Direction-Coded Temporal U-Shape Module for Multiframe Infrared Small Target Detection

Ruojing Li, Wei An, Chao Xiao, Boyang Li, Yingqian Wang, *Member, IEEE*, Miao Li, and Yulan Guo, *Senior Member, IEEE*

***Abstract*— Infrared small target (IRST) detection aims at separating targets from cluttered background. Although many deep learning-based single-frame IRST (SIRST) detection methods have achieved promising detection performance, they cannot deal with extremely dim targets while suppressing the clutters since the targets are spatially indistinctive. Multiframe IRST (MIRST) detection can well handle this problem by fusing the temporal information of moving targets. However, the extraction of motion information is challenging since general convolution is insensitive to motion direction. In this article, we propose a simple yet effective direction-coded temporal U-shape module (DTUM) for MIRST detection. Specifically, we build a motion-to-data mapping to distinguish the motion of targets and clutters by indexing different directions. Based on the motion-to-data mapping, we further design a direction-coded convolution block (DCCB) to encode the motion direction into features and extract the motion information of targets. Our DTUM can be equipped with most single-frame networks to achieve MIRST detection. Moreover, in view of the lack of MIRST datasets, including dim targets, we build a multiframe infrared small and dim target dataset (namely, NUDT-MIRSDT) and propose several evaluation metrics. The experimental results on the NUDT-MIRSDT dataset demonstrate the effectiveness of our method. Our method achieves the state-of-the-art performance in detecting infrared small and dim targets and suppressing false alarms. Our codes will be available at https://github.com/TinaLRJ/Multi-frame-infrared-small-target-detection-DTUM.**

***Index Terms*— Direction coding, infrared small target (IRST) detection, point-level supervision, spatial–temporal fusion.**

I. Introduction

INFRARED small target (IRST) detection aims at accurately locating small targets in various infrared backgrounds, which has been widely used in different fields, such as robust visual system [1], [2] and maritime surveillance [3]. Different from generic objects, IRSTs are small [4], shapeless [5], changeable [6], [7], and sometimes immersed in complex backgrounds. These characteristics introduce significant challenges to IRST detection.

Traditional paradigms, including multiple-filter-based methods [8], [9], [10], [11], sparsity-and-low-rank-based methods [12], [13], and human-visual-system (HVS)-based methods [14], have been deeply studied for IRST detection. Although promising results have been achieved, these methods essentially rely on handcrafted features and prior knowledge. Once the scenes dramatically change, these methods with fixed hyperparameters cannot well adapt to the scenarios and thus suffer significant performance degradation.

Due to the powerful modeling capability, deep learning-based methods have achieved remarkable performance improvement and thus attracted increasing research interests in recent years. Different from traditional methods, deep learning-based methods can extract highly discriminative characteristics of the targets from numerous data in a learnable manner. For example, Dai et al. [15] proposed the first segmentation-based single-frame IRST (SIRST) detection method using an asymmetric contextual modulation (ACM) module. This method achieves good detection performance via the interaction of high-level semantics and low-level details. Li et al. [5] proposed a dense nested attention network (DNANet) to enhance this interaction. Wu et al. [16] embedded a tiny UNet into a larger UNet backbone (i.e., UIUNet) to enable multilevel and multiscale representation learning of IRSTs. Nevertheless, once the target is dim and not salient (e.g., immersed by strong clutter) in the image, these methods tend to fail to detect targets using spatial information only. Extracting the spatial–temporal characteristics from multiple frames can effectively handle this issue. However, deep learning-based multiframe IRST (MIRST) detection is still in its primary stage due to the following challenges.

First, it is challenging to extract the motion information of targets from multiple frames since general convolution is insensitive to the motion direction of targets. After training, the convolutional kernel weights are fixed and are not affected by the relationship among pixels, which is important for the extraction of motion information. Second, there are few MIRST datasets, including dim targets with pixel-level annotations in this field, and the insufficient publicly available datasets obstruct the development of MIRST detection. Sun et al. [17] published the only MIRST dataset with mask, bounding box, and central pixel labels. However, it is unsuitable to be used to evaluate the performance of the methods on dim targets.

To tackle the aforementioned challenges, we propose a direction-coded temporal U-shape module (DTUM), which

Manuscript received 2 September 2022; revised 15 April 2023 and 26 August 2023; accepted 30 October 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62001478. *(Corresponding author: Miao Li.)*

The authors are with the College of Electronic Science and Technology, National University of Defense Technology (NUDT), Changsha 410000, China (e-mail: liruojing@nudt.edu.cn; anwei@nudt.edu.cn; xiaochao12@nudt.edu.cn; liboyang20@nudt.edu.cn; wangyingqian16@nudt.edu.cn; lm8866@nudt.edu.cn; yulan.guo@nudt.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2023.3331004.

Digital Object Identifier 10.1109/TNNLS.2023.3331004

can be equipped with most single-frame detection methods to achieve MIRST detection. First, we build a motion-to-data mapping to make the difference of target motion and clutter motion measurable. The mapping value represents the consistency of motion among frames, which is the distinctive property used in the module. Then, based on this mapping, we design a direction-coded convolution block (DCCB) to obtain the motion characteristics of the targets via encoding the target position into features. As shown in Fig. 1, after DCCB, the target will be enhanced since its motion is consistent in frames. Through several DCCBs and the connection between high- and low-level features, our module can help to detect the infrared small and dim target while suppressing the clutters. In addition, we develop a multiframe infrared small and dim target dataset (NUDT-MIRSDT) with mask and point-level labels, covering various scenes and extremely weak targets.

In summary, the main contributions of this work are summarized as follows.

1) We propose a simple yet effective DTUM to enhance dim targets while suppressing false alarms for MIRST detection. DTUM can be equipped with most single-frame networks to leverage spatial–temporal information for MIRST.
2) A motion-to-data mapping is built to distinguish targets and clutters according to the consistency of their motion. Besides, we design a DCCB to extract motion information by encoding the motion direction into features.
3) We develop an NUDT-MIRSDT dataset with both mask and point-level annotations for MIRST detection. Moreover, we preliminarily explore the point-level supervision on our dataset.
4) Experimental results demonstrate the superiority of our method. Compared with existing methods, our method achieves better detection performance, especially on dim targets with signal-to-noise ratio (SNR) lower than 3.

II. Related Work

In this section, we briefly review the major works in IRST detection, IRST dataset, and point-level supervision tasks (e.g., localization and segmentation).

A. IRST Detection

It has achieved advanced progress recently. Many traditional and deep learning-based methods are proposed for this task. According to the forms of input data, these methods can be classified into SIRST detection methods and MIRST detection methods.

SIRST detection is based on spatial local saliency of targets [18], [19], which is similar to salience object detection [20], [21], [22], [23], [24], [25]. Traditional methods include spatial-filtering-based methods [11], sparsity-and-low-rank-based methods [12], [13], and HVS-based methods [14]. Deep learning-based methods usually achieve better performance due to data-driven feature learning. Dai et al. [19] improved their ACM model with a local contrast measure method [26] to make the relatively long-range contextual interactions with clear physical interpretability. Li et al. [5] further relieved the opposition of high-level information acquisition and the low response of small target using a dense nested interactive module (DNIM). Moreover, Zhang et al. [27] built a Taylor finite difference (TFD)-inspired edge block to pay attention on shape matters for detecting IRSTs in shape. Wu et al. [28] designed a multilevel TransUNet with a specially designed copy-rotate-resize-paste data augmentation and FocalIoU loss to achieve both precise target localization and shape description. However, IRSTs are usually immersed in heavy clutter and complex background. Once the target is not visually salient in a single image, SIRST detection methods will suffer performance degradation. It is important to incorporate spatial and temporal information in a video for detecting infrared small and dim targets.

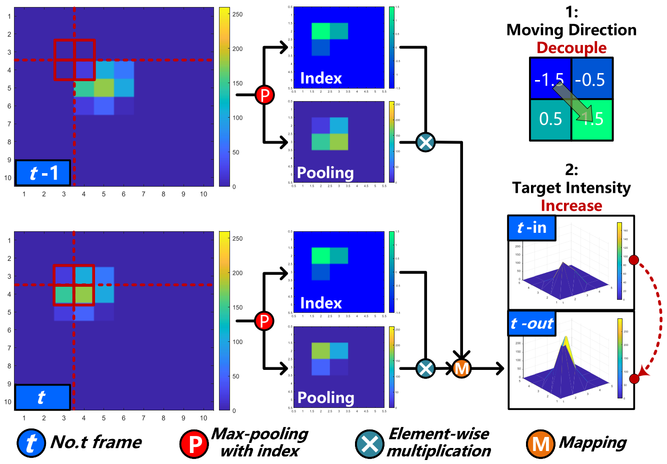


Fig. 1. Illustration of the target motion perception process using the proposed DCCB. The images in the middle column are the intermediate results of DCCB with direction coding. The bottom right image is the output map with the target significantly enhanced.

MIRST detection methods can incorporate both spatial local saliency and motion information of targets. MIRST detection methods can detect targets while reducing false alarms. Several traditional methods were proposed for MIRST detection. Reed et al. [8], [9], [10] proposed a typical method (i.e., 3-D matched filtering) to obviously improve the SNR of targets through spatial filtering and temporal energy accumulation. Sun et al. [29] proposed a multiple subspace learning and spatial–temporal patch-tensor (MSLSTIPT) model to detect infrared small and dim targets. Then, Liu et al. [30] developed a nonconvex tensor low-rank approximation (NTLA) method to achieve accurate background estimation. Deep learning-based methods have been successfully applied in small moving target detection for remote sensing (e.g., DSFNet [31]) and for visual systems (e.g., STMD+ [1] and feedback STMD [2]). Besides, there are some networks [32], [33] based on detection framework for MIRST detection. However, there is no open-source deep learning-based method based on a segmentation framework for MIRST detection due to the lack of suitable multiframe dataset.

Therefore, we focus on multiframe information to achieve good performance in detecting infrared small and dim targets.

B. Datasets for IRST Detection

Large and high-quality dataset is crucial for deep learning-based methods. There are two types of IRST datasets, single-frame datasets and multiframe datasets.

