A Degraded Reconstruction Enhancement-based Method for Tiny Ship Detection in Remote Sensing Images with A New Large-scale Dataset

Jianqi Chen, Keyan Chen, Hao Chen, Zhengxia Zou and Zhenwei Shi*, Member, IEEE

Abstract—The rapid detection of ships within the wide sea area is essential for intelligence acquisition. Most modern deep learning-based ship detection methods focus on locating ships in high-resolution (HR) remote sensing (RS) images. Seldom efforts have been made on ship detection in medium-resolution (MR) RS images. An MR image covers a much wider area than an HR one of the same size, thus facilitating quick ship detection. To this end, we propose a tiny ship detection method namely, Degraded Reconstruction Enhancement Network (DRENet), for MR RS images. Different from previous methods that mainly focus on feature fusion strategies to improve the expression ability of the detector, we design an additional network branch, i.e., degraded reconstruction enhancer, to learn to regress an object-aware blurred version of the input image in the training phase. Our intuition is that the proposed reconstruction branch may guide the backbone to focus more on tiny ship targets instead of the vast background. Moreover, we incorporate a CRoss-stage Multi-head Attention module in the detector to further improve the feature discrimination by leveraging the self-attention mechanism. To fill the gap of lacking a large-scale MR ship detection dataset, we introduce Levir-Ship, which contains 3876 GF-1/GF-6 multi-spectral images and over 3K tiny ship instances. Experiments on Levir-Ship validate the effectiveness and efficiency of the proposed method. Our method achieves 82.4 AP with 85 FPS, which outperforms many state-of-the-art ship detection methods. Our code and dataset are available at https://github.com/WindVChen/DRENet.

Index Terms—Convolutional neural network (CNN), ship detection, deep learning, optical image, remote sensing.

I. INTRODUCTION

SHIP detection in optical remote sensing (RS) images refers to locating ships in RS images and giving their positions and sizes automatically. As the main carrier of sea transportation, ship plays a very important role in the military and civilian fields [1]. The accurate and fast detection of ships in RS images has been a hot research topic.

With the rapid development of RS technology in recent years, research on ship detection in optical images is more active because of its content-rich and expression-intuitive features [2]–[6]. Traditional ship detection methods often require complicated hand-craft features extraction, and cannot adapt to changing environments [7], [8]. Nowadays, deep learning (DL) techniques, especially convolutional neural networks (CNN), have been widely applied in RS image ship detection. Compared with traditional methods, DL-based methods can learn robust multi-level features and classifiers in an end-to-end manner. Inspired by the great success of DL-based detectors in the field of computer vision (CV), in recent years, considerable works have introduced DL techniques into RS ship detection tasks [9]–[17], which confirmed better detection performance in terms of accuracy and stability than traditional counterparts.

The existing DL-based ship detection methods are mostly modified based on the object detectors in CV. Recent advances in optical RS ship detection include fusing the ship shape priors [18], extracting features from ship prow and stern [12], and applying the orientation information of ships [19]. However, most of these methods focus on high-resolution (HR) images, thus benefiting from the rich textures and clear edges.

Despite the great success in ship detection in HR RS images [19]–[21], seldom efforts have been made in ship detection in relatively lower resolution RS images, e.g., medium-resolution (MR, about 16m/pixel) RS images.1 Considering the urgent need for monitoring and early warning over a wide area of sea in practical applications [22], [23], ship detection in MR images is critically important. We argue that MR RS images are more suitable than HR ones for quick ship detection in vast sea areas, because an MR image can cover a much wider area than an HR image of the same size. For instance, to detect ships in a fixed sea area, supposed we need 1 hour in 16m/pixel MR images (GF-1), then it means we need 256 hours, about 10 days, to finish the same task in 1m/pixel HR images (SkySat-1). The time cost is apparently unbearable, especially in such a big data era.

Ship detection in real-world MR RS images has several challenges. First, ship detection in MR RS images can suffer from scarce textures and hazy edges of ships. For example, in a GF-1 image (16m/pixel), a ship may only occupy 20 pixels. Second, in the real scenario, an RS image may be covered by massive fractus clouds. The complex imaging conditions may

1Please refer to the website (https://doc.arcgis.com/en/imagery/workflows/resources/managing-medium-resolution-satellite-data.htm) for detailed definitions of high resolution and medium resolution.
Fig. 1. Samples of 512×512 size from different datasets. (a): Samples from Airbus Ship Detection Competition dataset \(^2\) with a spatial resolution of 2m and higher. (b): Samples from Levir-Ship dataset with a spatial resolution of 16m.

also induce many false positives in a conventional detector. Fig. 1 gives an intuitive comparison of HR images and MR ones. The ships in MR images are difficult to be recognized (zoomed for best view) and also hard to be distinguished from fractus clouds. Most detection networks focus on improving the feature expression ability of the backbone or the neck by feature fusion [11], [24] and cross-stage connection [21], [25]. We argue that such feature enhancing methods are struggling to extract discriminative features of tiny ship targets.

To solve the above problems, we propose a degraded reconstruction enhancement ship detection network called DRENet - Tiny Ship Detection Based on Degraded Reconstruction Enhancement in Remote Sensing Images, which contains an efficient backbone to extract robust feature representations, a degraded reconstruction enhancer to help the backbone learn to distinguish ships from irrelevant backgrounds, and an object detector to locate ships. The pipeline is depicted in Fig. 2. Different from previous methods that mainly focus on feature fusion strategies [11], [24] to improve the expression ability of the detector, we design an additional network branch, i.e., degraded reconstruction enhancer, to learn to regress an object-aware blurred version of the input image in the training phase. In the enhancer, we design an image processing operation called “Selective Degradation” to blur the background. Our intuition is that the proposed reconstruction branch may guide the backbone to focus more on tiny ship targets instead of the vast background (e.g., fractus clouds). Please note that our proposed enhancer is only used in the training phase and is computing-free in the inferencing phase. Moreover, we incorporate a CRoss-stage Multi-head Attention (CRMA) module in the detector to further improve the feature discrimination by leveraging the self-attention mechanism.

