




# Multitask Learning for Ship Detection From Synthetic Aperture Radar Images

Xin Zhang , Chunlei Huo , *Member, IEEE*, Nuo Xu , Hangzhi Jiang, Yong Cao, Lei Ni, and Chunhong Pan, *Member, IEEE*

**Abstract**—Ship detection from synthetic aperture radar (SAR) images is inherently subject to the special imaging mechanism of SAR. In recent years, deep-learning-based techniques for detecting objects from optical images have rapidly advanced and promoted the development of SAR image detection technology. However, the strong speckle noise in SAR images degrades low-level feature learning in shallow layers, hindering the higher level learning of semantic features for object detection. In view of the problems encountered in direct end-to-end feature learning for object detection and the close relationship between objects and auxiliary cues, a multitask learning-based object detector (MTL-Det) is proposed in this article to distinguish ships in SAR images. The proposed approach models the ship detection problem, not as a single object detection task, but as three cooperative tasks. The model involves two auxiliary subtasks that are focused on learning object-specific cues (e.g., texture and shape) for the ship detection task, which is constrained by the pseudoground truth generated by the main task. Assisted by auxiliary subtasks, the low-level features are robust to speckle noise and reliably support high-level feature learning. Compared with traditional single-task-based object detectors, more discriminative object-specific features are learned by multitask learning without the extra cost of manual labeling. The experiments conducted in this study help demonstrate the advantages of MTL-Det in improving the ship detection performance on two SAR datasets: high-resolution SAR images dataset and large-scale SAR ship detection dataset-v1.0.

**Index Terms**—Multitask learning, synthetic aperture radar (SAR), SAR ship detection.

## I. INTRODUCTION

SYNTHETIC aperture radar (SAR) is an active microwave imaging sensor. For practical applications, SARs are promising owing to their all-weather and all-day imaging capabilities. In this context, object detection from SAR images is considerably advantageous for disaster monitoring, emergency

rescue, maritime surveillance, ocean monitoring, and military intelligence acquisition [1]. The aim of object detection is to determine the object's position and identify its semantic label by differentiating between the object and its background. The underlying assumption is that objects of the same class are considerably similar with respect to certain cues and distinct from the background and other classes. However, the abovementioned assumption is refuted by the inconsistency between materials and spectra as well as the degradation of SAR images because of the presence of extreme noise and the incomprehensible imaging mechanism of SAR. For experts, as well as computers, these problems have complicated the recognition of objects from SAR images.

From the perspective of pattern recognition, the difficulty of object detection primarily lies in feature representation. Earlier studies on SAR object detection focused on handcrafted features, such as salient region-based features, shape and texture features, and complex domain features. However, before these object-specific features can be extracted, they are destroyed by the intense speckle noise and clutter background within the SAR image. In other words, a large semantic gap exists between low-level visual features and high-level semantic features. Furthermore, the handcrafted features of traditional methods considerably depend on prior knowledge, such as the type and size of a specific object; hence, extending their application to objects of different types and images of various spatial resolutions is difficult.

In recent years, with the rapid development of deep learning, researchers have gradually introduced convolutional neural networks (CNNs) to object recognition. With their powerful feature-learning capabilities, CNNs can effectively extract high-level semantic features driven by end-to-end learning, considerably improving object detection from optical images. However, the advantages of CNNs are substantially diminished by the complexities of SAR images. Progressive feature learning allows CNNs to recognize objects, edge features, texture features at shallow layers, and semantic layers at deep layers. Unfortunately, the strong speckle noise in SAR images degrades low-level feature learning in shallow layers, hindering the higher level learning of semantic features for object detection. For example, both inshore ships and buildings on the shore appear as bright white spots on SAR images; hence, the object boundary has been destroyed by the speckle noise, rendering it difficult to learn. Moreover, the texture difference between the foreground and background is not sufficiently distinct to discriminate.

Manuscript received May 19, 2021; revised July 23, 2021; accepted August 3, 2021. Date of publication August 6, 2021; date of current version August 26, 2021. This work was supported in part by the National Natural Science Foundation of China under Grants 62071466, 62076242, and 61976208 and in part by the National Key Research and Development Program of China under Grant 2018AAA0100400. (*Corresponding author: Chunlei Huo.*)

Xin Zhang, Chunlei Huo, Nuo Xu, Hangzhi Jiang, Yong Cao, and Chunhong Pan are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: xin.zhang2018@nlpr.ia.ac.cn; clhuo@nlpr.ia.ac.cn; nuo.xu@nlpr.ia.ac.cn; jianghangzhi2018@ia.ac.cn; yong.cao@nlpr.ia.ac.cn; chpan@nlpr.ia.ac.cn).

Lei Ni is with the Graduate School, Space Engineering University, Beijing 100192, China (e-mail: lalagood@qq.com).

Digital Object Identifier 10.1109/JSTARS.2021.3102989

In other words, if no object-specific edge and texture features can be effectively learned by end-to-end training, accomplishing the object detection task through deep learning, either by one-stage you only look once (YOLO) [2] or two-stage faster R-CNN [3], is difficult.

The progressive feature-learning nature of deep learning and the inadequacy of traditional object detectors on SAR images indicate that detecting objects from noisy SAR images is not a single simple detection task. It is a compound complex task in which low-level features must be robust to speckle noise and reliably support high-level feature learning. Motivated by this observation, a multitask learning approach is proposed for SAR ship detection. The contributions of this study are as follows.

1) A novel multitask learning framework is presented to model the exigent object detection problem in SAR images. In this case, two auxiliary tasks focused on learning object-specific cues (e.g., shape and texture) are proposed to improve the classification and regression quality of the detector. The low-level edge features and semantic features are learned through object-related pseudoannotation without the extra cost of manual labeling, and the object detection network is progressively enhanced by the auxiliary networks. Compared with traditional single-task learning, multitask learning is more powerful in learning reliable low-level features to be robust to speckle noise.

2) Task-guided network (TGN) is proposed for the auxiliary task features learning. The multiscale features alignment and multimodal features fusion modules are presented for multitask features fusion. In addition, a novel auxiliary task supervised learning method is formulated to train each proposed TGN efficiently, and a differentiable weighting method (DWM) is devised to further balance learning among multiple tasks. Through the elaborate multitask network, the low-level and high-level features interactively evolve, thus overcoming the impediment of traditional single-network-based detectors.

3) The proposed multitask learning-based object detector (MTL-Det) improves the SAR ship detection performance (i.e., the MTL-Det improves the average precision (AP) of both cascade R-CNN [4] and faster R-CNN by 1.9 points) and achieves the state-of-the-art (SOTA) performance on high-resolution SAR images dataset (HRSID) [5] and large-scale SAR ship detection dataset-v1.0 (LS-SSDD-v1.0) [6].

## II. RELATED WORKS

To understand the limitations of traditional approaches and clarify the novelty of multitask learning, some of the related studies on SAR object detection approaches are briefly reviewed in this section.

### A. Traditional SAR Image Object Detection

Generally, traditional SAR object detection can be broadly classified into the following four types.

1) *Threshold-Based Methods*: These methods statistically model the background clutter and adaptively determine the threshold; pixels with gray values exceeding the threshold are classified as objects. Among these techniques, the constant false

alarm rate (CFAR) is the most widely used method. The threshold is considerably degraded by the cluttered background; hence, the CFAR method is limited by complex scenes. Accordingly, many variants have been proposed to resolve these problems. For uniform background clutter, Gaussian, Rayleigh, Weibull, and  $K$  distributions are typically utilized [7]–[10]. For nonuniform backgrounds in complex scenes, Qin *et al.* [11] applied the generalized gamma distribution to model the sea clutter of high-resolution SAR images. Considering the inadequacy of the CFAR method in parameter estimation, various nonparametric statistical methods have been proposed based on the Parzen window [12], [13]. Regarding clutter sample selection, the unit average CFAR is employed for sampling uniform clutter [14]. It is focused on dealing with the clutter edge through maximum selection CFAR [15]. Thereafter, to resolve the multiobject detection problem via minimum selection and sequential statistics, the use of SO-CFAR [16] and OS-CFAR [17] are suggested.

2) *Salient Region-Based Methods*: Inspired by the attention mechanism of human vision, the salient region-based object detection methods were devised to detect objects of interest by extracting the salient region from the SAR image. Li *et al.* [18] proposed a dual-domain sparse reconstruction saliency strategy to improve the saliency detection performance and robustness to speckle noise. Tan *et al.* [19] proposed a gradient-based saliency detection algorithm for aircraft detection, where a directional local gradient distribution detector was utilized to create a gradient semantic saliency map. Zhao *et al.* [20] proposed a region-based saliency detection algorithm from which a saliency map was derived by combining the global and local region contrasts. To obtain a refined contour, Tu *et al.* [21] proposed a coarse-to-fine detection framework in which an active contour model was employed to segment the proposed region and determine the boundary of objects. Cui *et al.* [22] applied the similarity test on the central pixel with its neighborhood to explore the different scattering mechanisms between ships and sea clutter, and the similar pixel number was proposed to generate the saliency feature map.

3) *Shape and Texture-Based Methods*: These methods, which are based on shape and texture, further improve the detection performance. He *et al.* [23], [24] proposed a component-based detection framework in which the component information and probability of detection were combined to eliminate incorrectly detected objects according to the maximum probability principle. Huang *et al.* [25] attempted to enhance the interclass feature distance between an object and its background using a gray-level co-occurrence matrix.

4) *Complex Domain Image-Based Methods*: This type of method considers the imaging mechanism of SAR images. These images are coherent superposition of electromagnetic vectors that are generated by the interaction between electromagnetic waves and objects; they are complex data with phase information. Therefore, object detection is expected to be performed by analyzing complex data. At present, complex domain image-based methods can be classified into subaperture coherence methods and complex domain statistical modeling methods. The former is based on the concept that the degree of coherence of the object region exceeds that of the background;

hence, objects can be detected by analyzing the degree of coherence. The latter involves statistical modeling of complex images in the complex domain and reducing the false alarm caused by radio-frequency interference and azimuth ambiguity with the aid of complex information.

### B. CNN-Based SAR Object Detection

The CNN-based detection model is composed of the backbone, neck, and head networks. The backbone network extracts features from the image through convolution and pooling operations. The features are sent to the neck network for feature fusion and enhancement. Finally, the head network classifies and locates each candidate instance.

The success of CNN-based object detectors on optical images has considerably motivated research on SAR object detection methods. To resolve the problem with objects that are dense and of different scales, complex backgrounds, false alarm interference, and training with a few samples of SAR images, the use of certain CNN-based detectors has been proposed. Wang *et al.* [26] proposed a two-stage detector in which coarse and fine recognition stages were used to extract the foreground proposals and distinguish objects from the virtual shadow phenomenon, respectively. To solve the problems of dense and small-scale object detection in SAR images, Zhao *et al.* [27] proposed the implementation of an exhaustive ship proposal network and accurate ship discrimination network modules. The former uses three different sizes of filters to encode three feature maps with different scales, and the latter considers the influence of context on the classification accuracy and models the context of proposals. Zhao *et al.* [28] introduced the attention receptive block (ARB) to SAR ship detector, two efficient feature extract modules, receptive fields block and convolutional block attention module, were combined into ARB to build a fine-grained feature pyramid. An *et al.* [29] proposed a rotation-sensitive detection method for SAR images, which alleviates the problems (e.g., detection of small objects and the imbalance of positive and negative samples) and generates a set of oriented bounding boxes (bboxes).