There are many single-frame datasets acquired from different scenes. All these frames are labeled with mask. The first published synthetic dataset is NUST-SIRST [34], which contains 10 100 images from real scenes with simulated targets. Most targets in this dataset are large (>9 × 9) and bright. Dai et al. [15] released the first real dataset (i.e., NUAA-SIRST), which has totally 427 images that are insufficient for network training. Subsequently, larger datasets were developed, e.g., NUDT-SIRST [5] with 1327 images and IRSTD-1k [27] with 1001 images.

Most multiframe datasets are labeled with bounding boxes or position coordinates. Fu et al. [35] built an infrared target detection and tracking dataset, including 21 750 frames from 87 sequences with bounding boxes and trajectory annotations. The first infrared small aerial target dataset (SIATD) [33] was developed with position annotations, which contains 150 185 frames from 350 sequences. Sun et al. [17] developed an infrared dim small target dataset (IRDST) with labels in three types (mask, bounding box, and central pixel). There are totally 142 727 frames from 5930 sequences. Most targets in these datasets are intense and salient in the space domain. However, as for MIRST methods, the detection performance on dim targets is very important for their evaluation. There is no special MIRST dataset for dim targets.

C. Point-Level Supervision

It is a common weak supervision manner and has been widely used in many tasks, such as object detection [36], [37], instance segmentation [38], [39], and object counting [40], [41]. Bearman et al. [42] incorporated point supervision along with an objectness prior to the training loss function and achieved comparable results to those fully supervised semantic segmentation methods. Laradji et al. [38] proposed a point-level annotation method for instance segmentation. They used both point annotation and pseudo mask to supervise the training process. Liu et al. [43] proposed a deep detection network with only point supervision to achieve crowd counting. They mined useful person size information from point-level annotations.

All these works aim at segmenting and localizing generic objects with rich texture, shape, and context information. However, few works exploited the point-level supervision in IRST detection, even in small target detection.

III. Methodology

In this section, we introduce the motion-to-data mapping and its implementation, i.e., DCCB. Then, we introduce the DTUM, which consists of 3-D convolution layers and DCCBs. Besides, we describe our NUDT-MIRSDT dataset and a hard point mining (HPM) loss function for point-level supervision.

A. Motion-to-Data Mapping

In a single frame, the targets and clutters have similar spatial characteristics. Therefore, it is important for the detection methods to extract temporal characteristics due to the distinctive motion of the target. In this section, we provide a method to map different types of motion among frames to different value intervals. The results of mapping are beneficial for the distinction of the targets and clutters. Without considering the camera movement, the biggest difference between target motion and clutter motion is the consistency of the motion direction in frames. Motion-to-data mapping can make this difference measurable, and it is the base of the extraction of motion information.

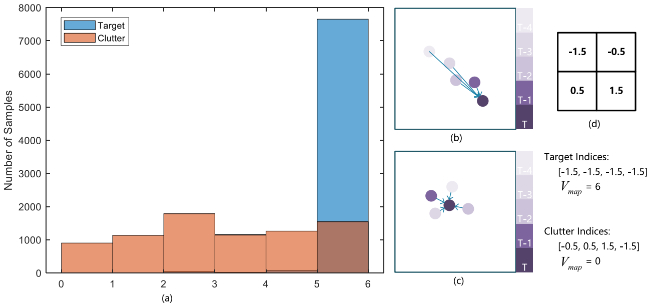


Fig. 2. Motion-to-data mapping. (a) Statistical results of the mapping values (i.e., sum of indexes) of targets and clutters. (b) Sample of the target motion in five frames. (c) Sample of the clutter motion in five frames. (d) Mapping index matrix. The target trajectory is more consistent, and its mapping value (i.e., V_{map}) is larger than the clutter trajectory.

According to the motion characteristics of targets and clutters, the mapping should meet two requirements. First, the value should decrease when the motion direction is reversed. Second, the value should achieve its maximum only when the motion direction is constant among input frames. To satisfy the above requirements, the mapping index matrix is specially designed, as shown in Fig. 2(d). The final mapping value is defined as

$$V_{\text{map}} = \left| \sum_{i=1}^{4} I_{t-i\to t} \right| \tag{1}$$

where $I_{t-i\to t}$ represents the index value of the motion direction from the $(t-i)$th frame to the tth frame. According to the index matrix, we count the mapping values of the targets and clutters, as shown in Fig. 2(a). Most sums of target motion direction indices among five frames are around 6, while those of clutter motion direction indices are evenly distributed between 0 and 6. The statistical distribution demonstrates that most target trajectories are mapped to large values via our motion-to-data mapping. Meanwhile, most mapping values of clutter trajectories are lower than those of target trajectories. Therefore, after the motion-to-data mapping, the difference between target motion and clutter motion is obvious and measurable and is easier for the network to learn.

B. Direction-Coded Convolution Block

In IRST detection, the motion difference between the targets and clutters is important to distinguish them. However, it is hard to perceive different motion directions accurately since the general 3-D convolution is a weighted summation of local features with fixed weights. The motion-to-data mapping is an efficient way to extract motion characteristics. The DCCB is the implementation of motion-to-data mapping in the temporal module, including a direction-coded max-pooling layer and a temporal convolution layer. To obtain the relative location of the target, we put the direction coding into max-pooling layer with the mapping index, considering that the index of local maximum can represent the relative location of the high response area. To capture the change of motion direction among frames, we use the temporal convolution layer to learn the mapping process from the indices in multiple frames to the mapping value V_{map}.

The max-pooling layer in DCCB is similar to the general max-pooling layer, where the output of the layer is replaced by the local maximums with direction information. The formulations of max-pooling layer in common and in DCCB can be written as

$$y = f_{\max}(x, k) \tag{2}$$

$$\begin{cases} y, \text{id} = f'_{\max}(x, k, I) \\ y' = y \cdot \text{id} \end{cases} \tag{3}$$

where x is the input pixels and k is the size of pooling kernel. I is the mapping index matrix, id is the direction coding of the output y', and $f_{\max}(\cdot)$ and $f'_{\max}(\cdot)$ represent the operation of max-pooling layer in general layer and DCCB, respectively.

The detailed structure of our DCCB is shown in Fig. 3(b). The input feature $F_{\text{in}} \in \mathbb{R}^{C_{i-1}\times 5\times H_{i-1}\times W_{i-1}}$ of the ith DCCB is first processed by direction-coded max-pooling layer to obtain the direction feature $F_{\text{direction}} \in \mathbb{R}^{C_i\times 5\times H_i\times W_i}$, i.e.,

$$F_{\text{direction}} = M_k(F_{\text{in}}) \otimes I_m \tag{4}$$

where $M_k(\cdot)$ represents the max-pooling operation with the kernel size of $k \times k$, I_m denotes the maximum index of each pixel in $M_k(F_{\text{in}})$, and $\otimes$ denotes the elementwise multiplication. Taking some pixel as the origin, $F_{\text{direction}}$ contains the position of the target relative to the origin in each frame.

To extract the motion information, a temporal fusion operation is needed to make the network focus on the change of target position among frames. We use a 5 × 1 × 1 convolution layer to catch this change and generate the final mapping result $F^i_{\text{motion}} \in \mathbb{R}^{C_i\times 5\times H_i\times W_i}$. The temporal fusion operation can be described as

$$\begin{aligned} F^i_{\text{motion}} &= H^i_{5\times 1\times 1}(F_{\text{direction}}) \\ &= \sigma\Big(\text{BN}\big(|\text{conv}(F_{\text{direction}})|\big)\Big) \end{aligned} \tag{5}$$

where $H^i_{5\times 1\times 1}$ represents a 5 × 1 × 1 convolution layer in the ith DCCB, and conv, BN, and σ denote the convolution, batch normalization, and ReLU activation, respectively.

Importantly, there is an operation in (5) changing the feature into the absolute value. As shown in Fig. 2, the index matrix has positive and negative values to achieve the functions of decreasing the mapping value of inconsistent motion and increasing that of consistent motion. If the target position is encoded by negative indices, this operation can ensure the mapping value increase. The absolute value represents the response to the targets in the feature map, which indicates the probability of the existence of the target. Finally, the DCCB helps the network to implement the motion-to-data mapping and extract the motion information of the targets.

C. Direction-Coded Temporal U-Shape Module

The DTUM is designed to extract temporal information and fuse spatial–temporal characteristics. Inspired by [44], DTUM is based on UNet, and it is simple and efficient to capture contextual information. Considering the salient difference of target motion and clutter motion, a simple structure and shallow semantic representation can be sufficient for this task, especially with our DCCB. Therefore, we construct the module with a framework similar to UNet to achieve MIRST detection simply and effectively, as shown in Fig. 3. Differently, besides the spatial features, DTUM pays more attention to the temporal features with the help of 5 × 1 × 1 convolutions. Besides, DTUM can perceive motion characteristics with the help of DCCBs.

To be equipped with most single-frame detection methods, the input of DTUM is the concatenation of the spatial features of multiple consecutive frames generated by the single-frame method, i.e., $F_{\text{spa}} \in \mathbb{R}^{C\times 5\times H\times W}$. The channel number C is large to ensure that the temporal module can receive sufficient spatial features and fuse spatial–temporal features well. The input feature F_{spa} is first processed by a batch normalization layer [45] and a ReLU layer to normalize the input. Then, the feature is sent to a conv3d layer $H^0_{5\times 1\times 1}$ to fuse the spatial features from different frames and suppress some simple false alarms. Next, to extract the motion information, a set of DCCBs H^k_{DCCB} are used to encode the direction and map the motion, which are only used in the first half of DTUM since the coded motion details mostly exist in low-level features. Finally, the motion feature $F_{\text{motion}} \in \mathbb{R}^{C\times 5\times H/8\times W/8}$ is generated. This process can be described as

$$F_{\text{motion}} = H^k_{\text{DCCB}}\Big(H^0_{5\times 1\times 1}\Big(\sigma\big(\text{BN}(F_{\text{spa}})\big)\Big)\Big). \tag{6}$$

Then, F_{motion} is recovered to the same size of the input feature F_{spa} through several upsampling layers, conv3d layers, and skip connections. Finally, after fusing the normalized spatial feature, the predicted results can be obtained. In this article, we cascade three DCCBs (i.e., $k = 1, 2, 3$) to code the motion direction and distinguish targets and clutters. Consequently, the motion information can be extracted and fused with spatial features, resulting in notable detection performance improvements.