To the most of our knowledge, there is still no public dataset for MR RS image ship detection. Therefore, we propose a dataset named “Levir-Ship” to promote our research. Levir-Ship contains 3876 images of \(512 \times 512\) pixels collected from GaoFen-1 and GaoFen-6 satellites with the spatial resolution of 16m/pixel. We conduct experiments on Levir-Ship and the results show the effectiveness and efficiency of our method.

The contribution of our work can be summarized as follows:
- We propose an effective method (DRENet) for efficient tiny ship detection. A degraded reconstruction enhancer is leveraged to guide the backbone to focus more on the target instead of the background, and the cross-stage multi-head attention is introduced to further improve the discrimination ability of the detector.
- We introduce the Levir-Ship detection dataset, consisting of 3876 GF-1/GF-6 images (each size of \(512 \times 512\) pixels with the spatial resolution of 16m/pixel) and more than 3K tiny ship instances.
- Extensive experiments on the Levir-Ship validate the effectiveness and efficiency of the proposed method. Our method achieves 82.4 AP with 85 FPS, and surpasses several state-of-the-art ship detection methods.

The rest of the paper are organized as follows. Related work is introduced in section II. In section III, we give a detailed description of our method. Section IV describes Levir-Ship dataset and the comparison with other ship detection datasets. Experimental results are reported in section V, and the conclusion is drawn in section VI.

II. RELATED WORK
Ship detection has been extensively studied for decades. Generally, traditional methods perform a multi-stage coarse-to-fine process to obtain detection results [28]–[32]. The detection process can often be divided into two steps - region proposal and region refining, and make use of hand-craft features. Although achieving some good results in certain scenarios, these methods are vulnerable to complex situations, together with the limitation of heavy reliance on prior and complex operating procedures.

In recent years, benefiting from the development of hardware and big data, many effective DL object detection methods
have been proposed [33]. They are more flexible, unified, and strong to deal with the task, and can be divided into two-stage methods and one-stage methods. Two-stage methods are mainly based on the RCNN series [34]–[37]. They often play the idea of obtaining candidate bounding boxes first, and then classifying and regressioning the results from the candidate. Comparing with two-stage methods, one-stage methods headed by Yolo [27], [38]–[41], SSD [42], RetinaNet [43], EfficientDet [44] directly give the detection results. They obtain a higher detection speed expending a little accuracy in general. However, both two-stage and one-stage methods rely on the setting of anchor boxes, which needs much prior knowledge and constrains the methods’ generalization. Thus, a series of anchor-free methods are proposed, such as CornerNet [45], CenterNet [46], FCOS [47], etc, which prevent anchor boxes initialization and also achieve accurate detection results.

Motivated by the above approaches, many effective RS image object detection methods have emerged in recent years [48]–[53]. For example, Wang et al. [49] proposed a one-stage method, Zhang et al. [50] proposed an anchor-free method, and Huang et al. [52] proposed an object-adaptation label assignment, which achieve many good results. While more specific to ship target, many methods have made great contributions to ship detection in RS images with DL technology. Li et al. [11] proposed a hierarchical selective filtering layer structure based on Faster-RCNN adapting to different scales of ships. Tian et al. [12] added atrous spatial pyramid pooling on the basis of Mask-RCNN, and used dense layer jump connections to use feature information in the backbone effectively achieving high detection accuracy. Chen et al. [24] introduced dilated convolution and attention mechanism in YOLOv3, and proposed ImYOLOv3. Also, researchers introduce prior information to get high performance. Zhang et al. [18] pre-designed the structure of the candidate object according to the ship structure, and then inputted the proposals into CNN to detect the ship. Wu et al. [20] used the prominence of the ship’s head to detect the ship by positioning the ship’s bow. Tang et al. [54] extracted the ship by using the difference in hue, saturation, and value between the ship and the surrounding. These methods often focus on HR images and utilize the features extracted by clear edges and rich textures to achieve high performance. However, when tackling the tiny ship detection problem, they may perform badly. In the paper, we propose DRENet to address tiny ship detection.

### III. METHODOLOGY

The DRENet, shown in Fig. 2, consists of an efficient backbone, a Degraded Reconstruction Enhancer (DRE), and a ship detector. We follow YOLOv5s (a small version of YOLOv5) [27] as the backbone to efficiently extract multi-scale image features. The degraded reconstruction enhancer is designed to guide the backbone to acquire many discriminative features by regressing an object-aware blurred version of the input image in the training phase. The supervision of the enhancer comes from a “Selective Degradation” operation we propose. In the detector, we introduce a CRoss-stage Multi-head Attention (CRMA) module to better locate ships within the input image. Details will be introduced as follows.

![Fig. 2. The overall structure of DRENet. a) Backbone: A lightweight backbone to enable robust features extraction efficiently. b) Enhancer: The proposed degraded reconstruction enhancer to guide the backbone to extract many discriminative features by regressing degraded images. The degraded images come from the proposed “Selective Degradation” operation. c) Detector: The detector to display the final detection results. We introduce into the detector a CRoss-stage Multi-head Attention (CRMA) module, which can gain a large receptive field to locate ships accurately while saving much computation. (For details of RCAB and SPP, please refer to [26] and [27]. The implementations of upsampling layers are aligned with [26] and [27], with nearest-neighbor interpolation in the detector while sub-pixel convolution in the enhancer.)](image-url)
Algorithm 1 Selective Degradation for obtaining DRE supervision information

Input: $I$, the input image
Input: $G$, all the ground-truth boxes set in $I$
Define $F(\cdot)$, an increasing function to adjust the mean kernel size
Define $Center(\cdot)$, a function to get the box’s central coordinate
Define $Distance(\cdot, \cdot)$, a function to compute the Euclidean Distance between two inputs
Define $EvenInt(\cdot)$, a function to return the nearest even number that is less than the input
Define $Resize(\cdot)$, a function to resize the image by nearest neighbor interpolation

Output: $\hat{I}$, the label of $I$ in DRE

1: for $i = 0; i < H; i + +$ do
2:   for $j = 0; j < W; j + +$ do
3:     $k \leftarrow +\infty$
4:     for each $g \in G$ do
5:       $c \leftarrow Center(g)$
6:       $t \leftarrow Distance((i, j), c)$
7:       if $t < k$ then
8:         $k = t$
9:     end if
10:   end for
11:   $k = F(k)$
12:   $k \leftarrow EvenInt(k)$
13:   $\hat{I}(i, j) = 1 \left| \frac{t}{k + 1} \right| \sum_{m=i-\frac{k}{2}, n=j-\frac{k}{2}}^{m=i+\frac{k}{2}, n=j+\frac{k}{2}} I(m, n)$
14: end for
15: end for
16: $\hat{I} = Resize(\hat{I})$
17: return $\hat{I}$

A. Degraded Reconstruction Enhancer

The degraded reconstruction Enhancer (DRE) utilizes the features extracted by the backbone to reconstruct the degraded input image, which could guide the backbone to pay more attention to ships, keep more information about the objects, and omit the complex background.