Compared with the two-stage object detection network, single-stage detectors have been introduced into SAR image detection because of their advantage in terms of speed. Du *et al.* [30] proposed a dual-flow neural network based on single shot multibox detector (SSD) and used the saliency map obtained by Itti's method to guide the network to focus on object regions. Deng *et al.* [31] proposed a SAR ship detector, which learned from scratch and in which position-sensitive score maps were introduced to encode the position information into each ship proposal for classification. Yang *et al.* [32] proposed a one-stage oriented bbox-based detector that focused on the three aspects, i.e., the scale distribution alignment, feature optimization process decoupling, and the unbalanced distribution correction.

The related studies mentioned above indicate that most methods model the object detection problem as a single task and ignore the close relationships among coherent tasks, such as saliency detection, shape extraction, and object detection. The strong speckle noise and clutter background impact the progressive feature-learning procedure. The absence of

cooperative evolution between low- and high-level features renders the feedback-driven end-to-end feature learning ineffective. Consequently, the traditional single-task-based object detectors for SAR images are constrained. Although related works [30] have attempted to utilize the saliency features for object detection, it considerably differs from the proposed approach. First, in [30], saliency features are extracted by the Itti's method whose performance influences the final detection performance. In contrast, the auxiliary features of the proposed approach are generated from detection annotation; this is consistent with the object detection task and requires no extra labeling. Second, Du *et al.* [30] merge different scale features, whereas the proposed approach injects positive-specific and label-specific features into the detection network. In the latter, the various feature types, which are driven by the multitask learning loss instead of the pure object detection loss, interactively evolve.

## III. METHODOLOGY

### A. Problem Formulation

Considering the problems involved in traditional single-task-based detectors for SAR images, the rationale of the proposed approach is to learn object-specific features  $\mathbf{f}_{\text{obj}}$  by multitask learning. The multiple tasks consist of one main task  $\mathcal{F}_{T_{\text{main}}}$  for object detection, and  $T$  auxiliary tasks  $\{\mathcal{F}_{T_i}\}_{i=1}^T$  to assist  $\mathcal{F}_{T_{\text{main}}}$

$$\mathbf{f}_{\text{obj}} = \mathcal{F}_{T_{\text{main}}}(\mathcal{F}_B(\mathcal{I}), \{\mathcal{F}_{T_i}(\mathcal{F}_B(\mathcal{I}))\}_{i=1}^T) \quad (1)$$

where  $\mathcal{F}_B$  is the backbone network. Considering the similarity among the different tasks with respect to object cues, the auxiliary and main tasks share a common backbone network  $\mathcal{F}_B$ .

Through classification and regression, the detection results can be obtained from  $\mathbf{f}_{\text{obj}}$ . Auxiliary tasks learn object-specific low-level features, such as edges and textures for the main task, and the main task learns object-specific high-level features, such as proposals and labels. In contrast, the auxiliary and main tasks cooperatively evolve instead of being trained independently.

Considering the varied objectives of different tasks, each  $\mathcal{F}_{T_i}$  is implemented by a TGN. To preserve label consistency among different tasks and avoid extra annotation, the ground truths (GTs) of auxiliary tasks  $\{\mathbf{A}_{T_i}\}_{i=1}^T$  are automatically generated from detection-specific labels  $\mathbf{A}_{\text{Det}}$ . The abovementioned label consistency is important for multitask cooperative learning. More specifically, the networks of various tasks should simultaneously converge, and the final features  $\mathbf{f}_{\text{obj}}$  learned by the auxiliary and main tasks should be unanimous. Otherwise, the performance of multitask learning becomes worse than that of traditional single-task learning.

For the main task, various detectors may be used, e.g., one-stage YOLO, or two-stage faster R-CNN. For simplicity, this study focuses on MTL-Det using faster R-CNN as an example. As illustrated in Fig. 1, MTL-Det consists of a backbone network, region proposal network (RPN), neck network, TGN, and head network. For the input image  $\mathcal{I} \in \mathbb{R}^{3 \times H \times W}$ , multiscale FPN features  $\mathbf{f}_S$  are generated through the backbone network (e.g., ResNet [33] and ResNeXt [34]) and neck network (e.g., FPN [35]). TGNs derive auxiliary task features  $\{\mathbf{f}_{T_i}\}_{i=1}^T \in \mathbb{R}^{T \times C_{\text{atf}} \times H_{\text{atf}} \times W_{\text{atf}}}$  and multimodal feature  $\mathbf{f}_M \in \mathbb{R}^{C_{\text{atf}} \times H_{\text{atf}} \times W_{\text{atf}}}$ . By aligning and combining multiscale features

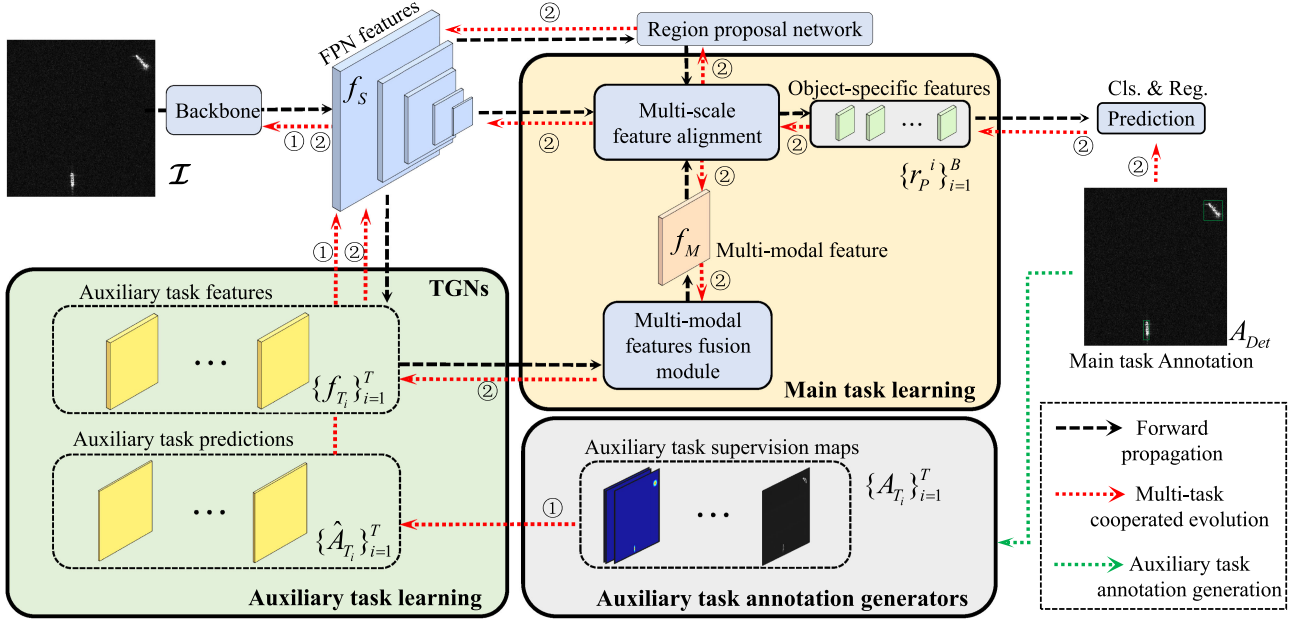


Fig. 1. MTL-Det Architecture. The multiscale FPN feature  $f_S$  obtained by the backbone network is fed into the TGNs to acquire the auxiliary task features  $\{f_{T_i}\}_{i=1}^T$ . Then,  $\{f_{T_i}\}_{i=1}^T$  passes through the fusion module to obtain the multimodal feature  $f_M$ . The MSFA module takes  $f_M$ ,  $f_S$  and the proposal boxes generated by RPN as the input to obtain the final object-specific features  $\{r_P^i\}_{i=1}^B$  for prediction. In the training phase, the auxiliary task supervision maps  $\{A_{T_i}\}_{i=1}^T$  are obtained by auxiliary task annotation generators, which are used for updating TGNs through supervised learning of auxiliary task predictions  $\{\hat{A}_{T_i}\}_{i=1}^T$ . Cooperated evolution of the main task and auxiliary tasks are conducted to update the entire network.

and multimodal features, reliable object-specific RoI features  $\{r_P^i\}_{i=1}^B \in \mathbb{R}^{B \times C_{\text{atf}} \times 7 \times 7}$  are derived for the object detection task. In this article,  $C_{\text{atf}}$ ,  $H_{\text{atf}}$ ,  $W_{\text{atf}}$  are set to 256,  $\frac{H}{8}$  and  $\frac{W}{8}$ , respectively. The following sections elaborate on MTL-Det.

### B. Auxiliary Task Learning

As shown in Fig. 2, two supplementary tasks (e.g., semantic feature learning and edge feature learning) aim to improve the object detection task with respect to the quality of the bbox regression and recognition accuracy.

1) *Semantic Feature Learning*: From the vertical viewpoint angle, the shape of the ship resembles a rectangle. Consequently, if a horizontal bbox is used to label the ship, the ship appears at the center of the bbox. For the same reason, the more distant a position is from the center, the less probable that a ship is present. Based on this prior knowledge and centerness in fully convolutional one-stage object detection (FCOS) [36], the semantic feature-learning task was designed to assist the detection task. The label generation procedure for semantic feature learning  $G_{\text{semantic}}$  is shown in Fig. 2(a)–(d). With the aid of the original annotations (horizontal bboxes) of the detection task, the position of each pixel in the horizontal bbox is converted into an object-specific probability. Fig. 2(b) and (c) display a set of overlapping bboxes detection annotations and corresponding semantic feature learning supervision maps. For each GT point  $(x, y)$  in bbox  $i$ , the probability is defined as follows:

$$p(x, y)^i = \sqrt{\frac{\min(D_{\text{left}}^i, D_{\text{right}}^i)}{\max(D_{\text{left}}^i, D_{\text{right}}^i)} \times \frac{\min(D_{\text{top}}^i, D_{\text{bottom}}^i)}{\max(D_{\text{top}}^i, D_{\text{bottom}}^i)}}} \quad (2)$$

where  $D_{\text{left}}^i$ ,  $D_{\text{right}}^i$ ,  $D_{\text{top}}^i$ , and  $D_{\text{bottom}}^i$  represent the distance from  $(x, y)$  to the left, right, top, and bottom boundaries of bbox  $i$ , respectively.

As shown in Fig. 2(d), multiple overlapping bboxes exist within the area of dense objects. The overlapping regions have a greater object-related probability, and the probabilities of GT points in all boxes are aggregated to obtain the final supervision map  $\mathcal{A}_{\text{semantic}} \in \mathbb{R}^{1 \times H \times W}$ . For each point  $(x, y)$  in  $\mathcal{A}_{\text{semantic}}$ , the object-related probability is

$$p(x, y) = \sum_{i=1}^K p(x, y)^i \quad (3)$$

where  $K$  is the number of bboxes containing point  $(x, y)$ .

The purpose of the centerness in [36] is to select the sample points close to the object center, whereas the semantic feature-learning task assists the detection task. Specifically, the features learned by the semantic feature-learning task aided in perceiving object semantics, thus improving the recognition performance.

2) *Edge Feature Learning*: The object shape and edge are beneficial for improving the regression quality of the bbox. The edge of the foreground is close to the background; hence, edge feature learning can aid the detector to more accurately identify the object position.