D. NUDT-MIRSDT Dataset

We develop a new dataset named NUDT-MIRSDT for MIRST detection. There are 10 000 frames from 100 sequences (80 for training and 20 for test) and with 9523 targets in total. Most of these target are smaller than 9 × 9. The scenes cover sky, sea, and land. The sequences are generated by shaking and adding noise to the infrared images captured from real world. The pixel- and point-level annotations are provided. The labeled point is the brightest point of the target, which is generated by a self-designed program. The shape of the target is always changing in a sequence to increase the diversity of samples. Besides, the target trajectory is a regular curve with jitter, which is randomly generated.

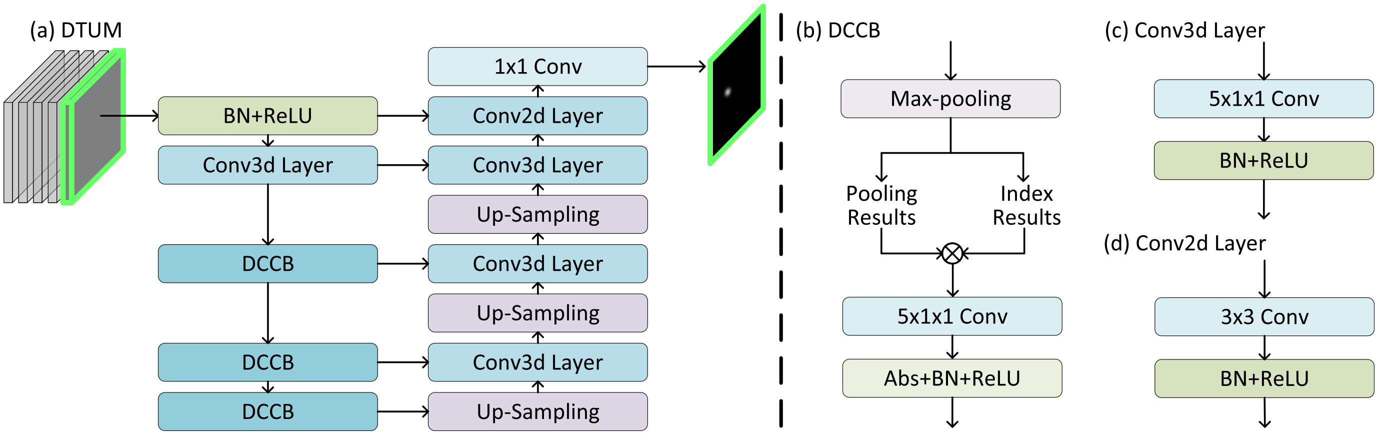


Fig. 3. DTUM architecture. (a) Structure of DTUM. (b) Structure of DCCB. (c) Structure of conv3d layer in DTUM. (d) Structure of conv2d layer in DTUM. The input of DTUM is the spatial features off multiple consecutive frames. Through a U-shape temporal convolution module, the motion features are extracted and fused with spatial features.

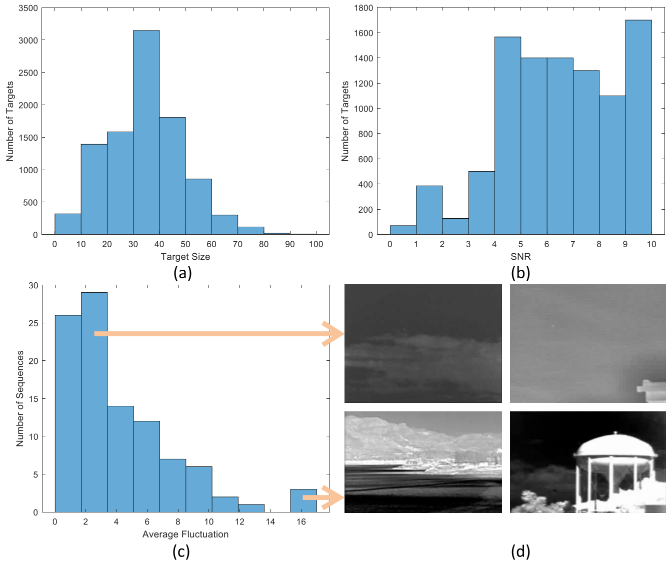


Fig. 4. Distribution of NUDT-MIRSDT. (a) Distribution of target size. (b) Distribution of target SNR. (c) Distribution of background average fluctuation. (d) Images in NUDT-MIRSDT.

The distribution of NUDT-MIRSDT is shown in Fig. 4. We divide the test set into two subsets according to their SNR. In this way, we can evaluate the detection performance with respect to different target intensities. The definition of SNR is

$$\mathrm{SNR} = \frac{|m_t - \mu_b|}{\sigma_b} \tag{7}$$

where m_t is the maximum value of the target and μ_b and σ_b represent the mean value and the standard deviation (i.e., fluctuation) of the local background area (11 × 11 area centered around the target), respectively. For the first test subset, there are eight sequences with SNR lower than 3. For the second test subset, there are 12 sequences with SNR ranging from 3 to 10. When the SNR of the target is large, most methods can achieve good detection performance. The performance on the second test subset of NUDT-MIRSDT dataset can reflect their capability under extremely dim targets and highly fluctuant scenes.

E. HPM Loss for Point-Level Supervision

1) Motivation: In MIRST detection, pixel-level annotation requires expensive labeling cost, while point-level annotation saves a lot of annotation efforts. However, point-level supervision can introduce an extreme imbalance of positive and negative samples since it assigns only one positive sample for a target. Considering that most background regions are smooth and locally similar, using the hard example mining method [46], [47], [48], [49] can significantly reduce negative examples in learning and make the network focus on hard and positive examples. To this end, we propose an HPM loss specially designed for IRST detection under point-level supervision, inspired by OHEM [47] and focal loss [50]. Specifically, HPM loss can suppress the negative impact of the blurry annotations through a protection box in the weight matrixes. The HPM loss can select the worth-learning pixels in background automatically for attentional learning. In this article, the worth-learning points mainly include real targets and some hard negative examples except outliers (i.e., incorrectly labeled pixels) with the highest losses. The HPM loss can dilute the losses of negative examples through replacing some hard negative examples with easy ones. The computational process of the HPM loss is shown in Algorithm 1 and introduced in detail as follows.

2) Initial Loss Calculation: To mine hard negative examples, the first step is to calculate the classification initial losses L_{ini} of all samples. Most loss functions can be used in this step, and we use the weighted L_1 loss, i.e.,

$$L_{\mathrm{ini}}(p, y) = w_p \cdot l_1(p, y) \tag{8}$$

where $y \in \{0, 1\}$ specifies the ground-truth class, $p \in [0, 1]$ is the predicted results of the model, and w_p represents the preset weight value, which is aimed at preventing the samples around targets being selected as hard examples.

Considering point-level annotation, most target points are incorrectly labeled as background. In the training process, these points have large losses and can be easily mined as hard negative points for targeted learning. To avoid focusing on incorrect information, we set a protection box centered around

Algorithm 1 Algorithm for HPM Loss

```
Input: The output of the network $F \in \mathbb{R}^{H\times W}$, the label $T \in \mathbb{R}^{H\times W}$.
1: Generate protection boxes according to the label, and produce a weight matrix $W_p$.
2: Compute the $L_1$ loss by Eq. (8).
3: Gain the total hard negative example set $\mathbf{T}$ by Eq. (9).
4: Generate the hard example set $\mathbf{H}$ by random selection from $\mathbf{T}$, $\mathbf{H} \subset \mathbf{T}$.
5: Generate the easy example set $\mathbf{E}$ by random selection from all.
6: Produce the weight matrix $W$ according to $\mathbf{H}$, $\mathbf{E}$ and $T$.
7: Calculate $HPM$ loss by Eq. (12).
Output: $HPM$
```

the labeled target point to reduce the losses of pixels in the box, that is, the significance of w_p. Specifically, w_p values of the pixels in this box are set to 0, and those outside this box are set to 1. In this way, the negative effects of point-level annotation can be suppressed.

3) Selecting Hard and Easy Negative Sets: In this step, the hard negative example set **H** and the easy negative example set **E** are selected from the whole negative set **S**.

To obtain **H**, we search the top u points with the highest losses by sorting L_{ini}. Considering some outliers (e.g., missed targets and target points outside the protection box), we remove the samples with top v highest losses to reduce the probability of those points being selected. Therefore, we generate the total hard negative examples **T** defined by (9). Then, we randomly select b_{hn} samples from **T** as the final hard negative example set **H** [i.e., (10)]

$$\mathbf{T} = \{\mathbf{S}(\text{top}(L_{\text{ini}}, u))\} - \{\mathbf{S}(\text{top}(L_{\text{ini}}, v))\} \tag{9}$$

$$\mathbf{H} = \text{Random}(\mathbf{T}, b_{hn}). \tag{10}$$

Moreover, some easy negative samples are selected to participate in backpropagation, to dilute the total loss of negative examples while ensuring enough negative examples in training, and to improve the robustness. We randomly select b_{en} samples from the whole negative samples as the easy negative example set **E**

$$\mathbf{E} = \text{Random}(\mathbf{S}, b_{\text{en}}). \tag{11}$$

4) Calculating HPM Loss: According to **H** and **E**, the weight matrix W used in the final loss function is generated to weight the losses of the samples involved in backpropagation. Then, the final HPM loss is calculated as

$$\text{HPM}(p_t) = \sum -\alpha W(1-p_t)^{\gamma}\log(p_t) \tag{12}$$

$$p_t = \begin{cases} p, & \text{if } y = 1 \\ 1-p, & \text{otherwise} \end{cases} \tag{13}$$

where $\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij} = b_{hn} + b_{\text{en}} + b_p$, $w_{ij} \in W$, and b_p represents the number of positive samples. $\alpha \in [0, 1]$ is a weight factor for class 1 and $1-\alpha$ for class 0. Through adjusting the ratio of different types of points, we can alleviate the imbalance between foreground and background samples, as well as mine hard examples in point-level supervision.

Similar to w_p, W also has a protection box centered around the labeled target point, that is, the weights of all samples in the box except the labeled target point are 0. Therefore, it can further suppress wrong learning of the blurred information in the box, which is caused by point-level supervision, through setting the losses of the residual points inside the box to 0 during training.