1) Supervision of DRE: In this part, we propose a method named “Selective Degradation” to generate labels for the enhancer. Selective Degradation, with the nature of filtering, performs mean filters of different sizes on different positions in original images. Details of Selective Degradation are illustrated in Algorithm 1. Fig. 3 depicts how to get the mean kernel size of a certain position in an image. By Selective Degradation, we can get a “pseudo saliency map” in which ships would be kept completely, and the background is processed to be fuzzy. Through the training of DRE, the backbone will pay more attention to the tiny ships instead of the textures, details, or the fractus clouds of the background.

2) Design of DRE: As is shown in Fig. 2, we take feature maps from the backbone as the input of the enhancer. Here we just utilize the feature maps with the shape of $64 \times 64$ to reconstruct the degraded label with a shape of $128 \times 128$, leveraging the performance and convenience. We refer to the existing super-resolution method [26] to design the structure of DRE, but there are some differences. In [26], as the goal is to improve the quality of the super-resolution image, the network is actually complex and deep. However, instead of reconstructing perfect Selective Degradation labels, our goal is to acquire a discriminative feature representation of the backbone. Therefore, we make a relatively shallow enhancer network, with just a single RCAB (Residual Channel Attention Block, a component module in [26]) used, to concentrate more on the training of the backbone. Based on the mechanism of “Selective Degradation”, the labels of the enhancer focus on the ship target and omit the complex background. Thus, the enhancer provides a tendency to obtain robust feature representations in the backbone, and classify ships more exactly.

Compared with the existing super-resolution methods [26], [55], [56], the proposed DRE focuses more on improving the detection results of tiny ships. The supervision of DRE is an object-aware blurred version of the input image (different from the clear image of existing super-resolution methods), which can guide the backbone to pay more attention to ships and be more robust against the disturbances from complex backgrounds. Further, different from the existing super-resolution methods that use labels of higher image resolution, our proposed DRE reconstructs labels of lower image resolution ($128 \times 128$) than the input image ($512 \times 512$), which saves much computation and memory. The structure of DRE is also much simpler and shallower than that of the existing super-resolution methods.

B. Efficient Detector

To achieve efficient ship detection, our detector is based on YOLOv5s. We have made some modifications for the tiny ship detection task to further improve efficiency.

In order to spread the rich spatial information of the low-level features to the deep-level features at a short and fast path, YOLOv5 adopts the PAN [57] structure. However, when assigning anchors to each output layer in tiny ship object
most of the bounding boxes fall on the shallow detection layers while few on the deep detection layers. It is actually a time-costing operation but retrieving few benefits to enrich the spatial information of the deep detection layer. In DRENet, we remove the PAN path to improve the detection speed.

Considering different output layers of FPN [58] are assigned with an imbalanced number of training samples, we add two “Scale Layers” behind the two Upsampling layers in the detector (depicted in Fig. 2), following the work of [59]. The Scale Layer is actually just a single learnable parameter that can balance the backpropagated gradients from different output layers in the bottom-up pathway layers of FPN. Supposed there are two layers C1, and C2 in the bottom-up pathway from shallow to deep, and two layers P1, and P2 in the top-down pathway corresponded. In the original FPN, \( P_1 = C_1 + \text{upsample}(P_2) \), after we add the scale layer, it turns to be \( P_1 = C_1 + \alpha \times \text{upsample}(P_2) \), where \( \alpha \) is a learnable parameter.

To achieve a full-image receptive field, we propose a Cross-stage Multi-head Attention (CRMA) module, and replace the CSP-like layers in the bottleneck of YOLOv5 with our CRMA in YOLOv5’s bottleneck, which like layers with our CRMA in YOLOv5’s bottleneck, which can gain much reduction of computation and complexity of the final output. In our work, we argue the many convolutions in the branch are low cost-effective and replace them with the Multi-Head Self-Attention (MHSA) layer designed in [61], which can enrich the network’s receptive field much while keeping low parameters and complexity, and also more suitable for vision task (with relative positional encodings used and Q, K, V in 2D style) compared with the original version in [62]. By fusing CSPNet’s idea and the MHSA layer, we construct the CRMA module. More details of CRMA are shown in Fig. 2 and TABLE I. Also note that we replace all the five CSP-like layers with our CRMA in YOLOv5’s bottleneck, which can gain much reduction of computation and complexity of the network while keeping good performance.

### C. Loss Function

The loss function of DRENet is composed of two parts, including the enhancer loss and the detector loss.

1) **Enhancer Loss**: We apply mean square error (MSE) to form the reconstruction error of the enhancer. The loss function is formulated as follows:

\[
\text{Loss}_{\text{enhancer}} = \frac{1}{N} \sum (y^* - y)^2
\]

(1)

where \( y^* \) represents the pixel value of the reconstruction result, \( y \) represents the pixel value of the “Selective Degradation” label, and \( N \) represents the number of total pixels.