The label generation routine of edge feature learning  $G_{\text{edge}}$  is shown in Fig. 2(e)–(h). The initial edge map  $\mathbf{I}_{\text{edge}}$  is generated by the Canny edge detection algorithm [37] [see Fig. 2(f)]. The edge of the foreground describes the shape of the object. The noise edge impacts the perception of the foreground edge because of the noise and clutter background in SAR images; hence, only

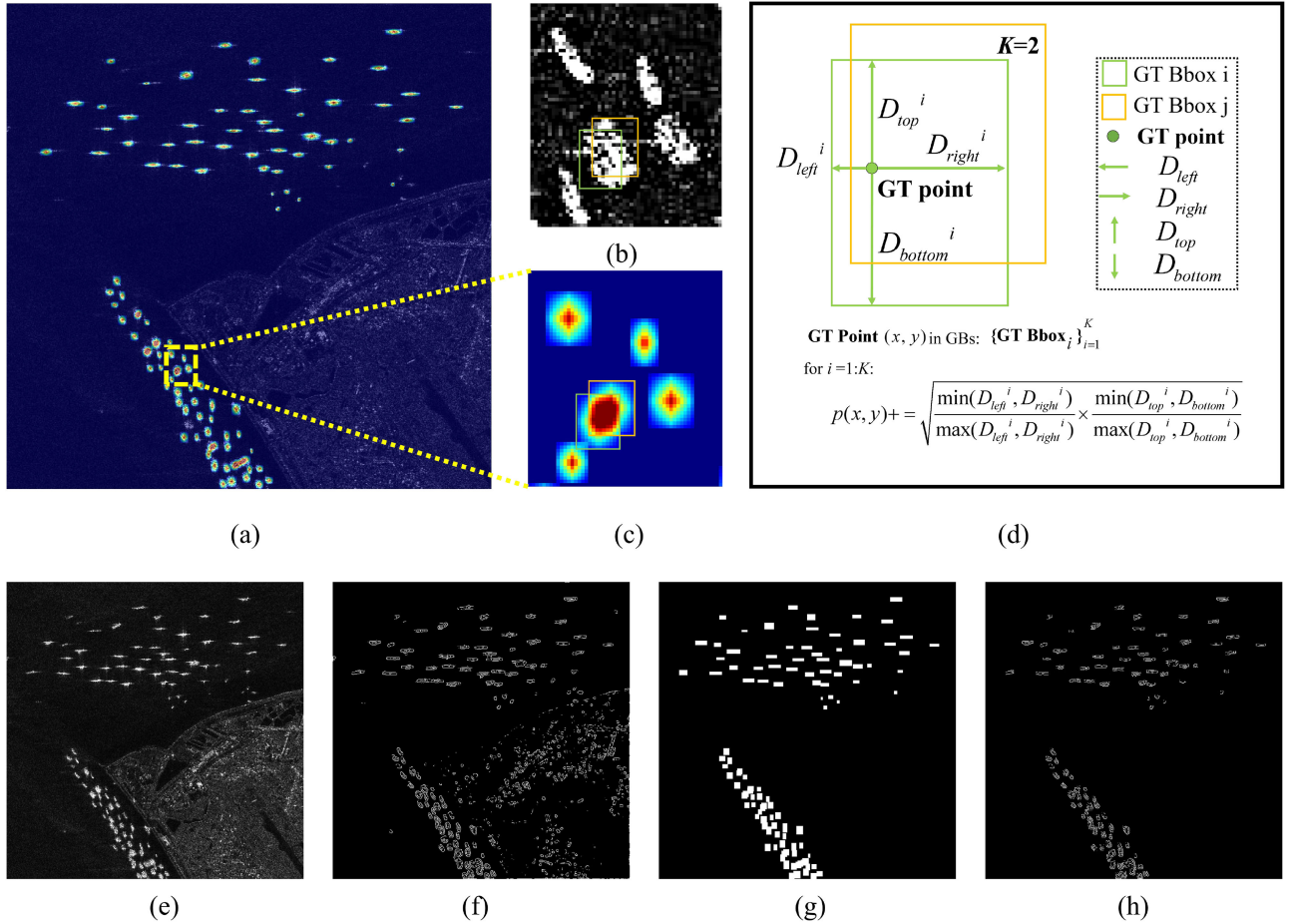


Fig. 2. Annotation generation routine of edge feature learning and semantic feature learning, where (a)–(d) and (e)–(h) show the annotation generators of the proposed auxiliary tasks  $G_{\text{semantic}}$  and  $G_{\text{edge}}$ . (a) Supervision map of semantic feature learning  $\mathcal{A}_{\text{semantic}}$ . (b) Detailed display of detection annotation  $\mathcal{A}_{\text{Det}}$ . (c) Detailed display of  $\mathcal{A}_{\text{semantic}}$ . (d) Calculation of the probability of a GT point in overlapping bboxes. (e) Original image. (f) Edge image generation. (g) Foreground mask generation. (h) Supervision map of edge feature learning  $\mathcal{A}_{\text{edge}}$ .

the edge map of the foreground is used as annotation to learn edge features. The horizontal bbox GT is employed to generate the object mask  $\mathbf{I}_{\text{mask}}$  [see Fig. 2(g)], and  $\mathbf{I}_{\text{mask}}$  is used to filter the foreground edges in  $\mathbf{I}_{\text{edge}}$ . As shown in Fig. 2(h), the final edge supervision map  $\mathcal{A}_{\text{edge}} \in \mathbb{R}^{1 \times H \times W}$  is obtained as follows:

$$\mathcal{A}_{\text{edge}} = \mathbf{I}_{\text{mask}} \odot \mathbf{I}_{\text{edge}} \quad (4)$$

where  $\odot$  represents the Hadamard product.

3) *Task-Guided Network*: Each auxiliary task learning was implemented using a TGN. In contrast to traditional detectors, which directly apply image processing methods, the proposed approach uses low-level features (e.g., edges) as the supervision information to learn the feature subspace and achieve object representation in different modal feature subspaces. The TGN consists of three stages: multiscale feature aggregation, feature learning, and prediction. As shown in Fig. 3, the multiscale FPN features  $\mathbf{f}_S$  enter the  $1 \times 1$  convolution block to align the channel. The FPN features are resized to the given scale (i.e., scale of  $l$ ) by bilinear interpolation. Element-wise addition was applied to all feature maps after scale normalization.

At the feature-learning stage, the aggregated feature map is extracted through the  $3 \times 3$  convolution block; the convolution

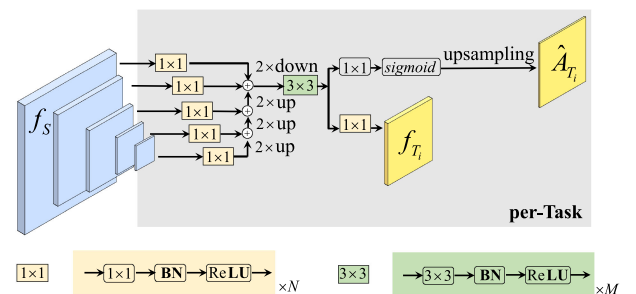


Fig. 3. TGN. FPN features  $\mathbf{f}_S$  of different levels are aligned in the channel dimension via a  $1 \times 1$  convolutional block. Then, the different scale levels feature maps are resized to the given scale by upsampling and downsampling, then the element-wise sum is conducted to aggregate scale-aligned feature maps. The resulting feature maps are fed into the CNN to derive the auxiliary task features  $\{\mathbf{f}_{T_i}\}_{i=1}^T$  and the prediction  $\{\hat{\mathcal{A}}_{T_i}\}_{i=1}^T$ . Each convolution block consists of a convolution layer, a batch normalization (BN) layer, and a rectified linear unit (ReLU) activation layer.

kernel projects the initial features to the feature subspaces of different modalities. Furthermore, the  $1 \times 1$  convolution block is used to correct auxiliary task features and obtain feature  $\mathbf{f}_{T_i}$  for task  $i$ .

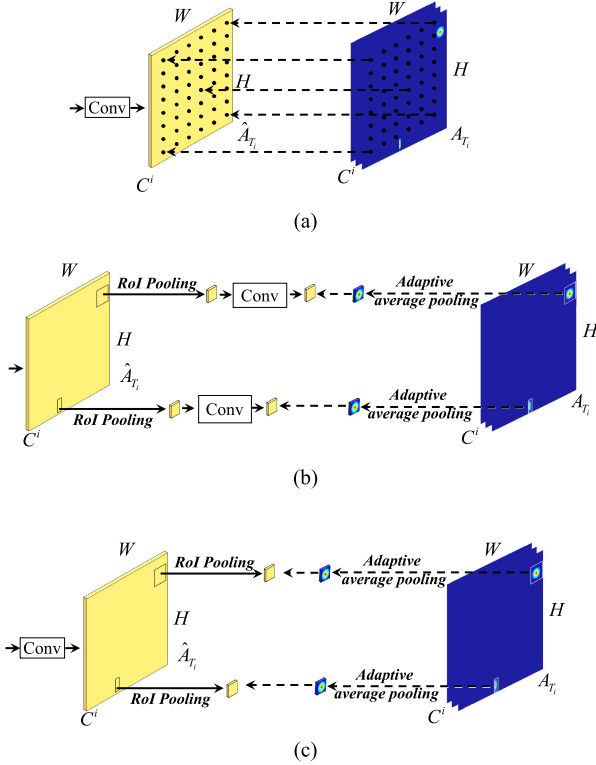


Fig. 4. Auxiliary task supervised learning: (a) and (b) commonly used in the framework of semantic segmentation and instance segmentation, and (c) proposed supervised learning method. The shared convolutional subnetwork accelerates the model reasoning, and the RoI-wise supervised learning alleviates the imbalance between positive and negative samples. (a) Shared convolutional subnetwork + Pixel-wise supervised learning. (b) RoI-wise subnetwork + RoI-wise supervised learning. (c) Shared convolutional subnetwork + RoI-wise supervised learning.

In the prediction stage, a  $1 \times 1$  convolution is employed to predict  $\hat{A}_{T_i}$  for task  $i$ . The number of output channels of the  $1 \times 1$  convolution kernel is determined by the task. For edge prediction, the output channel of the convolution kernel is 1, and the output uses the sigmoid function for nonlinear mapping. For semantic task prediction, the output channel is the number of object categories  $C$ . When  $C = 1$ , the output is activated by the sigmoid function; otherwise, it is processed using a softmax function. The number of positive samples (such as foreground edges) in the auxiliary task supervision information is considerably smaller than the number of negative samples (such as the background); hence,  $\hat{A}_{T_i}$  is up-sampled according to the size of the input image. To extract high-level semantic information, the convolution block contains multiple sets of convolution, batch normalization, and rectified linear unit activation.

4) *Auxiliary Task Supervised Learning*: The supervised learning methods for auxiliary tasks in MTL-Det are discussed in this section. The three types of supervised learning strategies are illustrated in Fig. 4. Among them, those shown in Fig. 4(a) and (b) are commonly used in semantic segmentation and instance segmentation. For the method shown in Fig. 4(a), all learnable layers are convolutional and shared across the entire image. Moreover, the convolutional layers encode spatial information to derive the prediction map  $\hat{A}$ . The production of

TABLE I  
POSITIVE AND NEGATIVE SAMPLES FOR DIFFERENT TYPES OF SUPERVISED LEARNING METHODS: AVER. IS THE AVERAGE NUMBER OF SAMPLES IN EACH IMAGE, AND SUM. IS THE NUMBER OF SAMPLES IN THE ENTIRE DATASET

Supervised learning method	No. positives		No. negatives		Proportion
	Sum.	Aver.	Sum.	Aver.	
Pixel-wise	4008970	715	3582551030	639284	1:894
RoI-wise	4048109	722	26630541	4752	1:7

an RoI-wise subnetwork is shown in Fig. 4(b). After obtaining RoI features, an RoI-wise convolution operation is required to obtain the instance-wise prediction map  $\hat{A}$ . The former method outperforms the latter with respect to computational complexity and inference speed; however, it also has a severe problem regarding the imbalance between positive and negative samples. The imbalance causes the negative samples to be well studied, inhibiting the learning of hard-positive samples. Considering edge learning as an example, we define edge pixels as positive samples and the other pixels as negative samples. Table I lists the proportions of positive and negative samples under the two supervision modes. The number of positive and negative samples is balanced using RoI-wise supervised learning.