IV. Experiments

A. Evaluation Metrics

In this article, we adopt probability of detection (P_d) [5], false alarm rate (F_a) [5], and area under curve (AUC) [51], [52] to evaluate the detection performance. Meanwhile, we modify the calculation method (i.e., shooting rules) of the three metrics to adapt to the MIRST detection task.

The definitions of three metrics are described as follows.

1) Probability of detection [5] (i.e., recall rate) is a target-level evaluation metric. It evaluates the capability of algorithms to find targets and locate targets accurately. P_d is defined as

$$P_d = \frac{T_{\text{TP}}}{T_{\text{All}}} \times 100\% \tag{14}$$

where T_{TP} and T_{All} represent the number of correctly predicted targets and all targets in label, respectively.

2) False alarm rate [5] is a pixel-level evaluation metric. It evaluates the capability of algorithms to suppress false alarms. F_a is defined as

$$F_a = \frac{P_{\text{FP}}}{\sum_{i=1}^{N} H_i \times W_i} \tag{15}$$

where P_{FP} represents the number of all pixels falsely predicted as targets, $H_i \times W_i$ is the size of the ith input image, and N is the number of test images.

3) AUC is the area value under the receiver operating characteristics (ROC) curve [51], [52], which is a comprehensive metric to evaluate the detection performance. The AUC value is within the range [0, 1]. A larger AUC value represents better target detection performance.

In the process of evaluation, the criterion of judging whether a prediction is correct or incorrect is important. The existing works widely use the distance between the centroids of the ground truth and the predicted result as the criterion [5], [12]. However, in IRST detection, most targets are smaller than 9 × 9 [9], [29] with the radius smaller than 4, and thus, a large threshold of this distance is unsuitable. A small threshold is also unsuitable since the centroid is usually inaccurate without pixel-level annotation, and this deviation can cause misjudgment. Therefore, we propose a new fine-grained criterion to fit the IRST detection task with point-level, pixel-level, or bounding-box annotations. This criterion, termed as shooting rules, is used to calculate T_{TP} in P_d and P_{FP} in F_a. The details are shown in Fig. 5.

First, we generate the centroid of the ground truth according to the annotation. The point label, the center of bounding-box label, and the centroid of mask label are regarded as

TABLE I

P_d ($\times10^{-2}$), F_a ($\times10^{-5}$), AUC, Param (M), and FPS4 (Frames/Second) Values Achieved by Different State-of-the-Art Methods on the NUDT-MIRSDT Dataset. The Best Results Are in Red, and the Second Best Results Are in Blue. The Worst Results Are in Bold. S-Frame and M-Frame Refer to Single-Frame and Multiframe Methods, Respectively

Methods			NUDT-MIRSDT ($SNR \leq 3$)			NUDT-MIRSDT ($SNR > 3$)			NUDT-MIRSDT (all)			Params(M)	FPS
			P_d	F_a	AUC	P_d	F_a	AUC	P_d	F_a	AUC		
Traditional Method	*S-Frame*	Tophat [11]	2.079	538.038	0.9202	20.583	4.538	0.9418	14.922	216.652	0.9351	-	51.56
		NRAM [13]	0.189	15.805	0.5005	42.333	6.380	0.7133	29.44	10.127	0.6482	-	0.78
		PSTNN [55]	2.268	11.516	0.5127	39.167	8.672	0.6966	27.877	9.803	0.6404	-	2.64
		WSLCM [56]	0.000	323.855	0.5010	44.417	0.946	0.9674	30.827	129.331	0.8242	-	0.26
	M-Frame	MSLSTIPT [29]	4.159	21.703	0.8953	25.500	11.190	0.9602	18.971	15.370	0.9404	-	0.17
		ASTTV-NTLA [30]	0.945	9.220	0.5043	3.750	0.127	0.5181	2.892	3.742	0.5139	-	0.40
		IMNN-LWEC [57]	0.000	7.220	0.5060	38.083	13.058	0.7472	26.431	10.737	0.6734	-	0.31
		SRSTT [58]	62.949	2.874	0.9991	98.167	0.773	0.9988	87.392	1.608	0.9989	-	**0.06**
Deep-Learning Method	*S-Frame*	ALCNet [19]	3.970	37.101	0.6743	74.000	17.845	0.9329	52.574	25.501	0.8435	0.771	77.185
		Res-UNet [59]	15.827	30.320	0.8134	83.833	47.770	0.9758	63.274	40.832	0.9198	0.228	77.060
		DNANet [5]	23.741	19.225	0.8161	86.083	12.329	0.9310	67.380	15.071	0.8843	1.134	12.169
		ISNet [27]	17.958	8.530	0.7750	87.167	26.300	0.9879	65.992	19.235	0.9123	0.967	25.143
		UIUNet [16]	15.123	17.460	0.8456	81.583	12.408	0.9951	61.249	14.417	0.9436	50.541	22.754
	M-Frame	ALCNet +***DTUM***	56.144	0.931	0.9489	99.500	2.370	0.9988	86.235	1.798	0.9818	0.842	7.573
		Res-UNet+***DTUM***	91.682	2.369	0.9921	100	3.415	0.9988	97.455	2.999	0.9967	0.298	7.441
		DNANet +***DTUM***	85.444	1.118	0.9882	99.833	3.611	0.9988	95.431	2.620	0.9951	1.205	4.603
		ISNet +***DTUM***	50.662	0.646	0.9482	99.750	3.448	0.9988	84.731	2.334	0.9816	1.038	5.639
		UIUNet +***DTUM***	72.023	1.916	0.9933	99.833	3.578	0.9988	91.324	2.917	0.9972	50.609	5.214

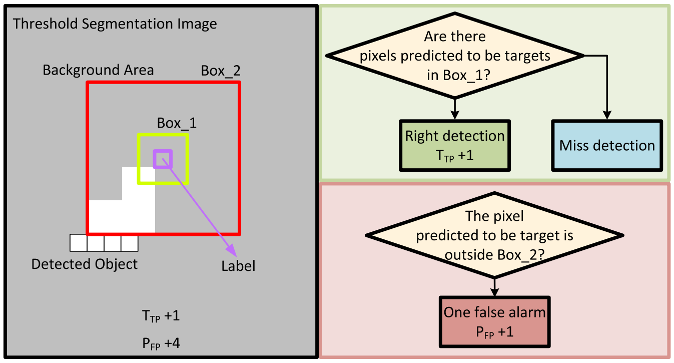

Threshold Segmentation Image
Background Area
Box_2
Box_1
Detected Object
Label
T_{TP} +1
P_{FP} +4
Are there pixels predicted to be targets in Box_1?
Right detection T_{TP} +1
Miss detection
The pixel predicted to be target is outside Box_2?
One false alarm P_{FP} +1

Fig. 5. Shooting rules. The top right of the figure shows rules about T_{TP}. If there are pixels predicted as the target in Box_1, this target is detected right, i.e., $T_{\text{TP}} + 1$. The bottom right shows rules about P_{FP}. Any pixel predicted as target outside Boxes_2 of all targets is regarded as a false alarm, i.e., $P_{\text{FP}} + 1$.

the centroid of the ground truth used in the rules. Then, we find the 3 × 3 region (Box_1) centered at this point. A predicted target having pixels inside Box_1 is considered to be correctly detected. Meanwhile, we find the 9 × 9 region (Box_2) centered at this point. All positively predicted pixels outside Box_2 are considered to be misclassified. According to these rules, T_{TP} and P_{FP} can be calculated. This fine-grained pixel-by-pixel judgment can well fit IRST detection by fully considering the response of the core target region and the response of the area outside of target.

B. Implementation Details

All deep learning-based methods in this article are implemented in PyTorch on a computer with an Intel Xeon Gold 6328H CPU @ 2.80 GHz and two Tesla V100s PCIe 32-GB GPUs. In detail, the networks were trained for 20 epochs with Adam optimizer [42] with an initial learning rate of 0.001. The learning rate is decayed after each epoch with a decaying rate of 0.5. We initialize the weights of convolution layers using the Kaiming method [53] and use random initialization with uniform distribution to initialize the bias. The input of our method is five consecutive frames and the size of each frame is 512 × 512. The ratio (b_p:b_{hn}:b_{en}) in HPM loss is set to 1:10:30 [54].

C. Comparison With State-of-the-Arts

To demonstrate the effectiveness of our method, we compare our method with state-of-the-art IRST detection methods, including traditional SIRST detection methods (Top-hat [11], NRAM [13], PSTNN [55], and WSLCM [56]), traditional MIRST detection methods (MSLSTIPT [29], ASTTV-NTLA [30], IMNN-LWEC [57], and SRSTT [58]), and deep learning-based SIRST detection methods (ALCNet [19], Res-UNet [59], DNANet [5], ISNet [27], and UIUNet [16]) on the NUDT-MIRSDT dataset. All traditional methods were implemented with their default parameters. In the implementation of our method, we adopt the same loss as in the single-frame network. Besides, we only adjust the output channels of the spatial architectures (from 1 to 32) to maintain enough spatial information of targets. We adopt 0 as threshold for all deep learning-based methods.