2) **Detector Loss**: The detector loss function contains regression loss and classification loss. We conduct typical loss in object detection task - CIoULoss [63] for regression error, and Binary Cross-Entropy Loss for classification. In DRENet, they are defined as:

\[
\text{Loss}_{\text{reg}} = 1 - \text{CIoU}
\]

(2)

\[
\text{CIoU} = \text{IOU} - \frac{\rho^2(b, b^{gt})}{c^2} - \frac{\nu^2}{1 - \text{IOU} + \nu}
\]

(3)

\[
\nu = \frac{4}{\pi^2} (\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2
\]

(4)

where \( b \) and \( b^{gt} \) denote the central points of predicted box and target box, \( \rho(\cdot) \) denotes the Euclidean distance, \( c \) is the diagonal length of the smallest enclosing box covering the two boxes, \( \nu \) measures the consistency of aspect ratio, and \( w^{gt}, h^{gt}, w, h \) respectively represent the width and height of two boxes.

\[
\text{Loss}_{\text{cls}} = -\sum \left( y_n \times \ln y^*_n + (1 - y_n) \times \ln(1 - y^*_n) \right) / n
\]

(5)

where \( n \) represents the number of all anchor boxes, \( y^*_n \) denotes the probability of the n-th predicted bounding box, \( y_n \) is the label corresponding to the predicted bounding box, which is formulated by CIoU. Finally, we design the detector loss with a balanced factor of 0.05.

\[
\text{Loss}_{\text{detector}} = \text{Loss}_{\text{reg}} + 0.05 \times \text{Loss}_{\text{cls}}
\]

(6)

3) **DRENet Loss**: The training of the enhancer and detector jointly may occur unbalance between the two branches, thus leading to sub-optimal. How to weigh the two branches has a great impact on the performance of ship detection. Motivated by the work of Kendall et al. [64], we design a method to balance the training preference of the enhancer and detector automatically, which is achieved by two learnable weight coefficients. Finally, the DRENet loss is defined as:

\[
\text{Loss} = \frac{1}{2a^2} \text{Loss}_{\text{enhancer}} + \frac{1}{2b^2} \text{Loss}_{\text{detector}} + \ln a + \ln b
\]

(7)

where \( a \) and \( b \) are two coefficients automatically learned by the network. In training, \( a \) and \( b \) in the first two items of \( \text{Loss} \) tend to be large values, while in the last two, they tend to be small. They are regulated by each other. This design can help the training progress smooth and steady.

### TABLE I

**Details of CRMA Configuration.** CBS represents operations of Convolution, Batch Normalization, and Silu activation function. MHSA denotes the Multi-head Self-attention Layer.

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Layer</th>
<th>Filters</th>
<th>Size</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CBS</td>
<td>256</td>
<td>1x1</td>
<td>16x16x256</td>
</tr>
<tr>
<td></td>
<td>CBS</td>
<td>128</td>
<td>1x1</td>
<td>16x16x128</td>
</tr>
<tr>
<td></td>
<td>MHSA</td>
<td>—</td>
<td>—</td>
<td>16x16x128</td>
</tr>
<tr>
<td></td>
<td>CBS</td>
<td>256</td>
<td>1x1</td>
<td>16x16x256</td>
</tr>
<tr>
<td></td>
<td>Residual</td>
<td>—</td>
<td>—</td>
<td>16x16x256</td>
</tr>
<tr>
<td>Output</td>
<td>Concat</td>
<td>—</td>
<td>—</td>
<td>16x16x512</td>
</tr>
</tbody>
</table>
D. Implementation Details

Our backbone follows the design of YOLOv5s. There are 5 stages in the backbone, each with downsampling by 2, thus we have different feature maps, $256 \times 256$, $128 \times 128$, $64 \times 64$, $32 \times 32$, and $16 \times 16$. Taking the features of $64 \times 64$, $32 \times 32$, and $16 \times 16$ as detector’s input, we obtain the probability, position, and shape of the ship by inference. As for the enhancer, we take the $64 \times 64$ feature maps for reconstructing a degraded label with a shape of $128 \times 128$.

We adopt Stochastic Gradient Descent (SGD) optimizer with 0.99 momentum and 0.0005 weight decay. The learning rate is initially set to 0.01 and decays in a Cosine annealing strategy until 500 epochs. The batch size is 16. While for the optimization of the two weight coefficients $a$ and $b$ in loss, we adopt Adaptive moment estimation (Adam) optimizer with learning rate set to 0.01. Notably, the enhancer and detector are jointly optimized in the training phase, but in the testing phase, the enhancer can be removed and only the detector is used.

IV. LEVIR-SHIP

In recent decades, there are many effective object detection datasets in optical remote sensing.

NWPU-VHR-10 [65] contains 800 images in 10 categories. However, there are only 57 images including total 302 ships, which is not enough to train a DL network and suffers from overfitting. In addition, the spatial resolution of the dataset is 0.5-2m.

HRSC2016 [66] is a single-class dataset for ship detection. The images are all from 6 well-known ports. There are 1070 images, and 2976 objects in total. The spatial resolution of the dataset is 0.4-2m.

DIOR [67] is a relatively large dataset proposed in recent years. It contains 20 categories, 23463 images, and 192472 instances. The ship category has 2702 images and 62400 objects in total. However, the ship is also under a high spatial resolution.

HRRSD [68] is another large-scale RS dataset recently proposed. There are 13 categories, 21761 images, and 55740 instances. It contains 2165 images with 3886 ship objects, but the resolution of 0.15-1.2m is still too high.

Most ship detection datasets focus on high spatial resolution, such as 0.3m, 0.5m, and 2m [65]–[68]. Few datasets concentrate on tiny ship objects with a lower spatial resolution. In our work, we propose a tiny ship detection dataset named Levir-Ship under the spatial resolution of 16m.

Images in Levir-Ship are captured from multispectral cameras of GaoFen-1 and GaoFen-6 satellites. We only use the R, G, and B bands. 85 scenes have been collected in the dataset with pixel resolutions between 10000x10000 and 50000x20000. We crop the original images to finally get 1973 positive samples and 192472 negative samples with the size of $512 \times 512$. As shown in Fig. 4, we plot the distribution of width/height of ships in the Levir-Ship dataset. We can observe that the ship pixel size in Levir-Ship is almost below $20 \times 20$, and centralizes at around $10 \times 10$. The ship in Levir-Ship is relatively small compared to the vast background, which brings many challenges to the detection network.

Fig. 4. Ship target size statistics in LEVIR-SHIP. We use the smallest enclosing rectangle of the actual ship target as a measure.

TABLE II shows the comparison between Levir-Ship and several existing RS object detection datasets. The ship in Levir-Ship is surrounded by complex background, and is hard to recognize, compared with those ships in HR images. In addition, due to the influence of different time, different photographers, and different locations, the images show different conditions, which brings many challenges to the task.