A novel supervised learning method is proposed for auxiliary tasks, as shown in Fig. 4(c). The convolutional layers are shared throughout the input feature set to ensure the inference speed. To overcome the problem of unbalanced samples, RoI-wise supervised learning is employed. Specifically, the network learns function  $\mathcal{F}_{\Theta}(\cdot)$ , which produces the prediction  $\hat{A}_{T_i}$  for task  $i$ . Function  $\mathcal{F}_{\Theta}(\cdot)$  is composed of convolutional layers shared throughout the input feature set. Given these proposals, RoI pooling is applied to  $\hat{A}_{T_i}$  and the supervision information  $A_{T_i}$  to obtain the instance-wise prediction map  $\hat{A}_{T_i}^{ins}$  and supervision map  $A_{T_i}^{ins}$

$$\begin{aligned}\hat{A}_{T_i}^{ins} &= \text{RP}(\mathcal{F}_{\Theta}(f_{in}), \text{proposals}) \\ A_{T_i}^{ins} &= \text{RP}(A_{T_i}, \text{proposals})\end{aligned}\quad (5)$$

where RP is RoI pooling.

### C. Main Task Learning

As shown in (1), the main task learning employs the outputs of the task-guided and backbone networks as input to obtain the object-specific features for prediction. In this section, multitask feature fusion and multiscale feature alignment (MSFA) are performed to obtain object-specific features.

1) *Multitask Features Fusion*: Multitask feature fusion merges auxiliary task features, as shown in Fig. 5. Given the auxiliary task features  $f_{T_i} \in \mathbb{R}^{C_{\text{atf}} \times H_{\text{atf}} \times W_{\text{atf}}}$  of task  $i$  and  $f_{T_j} \in \mathbb{R}^{C_{\text{atf}} \times H_{\text{atf}} \times W_{\text{atf}}}$  of task  $j$ , element-wise addition [see Fig. 5(a)] is the simplest fusion method. However, the feature redundancy among auxiliary task features increases the complexity and difficulty of network training. A feature fusion module based on the attention mechanism was presented to improve the accuracy and generalization ability of the model. Specifically, each auxiliary task feature  $f_{T_i}$  is fed into a  $3 \times 3$  convolution layer and sigmoid activation layer to generate the feature selection

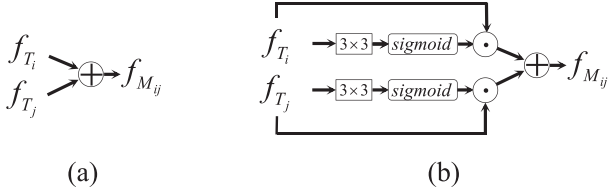


Fig. 5. Multitask feature fusion module: (a) Element-wise addition fusion; (b) Attention mechanism-based fusion.  $f_{T_i}$  and  $f_{T_j}$  are the auxiliary task features of task  $i$  and task  $j$ , respectively.  $f_{M_{ij}}$  is the multimodal feature.  $\oplus$  denotes the element-wise addition and  $\odot$  represents the Hadamard product.

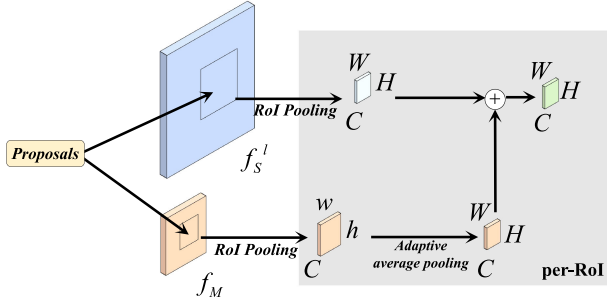


Fig. 6. MSFA. The multiscale feature  $f_S$  and multimodal feature  $f_M$  are used as inputs to perform RoI pooling to obtain a series of RoI features. Adaptive average pooling is applied to align and fuse the corresponding RoI features.

matrix  $S_i \in \{0, 1\}^{H_{\text{atf}} \times W_{\text{atf}}}$ . Feature selection is carried out by calculating Hadamard product of  $S_i$  and  $f_{T_i}$ .  $S_i$  is optimized by the back-propagation algorithm, and the network adaptively selects the features that can improve model performance. After feature selection, element-wise addition is applied to obtain the multimodal fusion feature  $f_M \in \mathbb{R}^{C_{\text{atf}} \times H_{\text{atf}} \times W_{\text{atf}}}$

$$f_M = \sum_{i=1}^T \text{Sigmoid}(W^3 * f_{T_i}) \odot f_{T_i} \quad (6)$$

where  $W^3$  represents the  $3 \times 3$  convolution kernels,  $*$  represents the convolutional operation,  $\odot$  represents the Hadamard product, and  $T$  represents the total number of tasks.

2) *Multiscale Features Alignment*: The MSFA aligns the multimodal fusion feature  $f_M$  and multiscale feature  $f_S$ . As an example, consider the two-stage region-based detector shown in Fig. 6. The RoI pooling operation is applied to  $f_M$  and  $f_S$  to obtain the corresponding RoI features  $\{r_M^i\}_{i=1}^B \in \mathbb{R}^{B \times Ch \times N \times N}$  and  $\{r_S^i\}_{i=1}^B \in \mathbb{R}^{B \times Ch \times M \times M}$ , for the given proposals. Here,  $B$  and  $Ch$  represent the batch size and number of output channels, respectively;  $N$  and  $M$  are the number of partition bins defined in RoI pooling.

For each feature  $r_M^i$  in  $\{r_M^i\}_{i=1}^B$ , the adaptive average pooling applied to  $r_M^i$  is referenced by  $r_S^i$ . Finally, scale-aligned features are combined to obtain the object-specific features  $r_P^i$  for prediction

$$\begin{aligned} r_P^i &= \text{AAP}(r_M^i) + r_S^i \\ &= \text{AAP}(\text{RP}(f_M, \text{proposals}_i)) + \text{RP}(f_S, \text{proposals}_i) \end{aligned} \quad (7)$$

where AAP is adaptive average pooling, and RP is RoI pooling.

#### D. Multitask Cooperated Evolution

To reduce the impact of the imbalance between positive and negative samples, dice loss is introduced to supervise the learning of auxiliary task features. Given the prediction map  $\hat{\mathcal{A}}_{T_i}^{\text{ins}}$  and supervision map  $\mathcal{A}_{T_i}^{\text{ins}}$ , the dice loss is defined as

$$\mathcal{L}_{T_i} = 1 - \frac{2 |\hat{\mathcal{A}}_{T_i}^{\text{ins}} \cap \mathcal{A}_{T_i}^{\text{ins}}|}{|\hat{\mathcal{A}}_{T_i}^{\text{ins}}| + |\mathcal{A}_{T_i}^{\text{ins}}|} \quad (8)$$

where  $|\hat{\mathcal{A}}_{T_i}^{\text{ins}} \cap \mathcal{A}_{T_i}^{\text{ins}}|$  represents the common elements between the prediction and supervision maps. It is approximated by the sum of the element-wise multiplication between the prediction and target mask.

The following can be inferred from (8) [38]:

$$\begin{aligned} \mathcal{L}_{T_i} &= \frac{1}{Z} \left( 1 - \sum_{b,c,h,w=1}^{B,C,H,W} \frac{\hat{\mathcal{A}}_{bchw} \times \mathcal{A}_{bchw} + \varepsilon}{\hat{\mathcal{A}}_{bchw} \times \hat{\mathcal{A}}_{bchw} + \mathcal{A}_{bchw} \times \mathcal{A}_{bchw} + \varepsilon} \right) \end{aligned} \quad (9)$$

where  $Z$  denotes the number of elements in the prediction map. The smooth factor  $\varepsilon$  was introduced to prevent the denominator from becoming 0.  $B$ ,  $C$ ,  $H$ , and  $W$  denote the batch size, the number of channels, the height, and the width of the prediction map  $\hat{\mathcal{A}}$  and supervision map  $\mathcal{A}$ , respectively.  $\mathcal{A} = [\mathcal{A}_{bchw}]_{B \times C \times H \times W}$ ,  $\hat{\mathcal{A}} = [\hat{\mathcal{A}}_{bchw}]_{B \times C \times H \times W}$ .

Cross-entropy and smooth  $L1$  losses are used for classification and regression tasks. The loss function of MTL-Det is given by the following:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{reg}} + \sum_{i=1}^T \omega_i \mathcal{L}_{T_i}. \quad (10)$$

Multitask learning is aimed at assisting the main task; however, the excessive loss contribution of auxiliary tasks can lead to an imbalance in task learning that can impact the learning of the main task. Thus, the performance of the multitask learning-based detector depends considerably on the relative weights of the loss of each auxiliary task. To set the loss weights of different subtasks, grid search and differentiable learning were utilized. For the grid search, the loss weights of two auxiliary tasks are individually chosen from  $\{0.1, 0.3, 0.5, 0.7, 1.0\}$  at random. The grid search considers various weight combinations; however, considerable training time and computing resources are required. For this reason, two DWMs are introduced. These are inspired by the techniques [39], [40], in which edge learning and semantic learning are regarded as regression tasks.

Let  $\mathcal{F}_{\Theta_i}(\mathbf{x})$  be the output of the network of auxiliary task  $i$  with weights  $\Theta_i$  on input  $\mathbf{x}$ . The likelihood of task  $i$  is defined as a Gaussian distribution

$$p(y_i | \mathcal{F}_{\Theta_i}(\mathbf{x})) = \mathcal{N}(\mathcal{F}_{\Theta_i}(\mathbf{x}), \sigma_i) \quad (11)$$

where  $\sigma_i$  is the observation noise, and  $\mathcal{F}_{\Theta_i}(\mathbf{x})$  is the mean with respect to the output.

By (9),  $\mathcal{L}_{T_i}$  is approximated by the following:

$$\begin{aligned}\mathcal{L}_{T_i} &= \frac{1}{Z} \left( 1 - \frac{2\mathbf{y}_i \mathcal{F}_{\Theta_i}(\mathbf{x})}{|\mathbf{y}_i|^2 + |\mathcal{F}_{\Theta_i}(\mathbf{x})|^2} \right) \\ &= \frac{1}{Z} \left( \frac{\|\mathbf{y}_i - \mathcal{F}_{\Theta_i}(\mathbf{x})\|^2}{|\mathbf{y}_i|^2 + |\mathcal{F}_{\Theta_i}(\mathbf{x})|^2} \right) \\ &\approx \frac{1}{2Z^2} (\|\mathbf{y}_i - \mathcal{F}_{\Theta_i}(\mathbf{x})\|^2) \\ &= \frac{1}{C} (\|\mathbf{y}_i - \mathcal{F}_{\Theta_i}(\mathbf{x})\|^2)\end{aligned}\quad (12)$$

where  $C = 2Z^2$ .

In the penultimate transition of (12), an explicit simplifying approximation is introduced, i.e.,  $0 \leq |\mathbf{y}_i|^2 + |\mathcal{F}_{\Theta_i}(\mathbf{x})|^2 \leq 2Z$ .