1) Quantitative Results: We test different methods in the cases of both SNR ≤ 3 and SNR > 3. The results are shown in Table I and ROC curves are shown in Fig. 6. It can be observed that our method achieves the best performance in terms of both P_d and F_a. For the traditional methods, MIRST detection methods achieve better performance with longer inference time in suppressing false alarms than SIRST detection methods. It is worth noting that SRSTT [58] achieves the highest P_d and lowest F_a compared to all traditional methods and deep learning-based SIRST detection methods. However, its inference time is the longest because it needs to calculate 30 frames for one output. ASTTV-NTLA [30] is specially designed for low-rank scenes, resulting in worst performance of compared methods on the NUDT-MIRSDT dataset. Similarly, the traditional methods with manually designed features easily suffer from performance decrease when the scenes

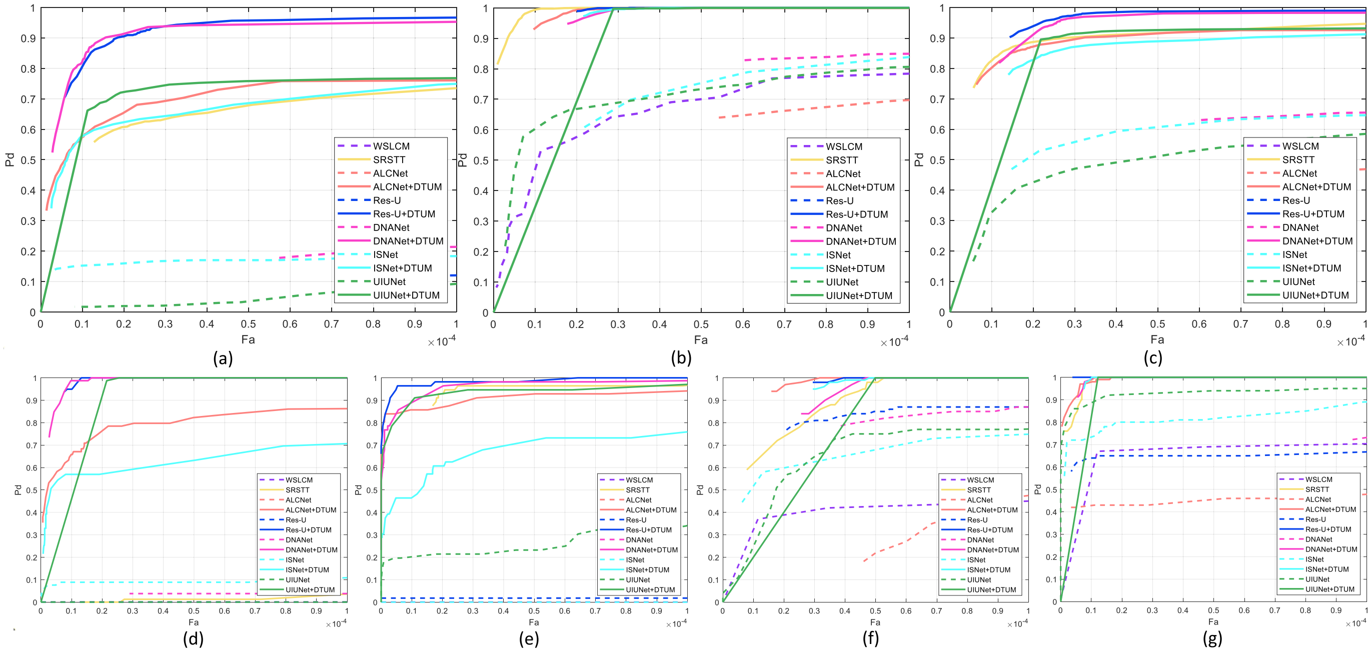


Fig. 6. ROC performances of different methods on the NUDT-MIRSDT dataset. (a) NUDT-MIRSDT (SNR $\leq$ 3). (b) NUDT-MIRSDT (SNR $>$ 3). (c) NUDT-MIRSDT (all). (d) Sequence 1 (SNR $\leq$ 3). (e) Sequence 4 (SNR $\leq$ 3). (f) Sequence 10 (SNR $>$ 3). (g) Sequence 11 (SNR $>$ 3).

change dramatically. That is why the traditional methods usually perform worse than deep learning-based methods.

Moreover, the multiframe methods achieve better performance under those scenes with high fluctuation. The targets and clutters are similar in the spatial domain so that it is hard to classify them from single frame. According to the motion characteristics of the target, many false alarms can be eliminated via multiframe features accumulation. As shown in Table I, our method can introduce significant performance gain (around 30% increases of P_d and obvious decreases of F_a) when compared with all the five SIRST detection methods on the NUDT-MIRSDT dataset. Especially on the subset with SNR $\leq$ 3, the increases of P_d are mostly more than 50%. Fig. 6 also shows that our method achieves significant improvement in terms of AUC. Note that, our DTUM only introduces 0.07 M additional parameters to this SIRST detection network. Benefited by the direction coding of DCCB in DTUM, the position of the target in each frame is explicitly fused into the features. After the temporal convolution, the motion features are well extracted and effectively suppress the clutters and enhance the targets.

2) Qualitative Results: As shown in Fig. 7, we show the output images of different detection methods on seven sequences from two NUDT-MIRSDT subsets. Fig. 7(a1)–(a3) shows the outputs of methods on the subset with SNR $\leq$ 3, and Fig. 7(b1)–(b4) shows the outputs of methods on the subset with SNR $>$ 3.

As shown in Fig. 7, the targets in the subset with SNR $\leq$ 3 are so weak that even an expert cannot find the target accurately from the single frame. All compared single-frame methods work badly on this subset. Only some multiframe methods (e.g., SRSTT, Res-UNet+DTUM, and DNANet+DTUM) can detect some targets by using their temporal salience. Compared to the top performing, our method achieves better performance on suppressing false alarms while maintaining a high detection rate. Moreover, different from other deep learning-based methods, our method can fuse spatial and temporal information and enhance the small and dim targets by extracting their motion information with the help of our DCCB.

D. Ablation Study

To demonstrate the effectiveness of DCCB and find the possibly optimal configuration of DTUM, we conduct our ablation study on the subset with SNR $\leq$ 3. Specifically, we change the depth of DTUM, the number of DCCBs used in DTUM, and some operations in DCCB, and evaluate the detection performance.

1) Depth of DTUM: The depth of DTUM is related to the velocity of the target on the image. Due to the long imaging distance, small targets only move a few pixels in five consecutive frames, and the module does not rely on deep structure to capture the motion information. Therefore, we change the number of direction-coded max-pooling layers in DTUM from 1 to 3 (i.e., the downsampling multiple from 2 to 8), to investigate its influence on MIRST detection. The results are shown in Table II.

The experimental results show that the DTUM with one direction-coded max-pooling layer performs much worse than that with two or three layers. This is because a single downsampling operation could introduce a limited receptive field and thus make it hard to capture the whole change of the motion direction among the input frames. In the end, DTUM with single downsampling cannot achieve the motion-to-data mapping and thus distinguish the targets and clutters. When comparing the DTUMs with two and three direction-coded max-pooling layers, they all achieve good

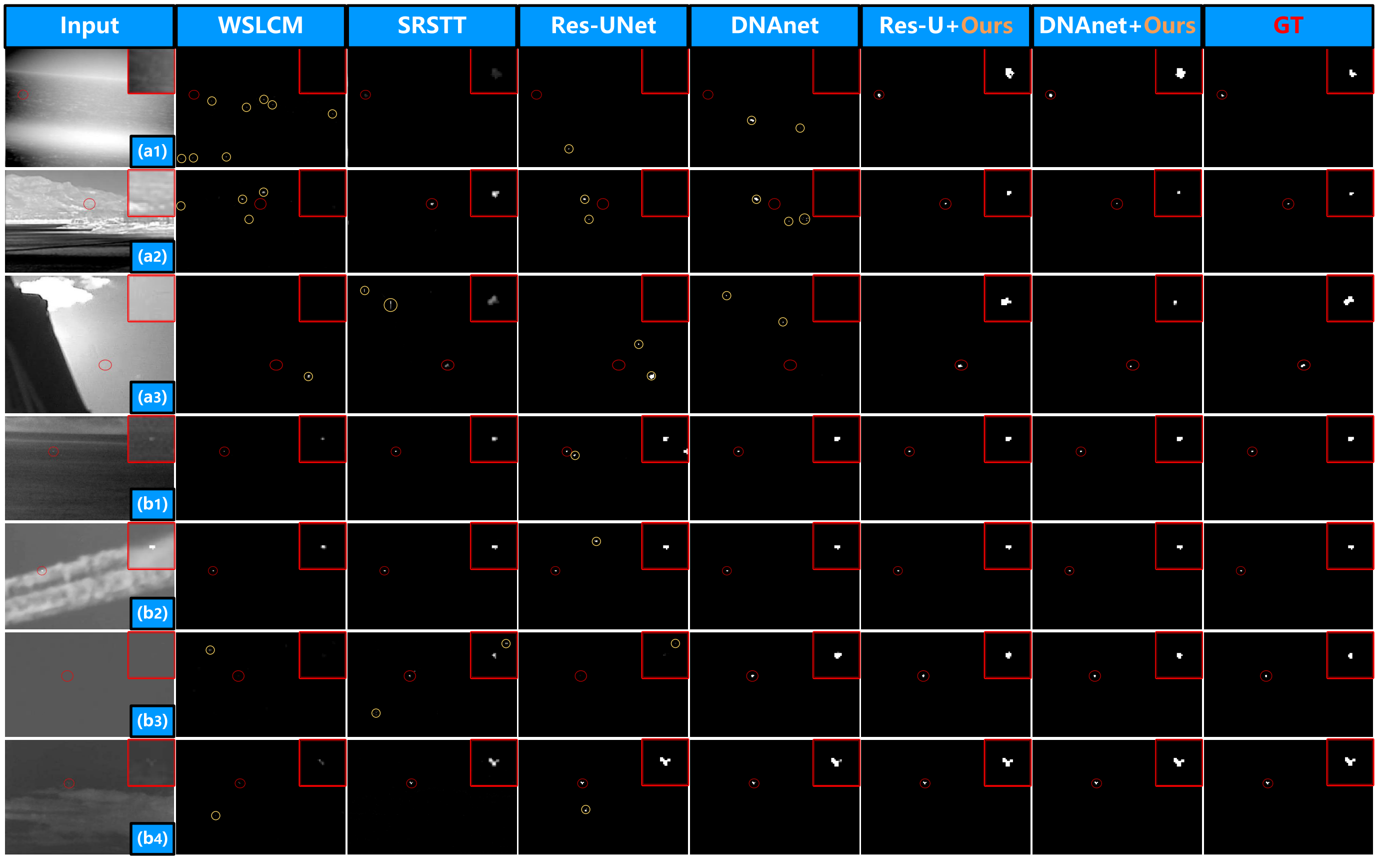


Fig. 7. Visualized results of different methods. (a1)–(a3) Results on the NUDT-MIRSDT (SNR $\leq$ 3) subset. (b1)–(b4) Results on the NUDT-MIRSDT (SNR $>$ 3) subset. For better visualization, the target area is enlarged in the top-right corner and highlighted by a red circle. The false alarm area is marked by a yellow circle. Our method achieves a higher detection rate and lower false alarm rate.