From Fig. 5, we can see that the background is complex, and ships in the images are hard to find even zoomed in several times. Meanwhile, since Levir-Ship covers a variety of environmental conditions, DL network can obtain better generalization, stability, and sufficiency.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we conduct experimental comparative analyses of the proposed method. Our experiment is based on Levir-Ship, and we randomly divide the dataset into training set, validation set, and test set (details shown in TABLE III). All experiments are implemented on an NVIDIA Tesla V100 GPU.

A. Evaluation Metrics

We evaluate the effectiveness and efficiency of the proposed method by detection accuracy and model complexity. The metrics are Average Precision (AP), Floating Point Operations (FLOPs), Model Parameters, and Frames Per Second (FPS). Details are described as follows.

To evaluate detection accuracy, We use $AP_{50}$ (prediction is a true positive sample when the IoU is larger than 0.5) as the evaluation metric to measure the detection accuracy of the model. In calculating AP value, we conduct the method illustrated in the COCO dataset (in the Precision-Recall curve, according to classification confidence, 101 points are sampled
### TABLE II

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Images Number</th>
<th>BB Number</th>
<th>Source</th>
<th>Resolution</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWPU VHR-10</td>
<td>57</td>
<td>302</td>
<td>Google Earth</td>
<td>0.5-2m</td>
<td>2014</td>
</tr>
<tr>
<td>HRSC2016</td>
<td>1070</td>
<td>2976</td>
<td>Google Earth</td>
<td>0.4-2m</td>
<td>2016</td>
</tr>
<tr>
<td>DIOR</td>
<td>2702</td>
<td>62400</td>
<td>Google Earth</td>
<td>0.5-30m</td>
<td>2018</td>
</tr>
<tr>
<td>HRRSD</td>
<td>2165</td>
<td>3886</td>
<td>Google Earth &amp; Baidu Earth</td>
<td>0.15-1.2m</td>
<td>2019</td>
</tr>
<tr>
<td>LEVIR-SHIP (Ours)</td>
<td>3876</td>
<td>3219</td>
<td>GaoFen-1 &amp; GaoFen-6</td>
<td>16m</td>
<td>2021</td>
</tr>
</tbody>
</table>

(a) Images with different amounts of cloud  
(b) Images of different land percentages  
(c) Images of different light intensity  
(d) Images of different sea area characteristics

Fig. 5. A partial display of the data contained in LEVIR-SHIP under different situations that may be encountered in the process of ship detection. The blue bounding boxes are the ground truth.

### TABLE III

<table>
<thead>
<tr>
<th>The division of training set, validation set, and test set</th>
<th>Image number</th>
<th>Target number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set</td>
<td>2320</td>
<td>2002</td>
</tr>
<tr>
<td>Validation set</td>
<td>788</td>
<td>665</td>
</tr>
<tr>
<td>Test set</td>
<td>788</td>
<td>552</td>
</tr>
</tbody>
</table>

at equal intervals, and then the Precision values corresponding to these points are accumulated and averaged to obtain the final AP value) to ensure that the calculation of the Precision-Recall curve area is more accurate. We use FLOPs, Parameters, and FPS to reflect the detection speed.

### B. State-of-the-art Comparison

We compare DRENet with other state-of-the-art networks in ship detection on Levir-SHIP dataset, including YOLOv3 [40], YOLOv5s [27], Retinanet [43], SSD [42], FasterRCNN [36], EfficientDet [44], FCOS [47], CenterNet [46], HSFNet [11], ImYOLOv3 [24], DFR and RFE structure proposed by Tian et al. [12]. The last three methods are proposed for ship detection task. We have also tried to compare with some traditional methods such as SVDNet [28] and PCANet [69] (configured as [28]), but unfortunately found these methods failed as the ship targets only cover a small area of the image. Unless otherwise specified, we train all networks for 500 epochs, using batch size 16.

TABLE IV shows the comparative results of networks. It can be seen that compared with other methods, our proposed method achieves the highest detection accuracy and real-time detection speed. DRENet achieves a healthy 1.5 point AP gap with the closest competitor, EfficientDet-D2 [44], while being much faster (85 vs. 21 FPS). Compared with the fastest YOLOv5s [27], DRENet is a minor slower but much more accurate (82.4 vs. 75.6 AP). Furthermore, the Params and FLOPs of DRENet are small. These can demonstrate the excellent detection performance of DRENet on the tiny ship detection task.

Fig. 6 gives some detection results between DRENet and other methods. Blue boxes represent true positive detections. Red boxes denote false negative detections. Yellow boxes indicate false positive samples. From the top to the bottom in the figure, the complexity of the background gradually rises. The ship in the image behaves very small, and the distinguishable features are not clear enough. Meanwhile, some ships are surrounded by fractus clouds, leading to a challenge in detection. We can observe that our method achieves better detection
TABLE IV
COMPARISON WITH SOME WELL-KNOWN DETECTION METHODS AND STATE-OF-THE-ART SHIP DETECTION METHODS. THE BOLD REPRESENTS THE OPTIMAL METRIC.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Params(Inference)</th>
<th>FLOPs(Inference)</th>
<th>AP</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv3</td>
<td>61.52M</td>
<td>99.2G</td>
<td>69.9</td>
<td>61</td>
</tr>
<tr>
<td>YOLOv5s</td>
<td>7.05M</td>
<td>10.4G</td>
<td>75.6</td>
<td>95</td>
</tr>
<tr>
<td>Retinanet (ResNet50)</td>
<td>36.33M</td>
<td>104.4G</td>
<td>74.9</td>
<td>12</td>
</tr>
<tr>
<td>SSD (VGG16)</td>
<td>24.39M</td>
<td>175.2G</td>
<td>52.6</td>
<td>25</td>
</tr>
<tr>
<td>FasterRCNN (VGG16)</td>
<td>136.70M</td>
<td>299.2G</td>
<td>70.8</td>
<td>10</td>
</tr>
<tr>
<td>EfficientDet-D0</td>
<td><strong>3.84M</strong></td>
<td><strong>4.6G</strong></td>
<td>71.3</td>
<td>32</td>
</tr>
<tr>
<td>EfficientDet-D2</td>
<td>8.01M</td>
<td>20.0G</td>
<td>80.9</td>
<td>21</td>
</tr>
<tr>
<td>FCOS (ResNet50)</td>
<td>5.92M</td>
<td>51.8G</td>
<td>75.5</td>
<td>37</td>
</tr>
<tr>
<td>CenterNet (Hourglass-104)</td>
<td>191.24M</td>
<td>584.6G</td>
<td>77.7</td>
<td>25</td>
</tr>
<tr>
<td>HSFNet</td>
<td>157.59M</td>
<td>538.1G</td>
<td>73.6</td>
<td>7</td>
</tr>
<tr>
<td>ImYOLOv3</td>
<td>62.86M</td>
<td>101.9G</td>
<td>72.6</td>
<td>51</td>
</tr>
<tr>
<td>MaskRCNN (ResNet50)+DFR+RFE</td>
<td>24.99M</td>
<td>237.8G</td>
<td>76.2</td>
<td>6</td>
</tr>
<tr>
<td>DRENet (ours)</td>
<td>4.79M</td>
<td>8.3G</td>
<td><strong>82.4</strong></td>
<td>85</td>
</tr>
</tbody>
</table>