The following multitask likelihood is, thus, obtained:

$$\begin{aligned}p(\mathbf{y}_1, \mathbf{y}_2 | \mathcal{F}_{\Theta}(\mathbf{x})) &= p(\mathbf{y}_1 | \mathcal{F}_{\Theta_1}(\mathbf{x})) p(\mathbf{y}_2 | \mathcal{F}_{\Theta_2}(\mathbf{x})) \\ &= \mathcal{N}(\mathcal{F}_{\Theta_1}(\mathbf{x}), \sigma_1) \mathcal{N}(\mathcal{F}_{\Theta_2}(\mathbf{x}), \sigma_2)\end{aligned}\quad (13)$$

where  $\mathbf{y}_1$  and  $\mathbf{y}_2$  represent the GT of auxiliary tasks.

The joint loss to be minimized is  $\mathcal{L}(\Theta_1, \Theta_2, \sigma_1, \sigma_2)$

$$\begin{aligned}&= -\log p(\mathbf{y}_1, \mathbf{y}_2 | \mathcal{F}_{\Theta}(\mathbf{x})) \\ &\propto \sum_{i=1}^2 \frac{1}{2\sigma_i^2} \|\mathbf{y}_i - \mathcal{F}_{\Theta_i}(\mathbf{x})\|^2 + \log \sigma_i \\ &\propto \sum_{i=1}^2 \frac{1}{2C\sigma_i^2} \|\mathbf{y}_i - \mathcal{F}_{\Theta_i}(\mathbf{x})\|^2 + \log \sigma_i \\ &\approx \sum_{i=1}^2 \frac{1}{2\sigma_i^2} \mathcal{L}_{T_i} + \log \sigma_i\end{aligned}\quad (14)$$

where  $\frac{1}{2\sigma_i^2}$  is the learnable loss weight of task  $i$ , and  $\log \sigma_i$  is the regularization term.

Considering that the regularization term in (14) may be a negative loss when  $\sigma_i \rightarrow 0$ , the loss can be rewritten as follows:

$$\mathcal{L}(\Theta_1, \Theta_2, \sigma_1, \sigma_2) \approx \sum_{i=1}^2 \frac{1}{2\sigma_i^2} \mathcal{L}_{T_i} + \log(1 + \sigma_i^2).\quad (15)$$

In summary, the loss function of MTL-Det is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{reg}} + \mathcal{L}(\Theta_1, \Theta_2, \sigma_1, \sigma_2).\quad (16)$$

## IV. EXPERIMENTAL RESULTS AND ANALYSIS

### A. Data Description

1) *High-Resolution SAR Images Dataset*: This dataset was acquired by Sentinel-1 and TerraSAR-X with mixed HH, HV, and VV polarizations for ship instance segmentation and ship detection. With an overlap ratio of 25%, 136 panoramic SAR images with the spatial resolution ranging from 5 to 1 m/pixel were cropped to patches of  $800 \times 800$  pixels. The patches and ships number 5604 and 16 951, respectively. They are divided into a training (65% SAR images) and test (35% SAR

images) sets in the Microsoft Common Objects in Context (MS COCO) [45] format.

2) *Large-Scale SAR Ship Detection Dataset-v1.0*: This dataset is composed of Sentinel-1 images in the interferometric wide swath mode, containing 15 large-size SAR images. The large-size images are cut into 9000 sub-images with the size of  $800 \times 800$  pixels. The dataset is divided into training (6000 SAR images) and test (3000 SAR images) sets in the PASCAL VOC [46] format.

Compared with HRSID, LS-SSDD-v1.0 is richer with respect to the background, i.e., ocean surface, farmlands, forests, etc., thus increasing the detection difficulty. Consequently, the use of LS-SSDD-v1.0 is more exacting.

### B. Evaluation Metrics

The evaluation criteria adopted for HRSID and LS-SSDD-v1.0 are MS COCO and Pascal VOC, respectively; the adoption is dependent on the annotation format. The main evaluation metric is the AP, which is defined based on precision and recall

$$\begin{aligned}\text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{AP} &= \int_0^1 P(r) dr\end{aligned}\quad (17)$$

where TP, FP, and FN represent true positives, false positives, and false negatives, respectively;  $P$  represents precision, and  $r$  denotes recall.

For the multiclass performance evaluation, the mean of the AP of all categories is defined as the mean AP (mAP). The mAP for PASCAL VOC is based on an intersection over union (IoU) threshold of 0.5. In contrast, the mAP in MS COCO is based across the IoU thresholds from 0.5 to 0.95 with an interval of 0.05.  $\text{AP}_S$ ,  $\text{AP}_M$ , and  $\text{AP}_L$  are used for evaluation; the three indexes represent objects with small, medium, and large scales, respectively. The area ranges of their corresponding objects are (0,  $32^2$ ), ( $32^2$ ,  $96^2$ ), and ( $96^2$ ,  $+\infty$ ).

### C. Implementation Details

The experiments were conducted by a server cluster with a 64-bit Linux operating system. The hardware includes Tesla V100 GPU (32 GB memory) and Intel(R) Xeon(R) Gold 6230 CPU @ 2.10 GHz.

For a fair comparison, all experiments were implemented based on the MMDetection<sup>1</sup> [47]. The input images were resized to  $1000 \times 1000$  pixels. The models were trained for 12 epochs, and the batch size was set to 4. For the single-stage models, the initial learning rate was set to  $1e-4$ , and for the two-stage and multistage models, the initial learning rate was set to  $4e-3$ . After the 8th and 11th epochs, the learning rate decreased by a ratio of 0.1. The other hyperparameters in this study follow the default configurations of MMDetection. The model weights

<sup>1</sup>MMDetection is an open source object detection toolbox based on PyTorch.



TABLE II  
COMPARISON OF MTL-DET WITH SOTA METHODS ON HRSID

Method	DA	ISA	SSA	Backbone	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
RetinaNet [41]	✓	×	×	ResNet-50	60.0	84.7	67.2	60.9	60.9	26.8
RetinaNet [41]	✓	×	×	ResNet-101	59.8	84.8	67.2	60.4	62.7	26.5
Faster R-CNN(*)	✓	×	×	ResNet-50	61.9	86.2	71.3	63.0	62.3	18.5
Faster R-CNN(*)	✓	×	×	ResNet-101	62.5	86.9	71.6	63.3	64.4	22.0
Mask R-CNN	✓	✓	×	ResNet-50	65.0	88.0	75.2	66.1	66.1	17.3
Mask R-CNN	✓	✓	×	ResNet-101	65.4	88.1	75.7	66.3	68.0	23.3
Mask Scoring R-CNN [42]	✓	✓	×	ResNet-50	64.1	87.6	75.0	65.3	65.8	22.2
Mask Scoring R-CNN [42]	✓	✓	×	ResNet-101	64.9	88.6	75.4	66.2	67.3	19.6
Cascade R-CNN	✓	×	×	ResNet-50	66.6	87.7	76.4	67.5	67.7	28.8
Cascade R-CNN	✓	×	×	ResNet-101	66.8	87.9	76.6	67.5	68.8	27.7
Cascade R-CNN(*)	✓	×	×	ResNeXt-101-64×4d	66.1	88.2	75.1	66.9	69.3	26.6
HTC(*)	✓	✓	✓	ResNet-50	66.8	88.0	76.2	67.9	67.3	17.0
HTC(*)	✓	✓	✓	ResNet-101	66.6	88.1	76.5	67.8	68.6	14.9
HTC(*)	✓	✓	✓	ResNeXt-101-64×4d	66.8	88.7	76.6	68.0	68.7	6.9
<i>Faster R-CNN:</i>										
MTL-Det	✓	×	×	ResNet-50	63.8[+1.9]	88.0[+1.6]	74.0[+2.7]	65.1	61.9	21.6
<i>Cascade R-CNN:</i>										
MTL-Det	✓	×	×	ResNeXt-101-64×4d	68.0[+1.9]	89.2[+1.0]	77.7[+2.6]	68.7	69.6	25.8

The Number in [-] represents relative improvement; DA, ISA, and SSA indicate whether the model requires detection annotations, instance segmentation annotations, or semantic segmentation annotations for training; and (\*) indicates re-implementation results.

pretrained on ImageNet [48] were employed to initialize the backbone networks.

#### D. Main Results on HRSID

The implementation of MTL-Det was based on faster R-CNN and cascade R-CNN. In the comparative experiment, the detectors that use different annotations (e.g., detection, semantic segmentation, and instance segmentation annotations) including SOTA single-stage, two-stage, and multistage detectors, were selected as the baseline. For each detector, various backbone networks were used for comparison.

The list in Table II indicates that for general detectors, the richer the annotations, the more the network learns complementary multimodal features, resulting in a higher gain in the detection performance. For example, when ResNeXt-101-64 × 4d-FPN is used as the backbone network, HTC improves over cascade R-CNN by 0.7% in AP; however, MTL-Det improves cascade R-CNN by 1.2% in AP over HTC. A SOTA performance is achieved by MTL-Det using only detection annotations, i.e., it achieves 89.2% AP<sub>50</sub> and 68.0% AP, which exceed all baselines. This shows that the proposed multitask learning substantially improves the representation ability of the network without requiring additional manual annotations.

The proposed MTL-Det improves the performance of detectors with different backbone networks and stages. When MTL-Det is employed to improve faster R-CNN with ResNet-50-FPN, AP reaches 63.8%, which is 1.9 points higher than that of the original faster R-CNN-FPN. This verifies the robustness and generalization performance of MTL-Det.

In addition to the detection accuracy, multitask learning can improve the quality of the detection bbox. In AP<sub>75</sub>, the proposed method is 2.7 points higher than the baseline method, indicating that multitask learning improves the position accuracy of the bboxes obtained by MTL-Det.

Fig. 7 shows the visualization results on the HRSID dataset. Owing to the influence of sea background clutter and buildings near the shore, missed alarms can potentially occur, as shown by

TABLE III  
PERFORMANCE COMPARISON ON LS-SSDD-V1.0

Method	Backbone	Off-Shore	In-Shore	All
YOLOv3 [43]	darknet-53	78.5	35.6	63.0
RetinaNet	ResNet-50	74.6	17.0	54.1
RetinaNet	ResNet-101	83.7	21.1	61.9
FCOS	ResNet-50	84.0	25.6	64.0
FCOS	ResNet-101	86.5	30.9	67.4
ATSS [44]	ResNet-50	87.9	33.3	70.0
ATSS [44]	ResNet-101	87.7	37.4	70.6
Faster R-CNN	ResNet-50	86.7	34.3	68.5
Faster R-CNN	ResNet-101	87.2	35.2	68.5
Cascade R-CNN	ResNet-50	87.5	37.4	69.8
Cascade R-CNN	ResNet-101	87.0	34.5	68.5
<i>Cascade R-CNN:</i>				
MTL-Det	ResNet-50	88.7[+1.2]	38.7[+1.3]	71.7[+1.9]
MTL-Det	ResNet-101	87.5[+0.5]	37.4[+2.9]	70.3[+1.8]

the red rectangle in Fig. 7. In addition, some ships similar to the background are easily overlooked, such as the white rectangle in Fig. 7. Compared with the baseline, the proposed MTL-Det significantly reduces the missed and false alarms.

#### E. Main Results on LS-SSDD-v1.0

The SOTA single-stage, two-stage, and multistage networks were utilized as baselines. The single-stage networks included the SOTA anchor-base and anchor-free methods. As listed in Table III, MTL-Det achieves the best performance; for example, the proposed method achieves 71.7% on AP<sub>50</sub> and improves cascade R-CNN by approximately two points. This illustrates that the proposed method improved the performance of general detectors for SAR image detection on different datasets.