TABLE II

RESULTS OF DTUMS WITH DIFFERENT DEPTHS ON THE NUDT-MIRSDT (SNR $\leq$ 3) SUBSET. THE BEST RESULTS ARE IN RED. THE VALUE IN PARENTHESES REPRESENTS THE DOWNSAMPLING MULTIPLE

Architectures	NUDT-MIRSDT ($SNR \leq 3$)		
	$P_d(\%)$	$F_a(\times 10^{-5})$	AUC
ALCNet+DTUM (2)	13.043	15.493	0.8245
ALCNet+DTUM (4)	53.875	0.764	0.9577
ALCNet+DTUM (8)	56.144	0.631	0.9489
Res-UNet+DTUM (2)	16.068	1.272	0.7544
Res-UNet+DTUM (4)	92.817	1.492	0.9848
Res-UNet+DTUM (8)	91.682	2.369	0.9921

performance in detecting the targets with small SNR (i.e., $\leq$3) and suppressing false alarms. This phenomenon shows that two or three direction-coded max-pooling layers are enough for MIRST detection to extract the motion information in five consecutive frames, and more layers cannot improve the detection performance significantly. Moreover, if the input frames cover a large movement of the target, deeper temporal module is necessary.

2) Configuration of DTUM: We replace the general convolution layer with DCCB in the first half of DTUM. Since there are four levels of convolution in the architecture of DTUM, we test five configurations, as shown in Table III. Four letters in architecture denote the first four convolution layers in DTUM.

TABLE III

RESULTS OF SOME CONFIGURATIONS OF DTUM ON THE NUDT-MIRSDT (SNR $\leq$ 3) SUBSET. THE BEST RESULTS ARE IN RED. “D” AND “G” INDICATE DCCB AND GENERAL CONVOLUTIONAL LAYER, RESPECTIVELY

Architectures	NUDT-MIRSDT ($SNR \leq 3$)		
	$P_d(\%)$	$F_a(\times 10^{-5})$	AUC
ALCNet+DTUM (GGGG)	46.403	1.331	0.9217
ALCNet+DTUM (GDGG)	52.698	1.517	0.9145
ALCNet+DTUM (GDDG)	56.115	6.762	0.9120
ALCNet+DTUM (GDDD)	56.144	0.931	0.9489
Res-UNet+DTUM (GGGG)	85.072	1.430	0.9621
Res-UNet+DTUM (GDGG)	88.489	1.620	0.9533
Res-UNet+DTUM (GDDG)	87.950	1.519	0.9892
Res-UNet+DTUM (GDDD)	91.682	2.369	0.9921
Res-UNet+DTUM (DDDD)	51.607	1.863	0.9426
Res-UNet+DTUM (5 × 3 × 3 Conv)	72.968	0.958	0.9890

Besides, to verify that the general 3-D convolution is insensitive to IRST motions, we replace all DCCBs and Conv3d layers in DTUM with 5 × 3 × 3 convolution layers. The results are shown in Table III.

Compared to the architectures with multiple DCCBs, the performance improvement of the architecture with one DCCB is insignificant (AUC is lower). This is because one DCCB cannot well capture the motion among five frames. We also find that more DCCBs ($\geq$2) in DTUM do not introduce significantly continuous performance improvement since the movement among five consecutive frames is small. After

TABLE IV

Results of Some Configurations of DCCB in Res-UNet+DTUM on the NUDT-MIRSDT (SNR $\leq$ 3) Subset. The Best Results Are in Red

Operations in DCCB					NUDT-MIRSDT ($SNR \leq 3$)		
Abs	Index	Dot	Cat	Sum	$P_d(\%)$	$F_a(\times 10^{-5})$	AUC
✓	✓	✓			91.682	2.369	0.9921
✓	✓		✓		85.633	1.378	0.9831
✓	✓			✓	84.499	1.167	0.9836
✗	✓	✓			89.414	1.186	0.9834
✗	✗	✓			85.066	1.089	0.9714
✗	✗		✓		85.822	1.264	0.9764

several DCCBs, the targets in different frames are processed to the same position, and thus, there is no target movement among the features of multiple frames. Therefore, the motion information can be fully extracted with a small number of DCCBs.

As shown in Table IV, the results of the architecture with all 5 × 3 × 3 convolution layers are far worse than those of the "GDDD" architecture, and even worse than those with all 5 × 1 × 1 convolution layers. This illustrates that the general 3-D convolution cannot well extract IRST motion characteristics due to the fixed weights, although it has more parameters.

Compared to DTUM ("GDDD") without DCCB, DTUM with DCCB achieves better performance in terms of P_d and F_a. Due to the direction-coding mechanism, DCCB helps the network extract motion information by exploring the relationship of the motion direction in different frames. This information can significantly improve the detection performance. To verify the effectiveness of DCCB, we also visualize the intermediate feature maps of different architectures, as shown in Fig. 8. With the single frame (in Fig. 8), an expert cannot well distinguish the target from the background, but our method can. Besides, we find that after removing DCCBs in DTUM, the network can no longer enhance the target when suppressing the background. These experimental results illustrate that our DCCB is very sensitive to the motion of targets.

Moreover, the performance of the architecture with four DCCBs is the worst among five configurations. As shown in the first row of Fig. 8, there is no visible response on the target area but much response on the edge of the cloud in the output features of Res-UNet. Applying DCCB directly on these features without downsampling can cause both targets and clutters to be maintained and suppressed indiscriminately since the motion information cannot be captured with a 5 × 1 × 1 convolution. Meanwhile, this operation can even harm the original spatial features of different frames irreversibly. Therefore, DCCB is not suitable to deal with the spatial features as the first convolution layer in DTUM.

3) Operations in DCCB: DCCB is designed to implement the motion-to-data mapping and make the network sensitive to different motion patterns by introducing the specific mapping index matrix and merging the direction information into the features. To investigate the effectiveness of our DCCB, we replace the important operations (i.e., direction combination, absolute value operation, and direction index) in DCCB with other similar operations. The results are shown in Table IV.

1) *Direction Combination:* It is used to merge the direction information into the pooling features. We combine the pooling results and index results by the elementwise multiplication (Dot). To investigate the effectiveness of this operation, we replace this operation with concatenation (Cat) and elementwise addition (Sum).
2) *Absolute Value Operation:* Equation (5) is designed to extract the motion information with the direction features. In this ablation, we delete its absolute value operation but keep others the same.
3) *Mapping Index Matrix:* Mapping index matrix is used to index the relative position of the target. In this ablation, we replace the mapping index matrix ($[-1.5, -0.5; 0.5, 1.5]$) with the original index matrix ($[0, 1; 2, 3]$). Since all the elements of the original matrix are positive, the absolute value operation is deleted in the meantime for a fair comparison.

As shown in Table IV, encoding direction information with the elementwise multiplication achieves the best detection performance. The other combination methods suffer decreases of 6.049%/7.183% and 0.009/0.0085 in terms of P_d and AUC. This is because the fixed-weight weighted summation of direction indices and features cannot retain both information and exploit the direction information.

Elementwise multiplication can not only preserve the response of targets and background on the feature map but also distinguish different directions according to the sign of the value. Comparing the results of the other combination methods with the results of Res-UNet+DTUM (GGGG) in Table III, those results are similar. This illustrates that concatenation and elementwise addition operations cannot merge the direction information into the features effectively but incorporate with noise.

After deleting the absolute value operation, the performance suffers decreases of 2.268% and 0.0087 in terms of P_d and AUC. This demonstrates the necessity of the absolute value operation since the mapping value of a target's motion can be negative. As shown in the last two rows of Table IV, replacing the mapping index matrix with the original index matrix suffers decreases of 6.616%/5.860% and 0.0207/0.0157 in terms of P_d and AUC. This clearly demonstrates the superiority of motion-to-data mapping in MIRST detection.

E. Supervision in MIRST Detection

In this section, we explore the effects of spatial deep supervision (SDS) and point-level supervision on the performance of IRST detection.

1) Spatial Deep Supervision: Some deep learning-based SIRST detection methods (e.g., DNANet [5] and UIUNet [16]) use SDS to train the networks. There are two types of SDS methods. In DNANet, the deeply supervised features are generated through the equal number downsampling and upsampling operations besides convolution layers. In UIUNet, the deeply supervised features are generated without upsampling. We train the DNANet+DTUM and UIUNet+DTUM

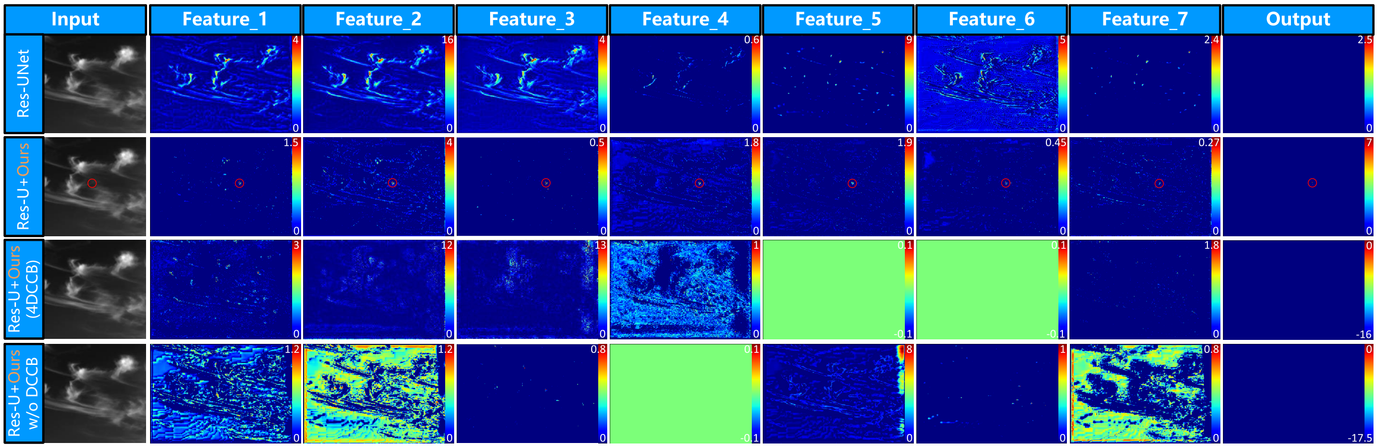


Fig. 8. Intermediate feature maps of Res-UNet+DTUM with different configurations. The first row is the output features of the Res-UNet in Res-UNet+DTUM. The second row is the output features of the last Conv2d layer in DTUM. The third row is the output features of the last Conv2d layer in DTUM (DDDD). The last row is the output features of the last Conv2d layer in DTUM (GGGG). The last column is the output results of different configurations. The target is in the red circle.