results than other methods. DRENet can detect more true positive ships and miss fewer targets. Additionally, the false positive detections can be well avoided, compared with other method detection results shown in the figure. For example, as for 3rd-row images, no methods mentioned can find the ship behind the cloud except for DRENet. The comparable results show DRENet has higher applicability and robustness for the tiny ship object detection task.

To further analyze the performance of our method under different background situations, we roughly divide the test set into further five subsets: “clam sea”, “thin cloud”, “thick cloud”, “strong wave” and “fractus cloud”, according to different background situations. To better illustrate the difference between these situations, for each subset we select an example image, displayed in Fig. 7. TABLE V shows the comparisons of DRENet with other methods in different situations, from which we can observe that the subsets “thick cloud”, “strong wave” and “fractus cloud” are more difficult than the other two, as the performance of other methods has dropped a lot in these scenes. From the results, we can see that our method is more robust to complex backgrounds like “fractus cloud” than other methods while also keeping good performance in the simple scene like “calm sea”.

C. Controlled Experiment

1) Degradation Function $F$: We compare the degradation effect on the original input image when taking different function $F(\cdot)$ to adjust the mean kernel size, including linear function, logarithmic function, and exponential function. As shown in Fig. 8, it can be seen that the exponential function conducts a more obvious degradation effect and can better meet the need of the enhancer to highlight the ship and blur the background. In order to keep the details of the ship and minimize the effect of the background, we apply the exponential function to calculate the size of the mean filter kernel at each pixel position. The steps are given in Algorithm 1.

2) Design of Enhancer: The degraded reconstruction enhancer is used for extracting more distinguishable basic feature representation between ship target and background. We conduct experiments on the different structures and supervisions of the enhancer.

Different Enhancer Structure. Here, we make comparisons on different inner structures of the enhancer to select the most suitable enhancer structure for the ship detection task. We evaluate the performance by AP and network complexity. TABLE VI shows the results in detail. “n-RCAB” represents the number of RCAB included in our enhancer network. “Upsampling” denotes different upsampling multiples, corresponding to reconstructing Enhancer labels of different resolutions. It can be seen from the results that when 1 RCAB structure and upsampling 2 times are carried out, the performance is the best. When the enhancer becomes more complex, the training of the enhancer focus more on high-level layers, instead of the basic layers of the backbone. This may illustrate why the performance declines as the structure becomes weighted. Meanwhile, when the upsampling factor gets larger, reconstruction of the degraded image becomes more difficult, which leads to the deterioration of the enhancer. Then the trained enhancer cannot achieve the proper feature representation the detector needs.

Different Supervision Form. We compare different label generation strategies in the enhancer to acquire better supervision for the reconstruction branch. As shown in TABLE VII, different generation strategies have different impacts on detection accuracy. The enhancer structure we take is 1 RCAB and upsampling 2 times. From the table, we can make some observations. First, direct down-sampling can improve the detection performance compared with only the detector without the enhancer. This may be because the backbone retains more spatial information by fitting the label, which
Second, we can also see the sudden-blur method reduces the detection accuracy, which is mainly because a ship in Levir-Ship dataset only occupies a small area, and the training of the network will suffer from a relatively small loss. Thus, the network learning deviates from the original direction and cannot focus on the details of the tiny ship while suppressing the background. Third, by using “Selective Degradation”, i.e. the continuity-blur in TABLE VII to generate the label, the detection performance has been significantly improved.

Fig. 6. Comparisons of the detection results by different methods. (a) YOLOv5s. (b) EfficientDet-D2. (c) ImYOLOv3. (d) Method [12]. (e) Ours. The blue box represents the real target detected, the red box represents the missed target and the yellow box represents false alarms. To facilitate observation, we randomly enlarge some missed ship targets.

makes more contributions to the detection of tiny ships.
Fig. 7. Images as examples for each background situation. (a) calm sea. (b) thin cloud. (c) thick cloud. (d) strong wave. (e) fractus cloud.

### TABLE V

<table>
<thead>
<tr>
<th>Situations</th>
<th>Number</th>
<th>YOLOv5s</th>
<th>EfficientDet-D2</th>
<th>ImYOLOv3</th>
<th>Method [12]</th>
<th>DRENet (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>calm sea</td>
<td>262</td>
<td>76.8</td>
<td><strong>83.0</strong></td>
<td>75.9</td>
<td>78.4</td>
<td><strong>82.1</strong></td>
</tr>
<tr>
<td>thin cloud</td>
<td>238</td>
<td>84.3</td>
<td>83.9</td>
<td>83.7</td>
<td>82.4</td>
<td><strong>87.3</strong></td>
</tr>
<tr>
<td>thick cloud</td>
<td>60</td>
<td>60.5</td>
<td>78.9</td>
<td>56.6</td>
<td>70.1</td>
<td><strong>86.8</strong></td>
</tr>
<tr>
<td>strong wave</td>
<td>101</td>
<td>73.4</td>
<td>73.3</td>
<td>62.8</td>
<td>71.8</td>
<td><strong>82.8</strong></td>
</tr>
<tr>
<td>fractus cloud</td>
<td>127</td>
<td>72.1</td>
<td>74.7</td>
<td>61.8</td>
<td>64.4</td>
<td><strong>76.5</strong></td>
</tr>
</tbody>
</table>