The summary in Table III compares the detection performance for inshore and offshore scenes. The inshore scenes contain many backgrounds that are extremely close to the objects. Compared with the backgrounds of offshore scenes, those of false alarms are more severe. With ResNet-50 and ResNet-101 as backbones, MTL-Det increased the baseline by 1.3% and 2.9% in AP<sub>50</sub>, respectively, on inshore scenes; these are higher

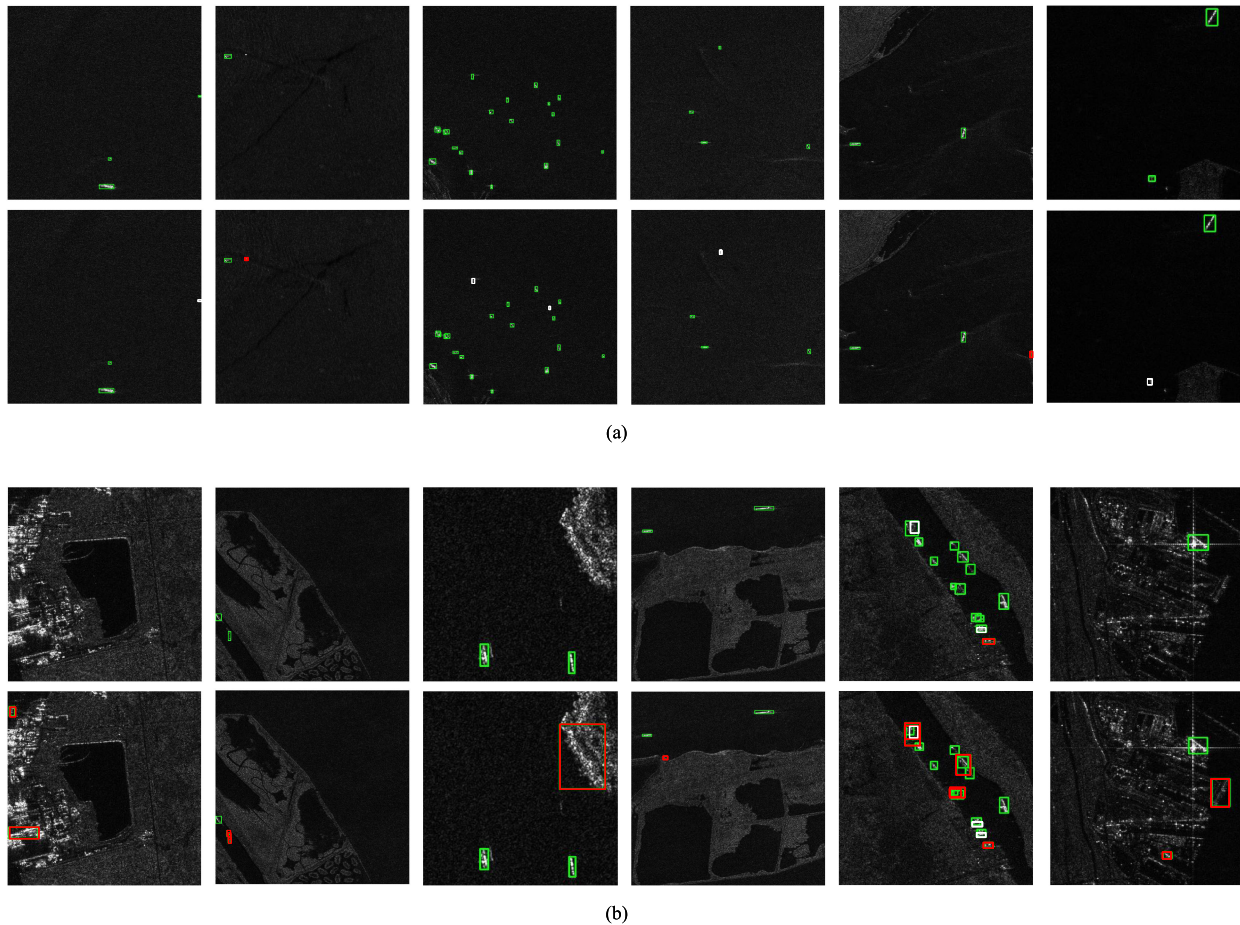


Fig. 7. Detection results on HRSID. The red, white, green rectangles represent false alarms, missed alarms, and correct detection results, respectively. A score threshold of 0.05 is used for displaying. (a) Detection results on off-shore scenes. Top: MTL-Det; Bottom: Baseline. (b) Detection results on in-shore scenes. Top: MTL-Det; Bottom: Baseline. Best viewed in zoom in.

than the performance gains on offshore scenes. This validates that multitask learning enhances the representation ability of the network, enabling the detection task to effectively overcome false alarms in complex scenarios.

The detection results for the LS-SSDD-v1.0 dataset are shown in Fig. 8. The image size is  $24\,000 \times 16\,000$ , and 600 patches with a size of  $800 \times 800$  are sent to the detector; the stitched large-size result is shown in Fig. 8. The images include inshore and offshore scenes. Faster R-CNN with ResNet-50 was used as the baseline, and MTL-Det obtained 85.8% and 82.6% on recall and  $AP_{50}$ , respectively. Compared with the baseline, the recall and  $AP_{50}$  increased by 1.2% and 1.5%, respectively.

#### F. Ablation Study

This section presents the ablation experiments that have been performed to analyze the contribution of various modules to the detection performance improvement.

1) *Ablation Study on Auxiliary Task Learning*: To analyze the performance of each auxiliary task in MTL-Det, semantic feature learning and edge feature learning were gradually applied to the baseline. The improvements resulting from the combination of the two tasks are also presented to verify the

TABLE IV  
ABLATION STUDY ON AUXILIARY TASK LEARNING

<i>Semantic</i>	<i>Edge</i>	AP	$AP_{50}$	$AP_{75}$	$AP_S$	$AP_M$	$AP_L$
		66.1	88.2	75.1	66.9	69.3	26.6
✓		67.2[+1.1]	88.3[+0.1]	76.9	68.1	69.4	26.0
	✓	67.5[+1.4]	88.4[+0.2]	77.8	68.3	70.2	28.9
✓	✓	68.0[+1.9]	89.2[+1.0]	77.7	68.7	69.6	25.8

complementary nature between the two. The baseline method is cascade R-CNN with ResNeXt-101-64  $\times$  4d; the performance levels are listed in Table IV.

As indicated by the list in Table IV, semantic learning improves the AP of baseline by 1.1 points. This result is attributed to the new type of supervision information for feature learning that the subtask provides. With the aid of auxiliary information, the detection task alleviated the false alarm caused by the complex background and improved the object probability.

The sole introduction of edge learning improved the AP of the baseline by 1.4%. Guided by the learned shape and edge of the object, the position of the bboxes can be predicted more accurately by the detection network. As summarized in Table IV, edge learning is higher by 0.9% than semantic learning on  $AP_{70}$ . This indicates that the former outperforms the latter with respect

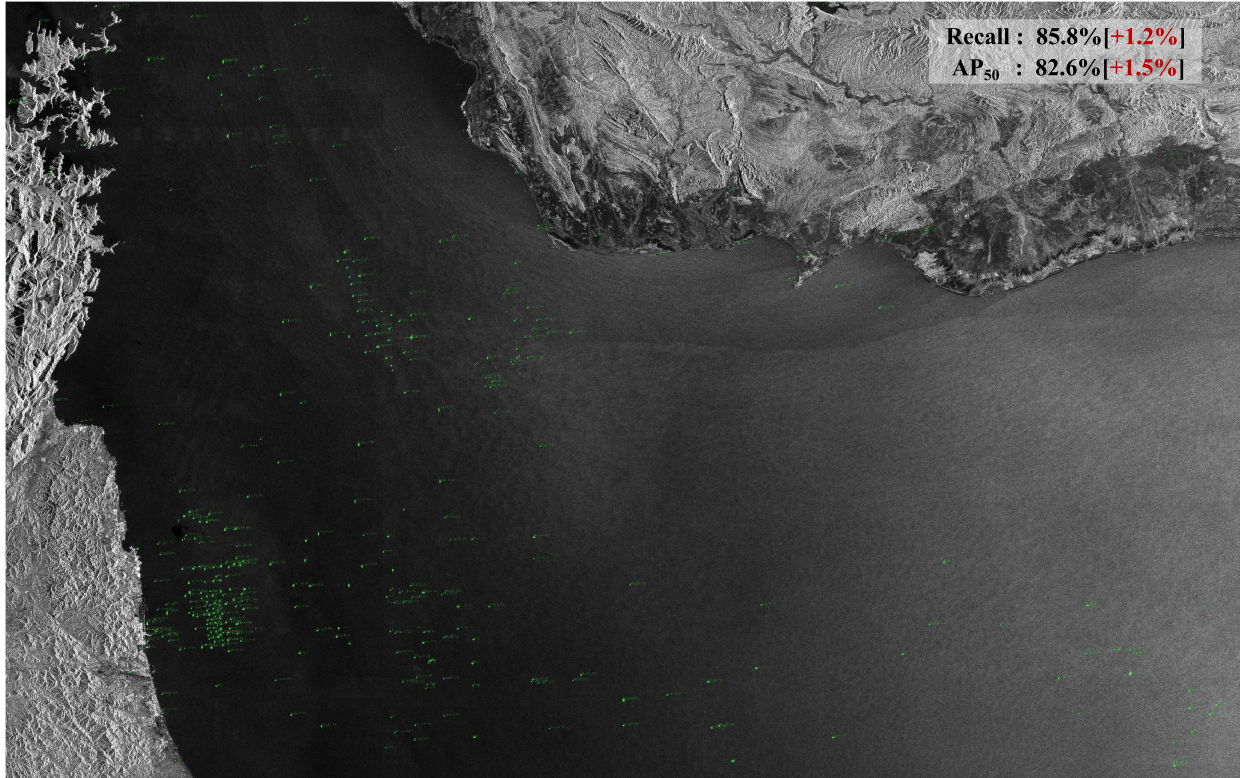


Fig. 8. Detection visualization results on LS-SSDD-v1.0. A score threshold of 0.05 is used for displaying. Best viewed in zoom in.

TABLE V  
ABLATION STUDY ON FUSION MODULE

Fusion method	AP	$AP_{50}$	$AP_{75}$	$AP_S$	$AP_M$	$AP_L$
baseline	61.9	86.2	71.3	63.0	62.3	18.5
Attention-base	63.3[+1.4]	87.6[+1.4]	72.9	64.4	64.1	18.4
Element-wise addition	62.8[+0.9]	86.6[+0.4]	72.1	63.8	64.5	18.7

TABLE VI  
ABLATION STUDY ON AUXILIARY TASK SUPERVISED LEARNING

Supervised learning method	AP	$AP_{50}$	$AP_{75}$	$AP_S$	$AP_M$	$AP_L$
baseline	61.9	86.2	71.3	63.0	62.3	18.5
Pixel-wise	62.6[+0.7]	87.1[+0.9]	72.0	63.8	62.7	11.1
RoI-wise	63.3[+1.4]	87.6[+1.4]	72.9	64.4	64.1	18.4

TABLE VII  
ABLATION STUDY ON LABEL QUALITY

Edge GT	AP	$AP_{50}$	$AP_{75}$	$AP_S$	$AP_M$	$AP_L$
ours	<b>63.3</b>	<b>87.6</b>	72.9	<b>64.4</b>	64.1	18.4
Edge GT	63.1	86.7	<b>73.2</b>	64.3	<b>64.2</b>	<b>19.1</b>

to bbox regression and aids the detection network to identify higher quality bboxes.

When the fusion module is employed to learn multimodal features, the AP improves to 68.0%, which is 1.9 points higher than the baseline. This improvement demonstrates that these tasks are complementary, and this complementarity improves the classification and regression performance in SAR ship detection.