TABLE V

Results of Networks With or Without SDS on the NUDT-MIRSDT Dataset

Methods	NUDT-MIRSDT (all)		
	$P_d(\%)$	$F_a(\times 10^{-5})$	AUC
DNANet(w/o SDS)+DTUM	95.431	2.620	0.9951
DNANet(with SDS)+DTUM	88.722	11.399	0.9763
UIUNet(w/o SDS)+DTUM	91.324	2.917	0.9972
UIUNet(with SDS)+DTUM	0	0	0.49947

TABLE VI

Results of Different Loss Functions With Point-Level Supervision on the NUDT-MIRSDT Dataset. The Best Results Are in Red, and the Second Best Results Are in Blue. The Worst Results Are in Bold. "T" Indicates the DTUM

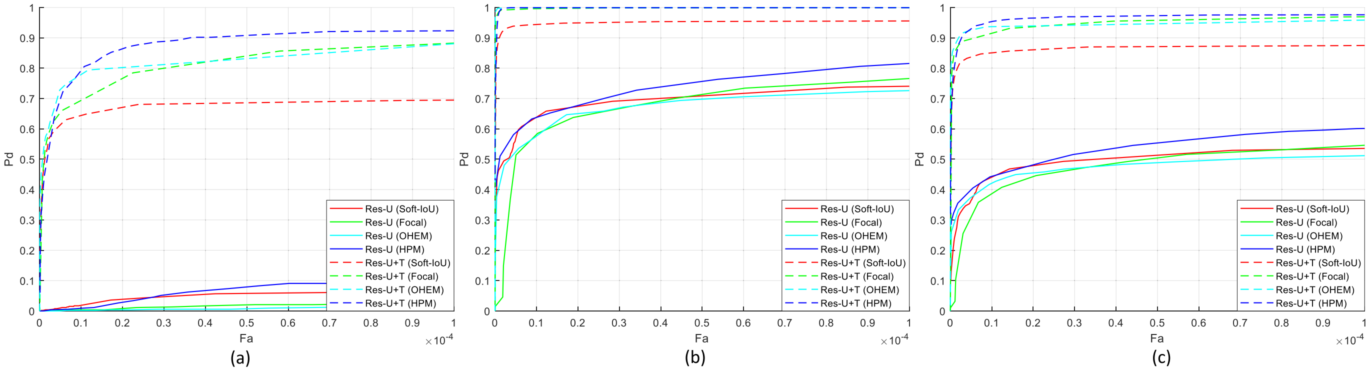
Methods		NUDT-MIRSDT (all) [1,2]		
		$P_d(\%)$	$F_a(\times 10^{-5})$	AUC
Res-U [59]	Soft-IoU Loss [60]	38.115	0.573	0.8430
	Focal Loss [50]	40.717	1.240	0.9361
	OHEM [47]	44.939	1.576	0.7740
	HPM Loss	61.654/63.274	**14.507/40.832**	0.8452/0.9198
Res-U+T	Soft-IoU Loss [60]	70.040	0.044	0.9493
	Focal Loss [50]	70.214	0.016	0.9986
	OHEM [47]	86.755	0.096	0.9962
	HPM Loss	96.298/97.455	**1.703/2.999**	0.9943/0.9967
DNA [5]	Soft-IoU Loss [60]	42.394	0.921	0.8844
	Focal Loss [50]	49.624	1.220	0.9606
	OHEM [47]	56.449	4.291	0.7849
	HPM Loss	65.818/67.380	**16.972/15.071**	0.9408/0.8843
DNA+T	Soft-IoU Loss [60]	69.751	0.053	0.9764
	Focal Loss*[50]	92.019	**1.607**	0.9962
	OHEM [47]	83.748	0.008	0.9923
	HPM Loss	95.662/95.431	1.184/2.620	0.9958/0.9951
UIU [16]	BCE Loss	33.488	0.775	0.9646
	Focal Loss [50]	39.618	0.822	0.9666
	OHEM*[47]	42.799	1.389	0.9159
	HPM Loss	46.906/61.249	**2.037/14.417**	0.9228/0.9436
UIU+T	BCE Loss*	78.138	0.127	0.9905
	Focal Loss*[50]	85.483	0.866	0.99364
	OHEM [47]	78.022	0.045	0.98785
	HPM Loss	86.235/91.324	**1.317/2.917**	0.9901/0.9972

with and without SDS to make the effect of SDS clear, as shown in Table V.

As the results shown, both two SDS methods cause the detection performance to decrease without changing DTUM. With SDS in MIRST detection, the spatial networks become insensitive to the targets. Before the temporal operation, some targets disappear from the features. Actually, the spatial and the temporal parts have different optimization directions. The spatial part should focus on detecting all potential targets, while the temporal part should pay attention to the real targets. A spatial network sensitive to the locally salient area is beneficial to achieve good detection performance in MIRST detection. This is also why Res-UNet performs worse than DNANet and ISNet, but Res-UNet+DTUM performs better than DNANet+DTUM and ISNet+DTUM. Therefore, it is not suitable to use the existing SDS in MIRST detection.

2) Point-Level Supervision: To explore the pointly supervised IRST detection, we replace the original loss functions of different models with focal loss [50], OHEM [47], and our HPM loss. Due to the point-level annotation, many target samples are labeled as background, which are highly likely to be trained as hard background examples in OHEM. To avoid this situation and make a fair comparison, the same protection box in the HPM loss is used in OHEM to prevent those target samples included in loss computation. We train models with different losses on the NUDT-MIRSDT dataset with point-level annotations. Results are shown in Table VI, and their ROC curves are shown in Fig. 9.

There are significant increases in terms of P_d and AUC after using our HPM loss function. Comparing the results of OHEM [47] with focal loss [50], the targets have stronger responses after the models trained with OHEM. This confirms the superiority of hard example mining in learning the target characteristics. Besides, as shown in Table VI, the models trained with other loss functions all have both low P_d and F_a, and some have to set very low segmentation thresholds for the final prediction. This illustrates that all samples have weak responses through those models since the point-level supervision contains limited information. However, the model trained with our HPM loss can generate strong target responses. This is because the HPM loss is specially designed according to the characteristics of IRST detection and point-level supervision.



(a)

(b)

(c)

Fig. 9. ROC performances of methods trained with different loss functions and point annotations on the NUDT-MIRSDT dataset. The blue curve is higher than other curves with the same line type, which illustrates that using our HPM loss can enhance the learning of targets and achieve better performance. (a) NUDT-MIRSDT (SNR $\geq$ 3). (b) NUDT-MIRSDT (SNR $>$ 3). (c) NUDT-MIRSDT (all).

The HPM loss can remove some outliers without annotations and with wrong annotations from the loss computation since they have such high losses that they may lead the network to learn wrong information.

It is worth noting that the detection performance of network trained with HPM loss and point-level supervision is close to the performance under full supervision. With the HPM loss function, the P_d values of most methods can exceed 90% or even 100% (e.g., DNANet+DTUM) of those fully supervised methods, and the F_a values are significantly reduced according to the hard mining mechanism. This illustrates that hard mining is very effective in IRST detection with point-level supervision. It helps the network to suppress false alarms better and capture the informative region of targets with extremely restricted supervision.

V. Conclusion

In this article, we propose a simple yet effective DTUM to achieve MIRST detection by using prior knowledge of target motion. Our proposed DTUM can be equipped with most SIRST detection networks to improve the detection performance, especially on scenes with extremely dim targets. To comprehensively evaluate the performance of MIRST detection methods, we develop a multiframe infrared small and dim target dataset (i.e., NUDT-MIRSDT), and the targets in it come in different sizes, trajectories, and SNR values. The experimental results on the dataset demonstrate the superiority of our method compared to other state-of-the-art methods. Finally, we explore the point-level supervision on the NUDT-MIRSDT dataset and propose the hard mining point (HPM) loss function for network training. The performance of the network trained with the HPM loss under point-level supervision can exceed 90% or even 100% of the performance of fully supervised methods.

References

[1] H. Wang, J. Peng, X. Zheng, and S. Yue, "A robust visual system for small target motion detection against cluttered moving backgrounds," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 3, pp. 839–853, Mar. 2020.

[2] H. Wang, H. Wang, J. Zhao, C. Hu, J. Peng, and S. Yue, "A time-delay feedback neural network for discriminating small, fast-moving targets in complex dynamic environments," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 1, pp. 316–330, Jan. 2023.

[3] M. Teutsch and W. Krüger, "Classification of small boats in infrared images for maritime surveillance," in *Proc. Int. WaterSide Secur. Conf.*, Nov. 2010, pp. 1–7.

[4] Y. Zhu et al., "Tiny object tracking: A large-scale dataset and a baseline," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 2, 2023, doi: 10.1109/TNNLS.2023.3239529.

[5] B. Li et al., "Dense nested attention network for infrared small target detection," *IEEE Trans. Image Process.*, vol. 32, pp. 1745–1758, 2023.

[6] L. Jiao et al., "New generation deep learning for video object detection: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3195–3215, Aug. 2022.

[7] S.-C. Huang and B.-H. Chen, "Highly accurate moving object detection in variable bit rate video-based traffic monitoring systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, pp. 1920–1931, Dec. 2013.

[8] I. S. Reed, R. M. Gagliardi, and H. M. Shao, "Application of three-dimensional filtering to moving target detection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-19, no. 6, pp. 898–905, Nov. 1983.

[9] I. S. Reed, R. M. Gagliardi, and L. B. Stotts, "Optical moving target detection with 3-D matched filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 24, no. 4, pp. 327–336, Jul. 1988.

[10] I. S. Reed, R. M. Gagliardi, and L. B. Stotts, "A recursive moving-target-indication algorithm for optical image sequences," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 3, pp. 434–440, May 1990.