### TABLE VI

<table>
<thead>
<tr>
<th>Structure</th>
<th>n-RCAB</th>
<th>Upsampling</th>
<th>AP</th>
<th>Params(Train)</th>
<th>FLOPs(Train)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv5s</td>
<td>—</td>
<td>—</td>
<td>75.6</td>
<td>7.05M</td>
<td>10.4G</td>
</tr>
<tr>
<td>YOLOv5s+Degraded reconstruction enhancer</td>
<td>1</td>
<td>×2</td>
<td><strong>76.8</strong></td>
<td>8.25M</td>
<td>20.3G</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>×4</td>
<td>76.2</td>
<td>8.84M</td>
<td>40.0G</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>×2</td>
<td>76.6</td>
<td>8.55M</td>
<td>22.7G</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>×8</td>
<td>75.6</td>
<td>9.73M</td>
<td>121.2G</td>
</tr>
</tbody>
</table>

### TABLE VII

<table>
<thead>
<tr>
<th>Label generation method</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original (YOLOv5s)</td>
<td>75.6</td>
</tr>
<tr>
<td>Down-sampling</td>
<td>76.6</td>
</tr>
<tr>
<td>Sudden-blur+Down-sampling</td>
<td>75.2</td>
</tr>
<tr>
<td>Continuity-blur(λ=1.01)+Down-sampling</td>
<td>76.8</td>
</tr>
<tr>
<td>Continuity-blur(λ=1.03)+Down-sampling</td>
<td><strong>78.7</strong></td>
</tr>
<tr>
<td>Continuity-blur(λ=1.05)+Down-sampling</td>
<td>77.1</td>
</tr>
</tbody>
</table>

In addition, we take different parameters $\lambda$ in degradation function $F(\cdot)$, and finally select 1.03 because of its relatively great performance.

3) **Structure of Detector**: We have made some improvements to the original YOLOv5s prototype to achieve better performance in tiny ship detection. To verify the effect of different modifications, we conduct a series of ablation study on the detector. Results are shown in TABLE VIII. We can get some observations from the table. First, by adding “Scale Layer”, a minor improvement appears without increasing the complexity. It may be because the Scale Layer alleviates the problem of unbalanced training of different layers in FPN to a certain extent. Second, after replacing the CSP bottleneck with the CRMA module we propose, the complexity is significantly reduced, and the detection AP has risen by 4.2. We believe this is due to the greater receptive field which leads to richer features that benefit tiny ship object detection. It verifies the effectiveness of the designed CRMA structure. We further make a trade-off of whether to remove PAN. From TABLE IX, we can observe the number of network parameters and operations be greatly reduced at the little expense of detection accuracy when the PAN structure is removed. Thus, we remove PAN in our final model.

Compared with the YOLOv5s prototype, the amount of parameters and FLOPs has been reduced by nearly $-32\%$ and $-19\%$ respectively, while AP has increased 4.7 points. With faster speed, smaller model, and higher accuracy, the improved detector is more suitable for detecting ships in MR RS images.

4) **Network Overall Setting**: In this part, we will conduct experiments on loss function and training strategy.

**Loss Function.** DRENet is composed of an enhancer and
**Table VIII**
The ablation study of the modifications in the detector. The optimal values are marked in bold.

<table>
<thead>
<tr>
<th>Network description</th>
<th>Scale</th>
<th>CRMA</th>
<th>AP</th>
<th>Params</th>
<th>FLOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (YOLOv5s)</td>
<td>×</td>
<td>×</td>
<td>75.6</td>
<td>7.05M</td>
<td>10.4G</td>
</tr>
<tr>
<td>Add Scale</td>
<td>✓</td>
<td>×</td>
<td>76.3</td>
<td>7.05M</td>
<td>10.4G</td>
</tr>
<tr>
<td>Replace CSP bottleneck with CRMA</td>
<td>×</td>
<td>✓</td>
<td>79.8</td>
<td>5.69M</td>
<td>9.1G</td>
</tr>
<tr>
<td>Add Scale and CRMA</td>
<td>✓</td>
<td>✓</td>
<td>80.8</td>
<td>5.69M</td>
<td>9.1G</td>
</tr>
</tbody>
</table>

**Fig. 8.** The degradation effect when taking a logarithmic function, linear function, and exponential function to calculate the mean filter kernel size at each position. The red circles in the images are the ships. \( \lambda \) is a constant of functions, which is set to ensure the ship areas are kept clear. \( x \) is the shortest distance to ship targets and \( y \) denotes the kernel size.

**Figure 8**
- (a) Origin images
- (b) Different function \( F(\cdot) \)
- (c) \( y = log_{\lambda} x \)
- (d) \( y = \lambda x \)
- (e) \( y = \lambda^x x \)

**Table IX**
The trade-off between keeping \( \text{PAN} \) and removing \( \text{PAN} \). \( w \) denotes keeping, while \( \text{w/o} \) denotes removing.

<table>
<thead>
<tr>
<th></th>
<th>AP</th>
<th>Params</th>
<th>FLOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{w} )</td>
<td>80.8</td>
<td>5.69M</td>
<td>9.1G</td>
</tr>
<tr>
<td>( \text{w/o} )</td>
<td>80.3</td>
<td>4.79M</td>
<td>8.3G</td>
</tr>
</tbody>
</table>

**Figure 9.** The degradation effect when takes a logarithmic function, linear function, and exponential function to calculate the mean filter kernel size at each position. The red circles in the images are the ships. \( \lambda \) is a constant of functions, which is set to ensure the ship areas are kept clear. \( x \) is the shortest distance to ship targets and \( y \) denotes the kernel size.

**Table X**
Comparison of different configuration methods of the loss function. \( \alpha \) denotes the weight of the sum of the two branch loss functions. The highest AP value is marked in bold.

<table>
<thead>
<tr>
<th>Combination description</th>
<th>weight ( \alpha )</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed weight summation</td>
<td>100</td>
<td>80.9</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>81.6</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>80.6</td>
</tr>
</tbody>
</table>

**Table XI**
Comparison of different training methods. Pre-training refers to training the enhancer first, and then joint training. The highest AP value is marked in bold.