2) *Ablation Study on Fusion Module*: Table V summarizes the ablation performance of the multitask feature fusion module; the baseline is faster R-CNN with ResNet-50. First, auxiliary

TABLE VIII  
ANALYSIS OF LOSS WEIGHT SETTING

Loss weight		AP	$AP_{50}$	$AP_{75}$	$AP_S$	$AP_M$	$AP_L$
$\sigma_{edge}$	$\sigma_{semantic}$						
0	0	61.9	86.2	71.3	63.0	62.3	18.5
1.0	1.0	63.3[+1.4]	87.6[+1.4]	72.9	64.4	64.1	18.4
0.7	0.1	<b>63.8[+1.9]</b>	87.6[+1.4]	73.2	<b>64.6</b>	<b>65.5</b>	15.4
0.5	0.3	63.6[+1.7]	<b>87.7[+1.5]</b>	<b>73.3</b>	64.5	64.5	18.6
0.1	0.7	63.5[+1.6]	87.5[+1.3]	73.1	64.4	64.1	<b>19.8</b>
0.1	0.5	63.5[+1.6]	87.3[+1.1]	73.1	64.5	64.6	13.4
1.0	0.1	63.5[+1.6]	87.4[+1.2]	73.4	64.7	63.9	16.5
DWM Eq. (14)		63.5[+1.6]	87.5[+1.3]	72.9	64.7	<b>63.7</b>	17.5
DWM Eq. (15)		<b>63.8[+1.9]</b>	<b>88.0[+1.8]</b>	<b>74.0</b>	<b>65.1</b>	61.9	<b>21.6</b>

task features are fused by element-wise addition. The list in Table V indicates that the fusion strategy is superior to the baseline, demonstrating that the fusion method improves the detection performance. To better comprehend the function of the fusion module, it is replaced by the proposed fusion method based on the attention mechanism. Therefore, the modified network is observed to be more capable in adaptively selecting features and increasing the complementarity between multimodal features and reducing feature redundancy. Table V summarizes that this method outperforms the element-wise addition methods by 0.5 points in AP (and 1.0 points in  $AP_{50}$ ). Accordingly, the attention-based fusion method is used as the default setting of MTL-Det.

3) *Ablation Study on Auxiliary Task Supervised Learning*: Table VI lists the performance levels of auxiliary task supervised learning in the ablation study. The baseline is faster R-CNN with ResNet-50. For each subtask, the feature map was obtained from

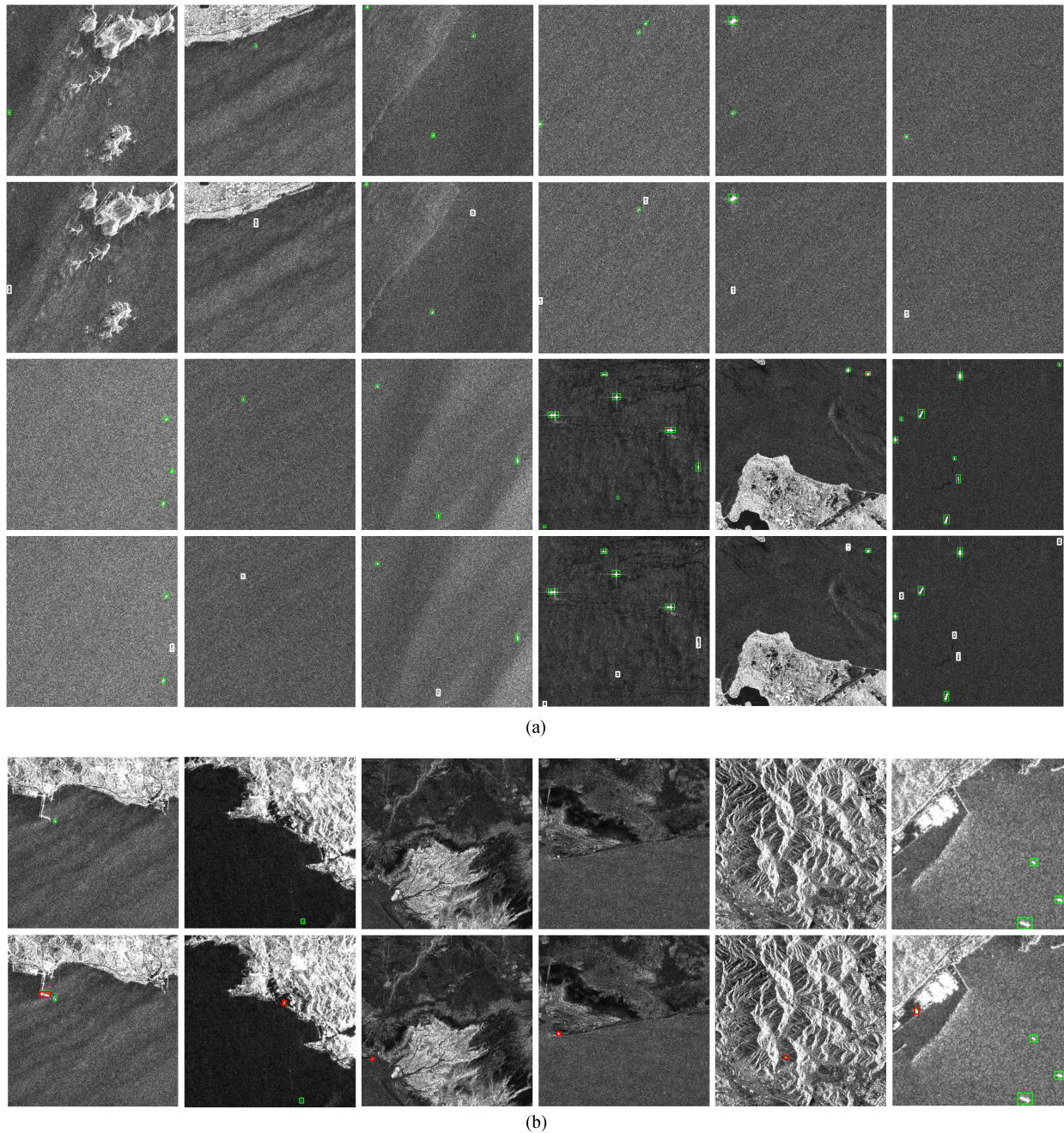


Fig. 9. Visualization analysis of robustness to speckle noise. The red, white, and green rectangles represent false alarms, missed alarms, correct detection results, respectively. Best viewed in zoom in. (a) Missed alarms. Odd rows: MTL-Det; Even rows: Baseline. A score threshold of 0.8 is used for displaying. b) False alarms. Top: MTL-Det; Bottom: Baseline. A score threshold of 0.05 is used for displaying.

the shared convolution subnetwork before the prediction map. The RoI-wise supervision and pixel-wise management were performed in multitask learning. Both methods aided MTL-Det to exceed the baseline to various levels. In contrast, the auxiliary task features were learned by the RoI-wise supervised learning better, and the model significantly improved the AP (e.g., 0.7 points higher than the pixel-wise supervised learning). This improvement is achieved because the method balances the number of positive and negative samples, and the model considers the simultaneous learning of positive and negative samples.

Accordingly, the RoI-wise supervised learning is selected as the default model learning method.

4) *Analysis of Label Quality Robustness*: In edge learning, pseudolabels are generated by the Canny operator. The experiment shows the influence of the pseudolabel quality on model learning; Table VII lists the performance levels. The semantic segmentation annotation was used to generate the edge GT of each object in training the edge learning branch. By comparing the results, the advantage of the model based on the edge GT can be deduced as enhanced by the better quality of the bbox. For

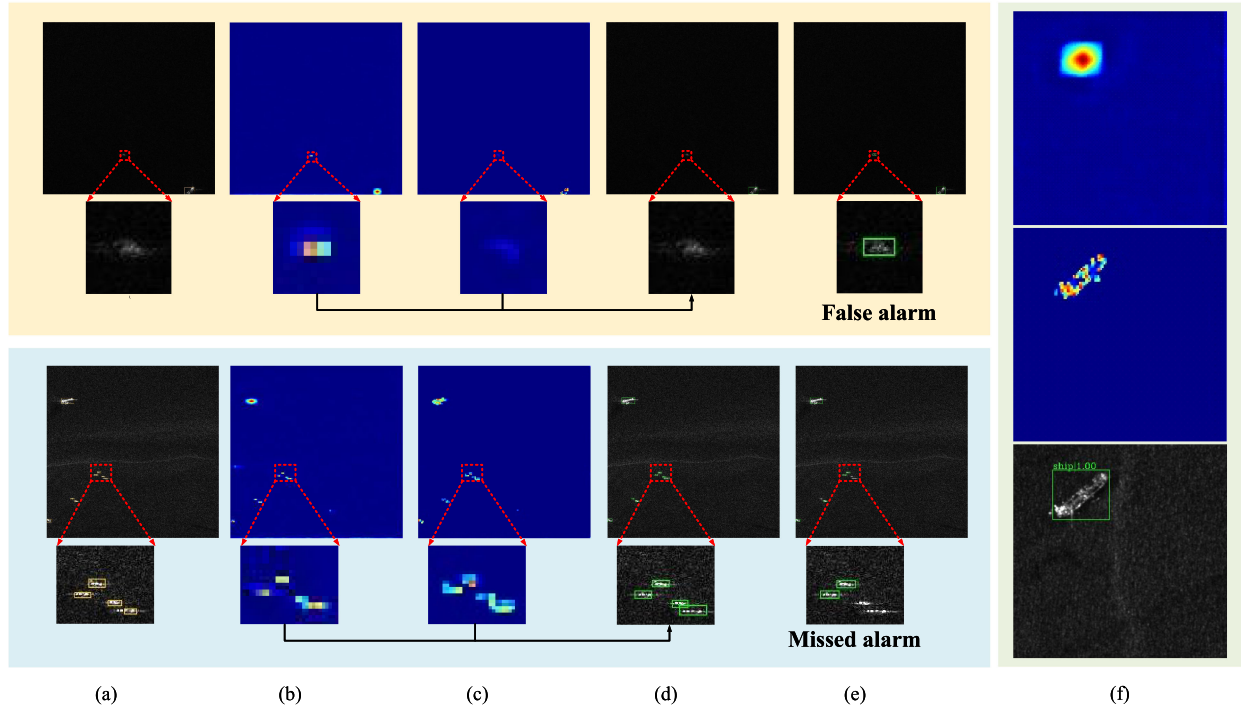


Fig. 10. Visualization analysis of multitask learning: (a) GT; (b) semantic prediction map  $\hat{A}_{\text{semantic}}$ ; (c) edge prediction map  $\hat{A}_{\text{edge}}$ ; (d) detection results of MTL-Det; (e) detection results of baseline; and (f) multitask learning results. In the pseudocolour image, the warmer the colour, the greater the response.

example,  $AP_{75}$  is slightly higher than the proposed method by 0.3%, indicating that higher quality bboxes are obtained by the accurate edge GT. However, the model with the proposed foreground edge pseudolabel generation method has an advantage over the model based on the edge GT by 0.2 points in AP (0.9 points in  $AP_{50}$ ). This confirms that the pseudolabel generation avoids many manual annotations without affecting the detection performance.

5) *Analysis of Loss Weight Setting*: The baseline sets the loss weight of subtasks as 1.0, i.e., the balance among the tasks is not considered. Table VIII lists the top-5 grid search performance levels. When the balance between subtasks is considered, the grid search can be inferred as improved by a maximum of 0.5 in AP over the baseline.

The performance levels of the DWM are listed in Table VIII. The loss weights are adjusted adaptively to balance the subtask and main task. Compared with the optimal performance obtained by grid search, the DWM performs best in the detection task. For example, using (14) to calculate the loss function, the performance levels of the detector are 63.8% in AP and 88.0% in  $AP_{50}$ , which are the highest among all weight settings. The optimal performance of the grid search obtained after 25 runs is noteworthy. However, note that the DWM is only trained once.