[11] J. Rivest, "Detection of dim targets in digital infrared imagery by morphological image processing," *Opt. Eng.*, vol. 35, no. 7, p. 1886, Jul. 1996.

[12] C. Gao, D. Meng, Y. Yang, Y. Wang, X. Zhou, and A. G. Hauptmann, "Infrared patch-image model for small target detection in a single image," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4996–5009, Dec. 2013.

[13] L. Zhang, L. Peng, T. Zhang, S. Cao, and Z. Peng, "Infrared small target detection via non-convex rank approximation minimization joint $L_{2,1}$ norm," *Remote Sens.*, vol. 10, no. 11, p. 1821, Nov. 2018.

[14] X. Wang, Z. Peng, P. Zhang, and Y. Meng, "Infrared small dim target detection based on local contrast combined with region saliency," *High Power Laser Part. Beams*, vol. 27, no. 9, 2015, Art. no. 091005.

[15] Y. Dai, Y. Wu, F. Zhou, and K. Barnard, "Asymmetric contextual modulation for infrared small target detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 949–958.

[16] X. Wu, D. Hong, and J. Chanussot, "UIU-Net: U-Net in U-Net for infrared small object detection," *IEEE Trans. Image Process.*, vol. 32, pp. 364–376, 2023.

[17] H. Sun, J. Bai, F. Yang, and X. Bai, "Receptive-field and direction induced attention network for infrared dim small target detection with a large-scale dataset IRDST," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5000513.

[18] G. Arce and M. McLoughlin, "Theoretical analysis of the max/median filter," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, no. 1, pp. 60–69, Jan. 1987.

[19] Y. Dai, Y. Wu, F. Zhou, and K. Barnard, "Attentional local contrast networks for infrared small target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 11, pp. 9813–9824, Nov. 2021.

[20] C. Li et al., "ASIF-Net: Attention steered interweave fusion network for RGB-D salient object detection," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 88–100, Jan. 2021.

[21] Z. Chen, R. Cong, Q. Xu, and Q. Huang, "DPANet: Depth potentiality-aware gated attention network for RGB-D salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 7012–7024, 2021.

[22] Z. Chen, Q. Xu, and R. Cong, "Global context-aware progressive aggregation network for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 10599–10606.

[23] Z. Deng et al., "R^{3}Net: Recurrent residual refinement network for saliency detection," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 684–690.

[24] X. Hu, L. Zhu, J. Qin, C.-W. Fu, and P.-A. Heng, "Recurrently aggregating deep features for salient object detection," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2018, vol. 32, no. 1, pp. 1–8.

[25] X. Hu, C.-W. Fu, L. Zhu, T. Wang, and P.-A. Heng, "SAC-Net: Spatial attenuation context for salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 3, pp. 1079–1090, Mar. 2021.

[26] Y. Wei, X. You, and H. Li, "Multiscale patch-based contrast measure for small infrared target detection," *Pattern Recognit.*, vol. 58, pp. 216–226, Oct. 2016.

[27] M. Zhang, R. Zhang, Y. Yang, H. Bai, J. Zhang, and J. Guo, "ISNet: Shape matters for infrared small target detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 867–876.

[28] T. Wu et al., "MTU-Net: Multilevel TransUNet for space-based infrared tiny ship detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5601015.

[29] Y. Sun, J. Yang, and W. An, "Infrared dim and small target detection via multiple subspace learning and spatial–temporal patch-tensor model," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 3737–3752, May 2021.

[30] T. Liu et al., "Nonconvex tensor low-rank approximation for infrared small target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5614718.

[31] C. Xiao et al., "DSFNet: Dynamic and static fusion network for moving object detection in satellite videos," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[32] J. Du et al., "A spatial–temporal feature-based detection framework for infrared dim small target," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 3000412.

[33] X. Sun, L. Guo, W. Zhang, Z. Wang, and Q. Yu, "Small aerial target detection for airborne infrared detection systems using LightGBM and trajectory constraints," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 9959–9973, 2021.

[34] H. Wang, L. Zhou, and L. Wang, "Miss detection vs. False alarm: Adversarial learning for small object segmentation in infrared images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8508–8517.

[35] R. Fu et al., "A dataset for infrared time-sensitive target detection and tracking for air-ground application," *China Sci. Data*, vol. 7, no. 2, pp. 206–221, 2022.

[36] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari, "Training object class detectors with click supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 180–189.

[37] L. Chen, T. Yang, X. Zhang, W. Zhang, and J. Sun, "Points as queries: Weakly semi-supervised object detection by points," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8819–8828.

[38] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, "Instance segmentation with point supervision," 2019, *arXiv:1906.06392*.

[39] B. Cheng, O. Parkhi, and A. Kirillov, "Pointly-supervised instance segmentation," 2021, *arXiv:2104.06404*.

[40] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, "Where are the blobs: Counting by localization with point supervision," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 547–562.

[41] P. Akiva, K. Dana, P. Oudemans, and M. Mars, "Finding berries: Segmentation and counting of cranberries using point supervision and shape priors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 219–228.

[42] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, "What's the point: Semantic segmentation with point supervision," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2016, pp. 549–565.

[43] Y. Liu, M. Shi, Q. Zhao, and X. Wang, "Point in, box out: Beyond counting persons in crowds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6462–6471.

[44] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI*. Berlin, Germany: Springer, 2015, pp. 234–241.

[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.

[46] W. Liu et al., "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2016, pp. 21–37.

[47] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 761–769.

[48] S. Cai, Y. Guo, S. Khan, J. Hu, and G. Wen, "Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8390–8399.

[49] Y. Guo, M. Choi, K. Li, F. Boussaid, and M. Bennamoun, "Soft exemplar highlighting for cross-view image-based geo-localization," *IEEE Trans. Image Process.*, vol. 31, pp. 2094–2105, 2022.

[50] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[51] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[52] H. Deng, X. Sun, M. Liu, C. Ye, and X. Zhou, "Entropy-based window selection for detecting dim and small infrared targets," *Pattern Recognit.*, vol. 61, pp. 66–77, Jan. 2017.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[54] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[55] L. Zhang and Z. Peng, "Infrared small target detection based on partial sum of the tensor nuclear norm," *Remote Sens.*, vol. 11, no. 4, p. 382, Feb. 2019.

[56] J. Han et al., "Infrared small target detection based on the weighted strengthened local contrast measure," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 9, pp. 1670–1674, Sep. 2021.

[57] Y. Luo, X. Li, S. Chen, C. Xia, and L. Zhao, "IMNN-LWEC: A novel infrared small target detection based on spatial–temporal tensor model," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5004022.

[58] J. Li, P. Zhang, L. Zhang, and Z. Zhang, "Sparse regularization-based spatial–temporal twist tensor model for infrared small target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5000417.

[59] X. Xiao, S. Lian, Z. Luo, and S. Li, "Weighted res-UNet for high-quality retina vessel segmentation," in *Proc. 9th Int. Conf. Inf. Technol. Med. Educ. (ITME)*, Oct. 2018, pp. 327–331.

[60] M. A. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *Proc. Int. Symp. Vis. Comput. (ISVC)*. Cham, Switzerland: Springer, 2016, pp. 234–244.

Ruojing Li received the B.E. degree in electronic engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2020, where she is currently pursuing the Ph.D. degree in information and communication engineering.

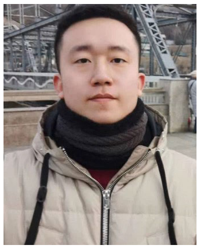
Her research interests include infrared small target detection, particularly on multiframe detection and deep learning.

Wei An received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China, in 1999.

She was a Senior Visiting Scholar with the University of Southampton, Southampton, U.K., in 2016. She is currently a Professor with the College of Electronic Science and Technology, NUDT. She has authored or coauthored over 100 journal and conference publications. Her current research interests include signal processing and image processing.

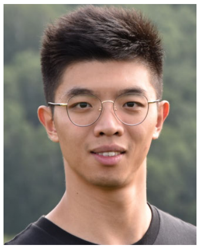
Chao Xiao received the B.E., master's, and Ph.D. degrees from the National University of Defense Technology (NUDT), Changsha, China, in 2016, 2018, and 2023, respectively.

He is currently a Lecturer with the College of Electronic Science and Technology, NUDT. His recent research interests focus on small target detection and tracking in satellite videos.



Boyang Li received the B.E. degree in mechanical design manufacture and automation from Tianjin University, Tianjin, China, in 2017, and the M.S. degree in biomedical engineering from the National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing, China, in 2020. He is currently pursuing the Ph.D. degree in information and communication engineering with the National University of Defense Technology (NUDT), Changsha, China.

His research interests include infrared small target detection, weakly supervised semantic segmentation, and deep learning.

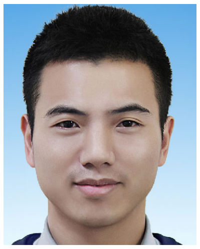


Yingqian Wang (Member, IEEE) received the B.E. degree in electrical engineering from Shandong University, Jinan, China, in 2016, and the master's and Ph.D. degrees in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2018 and 2023, respectively.

He is currently an Assistant Professor with the College of Electronic Science and Technology, NUDT. His research interests focus on optical imaging and detection, particularly on light field imaging, image super-resolution, and infrared small target detection.

Miao Li received the M.E. and Ph.D. degrees from the National University of Defense Technology (NUDT), Changsha, China, in 2012 and 2017, respectively.

He is currently an Associate Professor with the College of Electronic Science and Technology, NUDT. His current research interests include infrared dim and small target detection.



Yulan Guo (Senior Member, IEEE) received the B.E. and Ph.D. degrees from the National University of Defense Technology (NUDT), Changsha, China, in 2008 and 2015, respectively.

He has authored over 150 articles at highly referred journals and conferences. His research interests lie in 3-D vision, low-level vision, and machine learning.

Dr. Guo is a Senior Member of Association for Computing Machinery (ACM). He served as the Area Chair for IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2021 and 2023, IEEE International Conference on Computer Vision (ICCV) 2021, and ACM Multimedia 2021. He also served as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, *IET Computer Vision*, *IET Image Processing*, *Computers & Graphics*, and *The Visual Computer*.