<table>
<thead>
<tr>
<th>Training method</th>
<th>Pretrained epoches</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint train without pre-training</td>
<td>—</td>
<td>82.4</td>
</tr>
<tr>
<td>Joint train with pre-training</td>
<td>100</td>
<td>81.9</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>81.4</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>80.5</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>77.5</td>
</tr>
</tbody>
</table>

**Table XII**
Generalization of DREN on the downsampled version of existing HR dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Enhancer</th>
<th>Efficient Detector</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv5s</td>
<td>-</td>
<td>-</td>
<td>51.5</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>-</td>
<td>52.8</td>
</tr>
<tr>
<td>DRENNet</td>
<td>-</td>
<td>+</td>
<td>50.9</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>+</td>
<td>51.6</td>
</tr>
</tbody>
</table>

Training Strategy. TABLE XI shows the results of different training methods. Our goal is to seek better performance on ship detection, and the enhancer is just an auxiliary branch. So we try to pre-train the enhancer and then perform joint training. However, the experimental result shows that the performance of pre-training is worse than that of direct joint training, and as the pre-training epoch increases, AP gradually decreases. To find a better weight for \( \alpha \), we design a fixed balanced factor, as the factor \( \alpha \) gets larger from 0.1 to 10, AP also increases, which again reveals the effectiveness of the enhancer.

**Training Strategy**

Comparing to the baseline, DREN improves the precision by 0.1% in AP. It indicates that the automatic weight learning method can bring a significant improvement. Meanwhile, in the method with a fixed balanced factor, as the factor \( \alpha \) gets larger from 0.1 to 10, AP also increases, which again reveals the effectiveness of the enhancer.
declines. We believe that this phenomenon is mainly because the two branches complement each other. Since the enhancer can be regarded as a target location plus target segmentation, it is more difficult to directly train the enhancer, and network learning may appear to be deteriorated and divergent. While the detector can provide targets’ position guidance for the enhancer, so that it can have a better learning effect. Therefore, the pre-training of the enhancer will result in poor detection accuracy, and will intensify with the number of epochs.

5) Generalization on Existing High-Resolution Dataset: We also analyze the generalization of our method on the existing HR dataset, that is, implementing experiments on the downsampled version of HR images.

Here we choose the HRRSD dataset to implement the experiments. Considering the size and resolution (0.15-1.2m) of the images in HRRSD, we first reshape all the images into 64x64, and then do ablation experiments to reveal the performance of DRENet.

From the results in TABLE XII, it appears that when only our efficient detector applied, the AP value drops a bit (-0.6), and we argue it is mainly caused by the size of the input. As the image size is only 64x64, after being processed by the network’s backbone, the features consumed by CRMA will be at an even smaller size, thus the large-receptive-field characteristic of CRMA cannot gain more advantages than the convolution operations in YOLOv5s’s CSP-like structure. Therefore, as we remove PAN structure in our efficient detector, the performance then deteriorates, which is also verified in TABLE IX.

While, we can also observe from the results that our enhancer continues displaying a big boost. When we only apply the enhancer, AP increases +1.3 points, and when we combine both the enhancer and detector, our DRENet achieves a near performance (AP +0.1) compared with baseline YOLOv5s. These results can further demonstrate the effectiveness of our enhancer. Considering the computation and complexity saved in our DRENet (demonstrated in TABLE IV), DRENet actually achieves good performance even in the downsampled version of the existing high-resolution dataset.

VI. CONCLUSION

In this paper, we propose a tiny ship detection dataset - “Levir-Ship”, and an efficient tiny ship detection network - DRENet towards the real-world tiny ship detection task. Levir-Ship contains 3876 images from GaoFen-1 and GaoFen-6 satellites with multispectral cameras of 16m spatial resolution. DRENet performs two parts, including an enhancer to obtain more distinguishable features in the backbone, and a detector to carry efficient detection on tiny ships. The enhancer is inference-free, so it would not introduce extra inference cost on application. Experiments on Levir-Ship verify that our proposed DRENet has higher detection accuracy and ensures real-time detection speed at the same time compared with other state-of-the-art detection methods.

REFERENCES

Jianqi Chen received his B.S. degree from the Image Processing Center, School of Astronautics, Beihang University in 2021. He is currently pursuing his M.S. degree in the Image Processing Center, School of Astronautics, Beihang University. His research interests include deep learning, object detection and artificial intelligence safety.

Keyan Chen received the B.S. degree from the School of Astronautics, Beihang University, Beijing, China, in 2019. He is currently working toward the M.S. degree in the Image Processing Center, School of Astronautics, Beihang University. His research interests include image processing, machine learning and pattern recognition.

Hao Chen received his B.S. degree from the Image Processing Center School of Astronautics, Beihang University in 2017. He is currently pursuing his doctorate degree in the Image Processing Center, School of Astronautics, Beihang University. His research interests include machine learning, deep learning and semantic segmentation.

Zhengxia Zou received his BS degree and his Ph.D. degree from Beihang University in 2013 and 2018. He is currently an Associate Professor at the School of Astronautics, Beihang University. During 2018-2021, he was a postdoc research fellow at the University of Michigan, Ann Arbor. His research interests include computer vision and related problems in remote sensing. He has published more than 20 peer-reviewed papers in top-tier journals and conferences, including TPAMI, TIP, TGRS, CVPR, ICCV, AAAI. His research was featured in more than 30 global tech media and was adopted by a number of application platforms with over 50 million users worldwide. His personal website is https://zhengxiazou.github.io/.

Zhenwei Shi (Member, IEEE) received the Ph.D. degree in mathematics from the Dalian University of Technology, Dalian, China, in 2005. He was a Post-Doctoral Researcher with the Department of Automation, Tsinghua University, Beijing, China, from 2005 to 2007. He was Visiting Scholar with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA, from 2013 to 2014. He is currently a Professor and the Dean of the Image Processing Center, School of Astronautics, Beihang University, Beijing. He has authored or coauthored over 200 scientific papers in refereed journals and proceedings, including the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Image Processing, the IEEE Transactions on Geoscience and Remote Sensing, the IEEE Geoscience and Remote Sensing Letters, the IEEE Conference on Computer Vision and Pattern Recognition, and the IEEE International Conference on Computer Vision. His research interests include remote sensing image processing and analysis, computer vision, pattern recognition, and machine learning.

Dr. Shi also serves as an Editor for the Pattern Recognition, the ISPRS Journal of Photogrammetry and Remote Sensing, the Infrared Physics and Technology, and so on. His personal website is http://levir.buaa.edu.cn/.