualization results clearly illustrate that our network combines the advantages of optimization-based and DL-based methods.

**Quantitative Results:** To showcase the effectiveness of our network, we compare its detection performance with the SOTA baseline. Table 5 demonstrates that RPCANet outperforms most model-driven and data-driven ISTD frameworks in four indicators using fewer parameters. Model-driven methods like IPI excel in target-level metrics but lack pixel-level accuracy. Data-driven networks improve mIoU and F1 scores while preserving the target's shape. However, DL-based algorithms like AGPCNet and UIUNet suffer from overfitting and require more parameters. In contrast, RPCANet combines model-driven priors with accurate object extraction and segmentation guided by data, achieving superior performance with fewer parameters. ROC curves in Fig. 7 highlight that our model rapidly reaches the upper-left corner and exhibits competitive performance in terms of AUC: RPCANet (0.9857), ACM (0.9830), and UIUNet (0.9477). Moreover, the last column in Table 5 shows that our framework obtains great computational efficiency.

6. Conclusion

In this paper, we propose an interpretable framework for infrared small target detection based on deep unfolding networks. We model the ISTD task as a relaxed RPCA problem and solve the optimization steps via network emulations, including the proximal network and sparsity-constrained neural layers. Our scheme produces trustworthy visualizations and outstanding detection results in extensive experiments on various public datasets. The architecture of RPCANet effectively guides neural layers to learn low-rank backgrounds and sparse targets, facilitating the detection tasks in an almost “white box” manner. We hope our findings cloud inspire researchers to explore more interpretable solutions for ISTD problems in the future.

Acknowledgments

This work was supported by the Natural Science Foundation of Sichuan Province of China (No.2022NSFSC40574) and National Natural Science Foundation of China (No.61775030, No.61571096).
References