6) *Analysis of Robustness to Speckle Noise*: Multiple low signal-noise ratio SAR images were selected from the LS-SSDD-v1.0 dataset to demonstrate the robustness to speckle noise. Fig. 9(a) shows that MTL-Det overcomes the missed alarms caused by speckle noise. In order to increase the difficulty of the missed alarm test, the score thresh is set to 0.8. With the help of the proposed edge and semantic feature learning tasks,

MTL-Det can learn more discriminative object-specific features, and can recognize the objects from the speckle noise images with higher confidence.

Fig. 9(b) shows that MTL-Det overcomes the false alarms. Similarly, in order to increase the difficulty of the false alarm test, the score threshold is set to 0.05. Compared with the speckle noise on the sea surface, the speckle noise on the land is more likely to affect the false alarm results. The baseline model mistakenly recognizes the bright speckle on the port and land as ships. In the same case, MTL-Det is even better, which shows that MTL-Det is more robust to speckle noise than baseline. For quantitative analysis, please refer to Section IV-E, detection results comparison on LS-SSDD-v1.0 dataset.

7) *Visualization Analysis of Multitask Learning*: For the analysis of multitask learning, the prediction maps  $\hat{A}_{\text{semantic}}$  and  $\hat{A}_{\text{edge}}$  corresponding to the two proposed subtasks are shown in Fig. 10(f). The two tasks were effectively trained by the multitask learning module and supervised learning method proposed in this article. Semantic learning aids in predicting the existence probability of an object, whereas edge learning is beneficial for perceiving the shape and edge information of the object.

The proposed multitask learning effectively deals with false alarms (yellow region) and missed alarms (blue region), as shown in Fig. 10. Semantic learning increases the confidence of the foreground object and improves the recall rate; however, it may also produce false alarms. In edge learning, the influence of false alarms on detection is weakened by perceiving the shape and edge information. Moreover, the detector is more powerful in improving the quality of the bbox by identifying the object scale. Thus, the two tasks are complementary, and the proposed

multitask learning has considerable potential for ship detection from SAR images.

## V. CONCLUSION

A novel multitask learning framework, MTL-Det, is proposed in this article for ship detection from SAR images. The framework consists of an auxiliary task learning module, a main task learning module, and a multitask evolution module. Specifically, edge learning and semantic feature learning are advantageous for improving the classification and regression quality of the detector. The feature fusion module based on the attention mechanism and MSFA are beneficial for obtaining object-specific features for prediction. An auxiliary task supervised learning method is formulated to train each proposed TGN efficiently, and a DWM is devised to further balance learning among multiple tasks. Without introducing additional features, MTL-Det achieves a 1.9% improvement over the multistage detector baseline (cascade R-CNN). Similarly, it improves by 1.9% over the two-stage detector baseline (faster R-CNN) on the HRSID dataset. Finally, the SOTA performance is achieved by MTL-Det on HRSID and LS-SSDD-v1.0.

## REFERENCES

- [1] I. G. Cumming and F. H.-C. Wong, *Digital Processing of Synthetic Aperture Radar Data: Algorithms and Implementation*. Norwood, MA, USA: Artech House, 2005.
- [2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.
- [3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [4] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6154–6162.
- [5] S. Wei, X. Zeng, Q. Qu, M. Wang, H. Su, and J. Shi, "HRSID: A high-resolution SAR images dataset for ship detection and instance segmentation," *IEEE Access*, vol. 8, pp. 120 234–120254, 2020.
- [6] T. Zhang *et al.*, "LS-SSDD-v1. 0: A deep learning dataset dedicated to small ship detection from large-scale sentinel-1 SAR images," *Remote Sens.*, vol. 12, no. 18, p. 2997, 2020.
- [7] J. Zhao, R. Jiang, X. Wang, and H. Gao, "Robust CFAR detection for multiple targets in  $k$ -distributed sea clutter based on machine learning," *Symmetry*, vol. 11, no. 12, p. 1482, 2019.
- [8] S. Kuttikkad and R. Chellappa, "Non-Gaussian CFAR techniques for target detection in high resolution SAR Images," in *Proc. 1st Int. Conf. Image Process.*, 1994, vol. 1, pp. 910–914.
- [9] D. Fernandes, "Segmentation of SAR images with Weibull distribution," in *Proc. Sens. Managing Environment IEEE Int. Geosci. Remote Sens. Symp.*, 1998, vol. 1, pp. 24–26.
- [10] C. Brekke and S. N. Anfinsen, "Ship detection in ice-infested waters based on dual-polarization SAR imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 3, pp. 391–395, May 2011.
- [11] X. Qin, S. Zhou, H. Zou, and G. Gao, "A CFAR detection algorithm for generalized gamma distributed background in high-resolution SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 4, pp. 806–810, Jul. 2013.
- [12] G. Gao, "A Parzen-window-kernel-based CFAR algorithm for ship detection in SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 3, pp. 557–561, May 2011.
- [13] H. Liu, "Fast target detection for SAR images based on weighted Parzen-window clustering algorithm," in *Proc. IEEE Int. Conf. Commun. Intell. Inf. Secur.*, 2010, pp. 164–167.
- [14] M. Weiss, "Analysis of some modified cell-averaging CFAR processors in multiple-target situations," *IEEE Tran. Aerosp. Electron. Syst.*, vol. AES-18, no. 1, pp. 102–114, Jan. 1982.
- [15] H. A. Meziani and F. Soltani, "Performance analysis of some CFAR detectors in homogeneous and non-homogeneous pearson-distributed clutter," *Signal Process.*, vol. 86, no. 8, pp. 2115–2122, 2006.
- [16] G. V. Trunk, "Range resolution of targets using automatic detectors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-14, no. 5, pp. 750–755, Sep. 1978.
- [17] J. A. Ritcey and H. Du, "Order statistic CFAR detectors for speckled area targets in SAR," in *Conf. Rec. 25th Asilomar Conf. Signals, Syst. Comput. IEEE Comput. Soc.*, 1991, pp. 1082–1083.
- [18] L. Li, L. Du, and Z. Wang, "Target detection based on dual-domain sparse reconstruction saliency in SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4230–4243, Nov. 2018.
- [19] Y. Tan, Q. Li, Y. Li, and J. Tian, "Aircraft detection in high-resolution SAR images based on a gradient textural saliency map," *Sensors*, vol. 15, no. 9, pp. 23 071–23094, 2015.
- [20] L. Zhai, Y. Li, and Y. Su, "Inshore ship detection via saliency and context information in high-resolution SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1870–1874, Dec. 2016.
- [21] S. Tu and Y. Su, "Fast and accurate target detection based on multiscale saliency and active contour model for high-resolution SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 5729–5744, Oct. 2016.
- [22] X.-C. Cui, Y. Su, and S.-W. Chen, "A saliency detector for polarimetric SAR ship detection using similarity test," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3423–3433, Sep. 2019.
- [23] C. He, M. Tu, D. Xiong, F. Tu, and M. Liao, "Adaptive component selection-based discriminative model for object detection in high-resolution SAR imagery," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 2, p. 72, 2018.
- [24] C. He, M. Tu, D. Xiong, F. Tu, and M. Liao, "A component-based multi-layer parallel network for airplane detection in SAR imagery," *Remote Sens.*, vol. 10, no. 7, p. 1016, 2018.
- [25] M. Huang, W. Xia, L. Huang, Y. Zhou, and Y. Pan, "Passive ground camouflage target recognition based on gray feature and texture feature in SAR images," in *Proc. CIE Int. Conf. Radar*, 2016, pp. 1–4.
- [26] J. Wang, T. Zheng, P. Lei, and X. Bai, "A hierarchical convolution neural network (CNN)-based ship target detection method in spaceborne SAR imagery," *Remote Sens.*, vol. 11, no. 6, p. 620, 2019.
- [27] J. Zhao, W. Guo, Z. Zhang, and W. Yu, "A coupled convolutional neural network for small and densely clustered ship detection in SAR images," *Sci. China Inf. Sci.*, vol. 62, no. 4, 2019, Art. no. 42301.
- [28] Y. Zhao, L. Zhao, B. Xiong, and G. Kuang, "Attention receptive pyramid network for ship detection in SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 2738–2756, May 2020.
- [29] Q. An, Z. Pan, L. Liu, and H. You, "Drbox-v2: An improved detector with rotatable boxes for target detection in SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8333–8349, Nov. 2019.
- [30] L. Du, L. Li, D. Wei, and J. Mao, "Saliency-guided single shot multibox detector for target detection in SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3366–3376, May 2020.
- [31] Z. Deng, H. Sun, S. Zhou, and J. Zhao, "Learning deep ship detector in SAR images from scratch," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 4021–4039, Jun. 2019.
- [32] R. Yang, Z. Pan, X. Jia, L. Zhang, and Y. Deng, "A novel CNN-based detector for ship detection based on rotatable bounding box in SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1938–1958, 2021.
- [33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [34] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1492–1500.
- [35] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.
- [36] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9627–9636.
- [37] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- [38] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 565–571.
- [39] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7482–7491.

- [40] L. Liebel and M. Körner, "Auxiliary tasks in multi-task learning," 2018, *arXiv:1805.06334*.
- [41] 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.*, 2017, pp. 2980–2988.
- [42] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6409–6418.
- [43] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [44] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9759–9768.
- [45] T.-Y. Lin *et al.*, "Microsoft CoCo: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
- [46] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
- [47] K. Chen *et al.*, "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*.
- [48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.



**Xin Zhang** received the B.S. degree in information and computing science and the M.S. degree in applied mathematics from the Beijing University of Technology, Beijing, China, in 2014 and 2018, respectively. He is currently working toward the Ph.D. degree in pattern recognition and intelligent system with the Institute of Automation, Chinese Academy of Sciences, Beijing, and University of Chinese Academy of Sciences, Beijing.

His research interests are object detection, computer vision, pattern recognition, and remote sensing.



**Chunlei Huo** (Member, IEEE) received the B.S. degree in applied mathematics from Hebei Normal University, Shijiazhuang, China, in 1999, the M.S. degree in applied mathematics from Xidian University, Xi'an, China, in 2002, and the Ph.D. degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2009.

He is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His current research

interests include remote sensing image processing, computer vision, pattern recognition, so on.



**Nuo Xu** received the B.S. degree in applied mathematics from Beijing Forestry University, Beijing, China, in 2017.

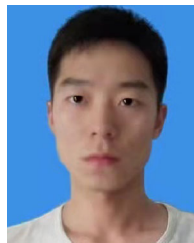
He is currently working toward the Ph.D. degree in pattern recognition and intelligent system with the Institute of Automation, Chinese Academy of Sciences and University of Chinese Academy of Sciences, Beijing.

His research interests are image processing, object detection, reinforcement learning.



**Hangzhi Jiang** received the B.S. degree in computer science and technology from Xidian University, Xi'an, China. He is currently working toward the Ph.D. degree in computer application technology with the University of Chinese Academy of Sciences, Beijing, China.

His current research interests include pattern recognition, computer vision, especially on object detection.



**Yong Cao** received the B.S. degree in automation from Northeastern University, Shenyang, China, in 2012. He is currently working toward the Ph.D. degree in pattern recognition and intelligent system with the Institute of Automation, Chinese Academy of Sciences, Beijing, China.

His current research interests include remote sensing image interpretation, computer vision, semantic segmentation.



**Lei Ni** received the B.S. degree in communication engineering from Jiangsu University Zhenjiang, Zhenjiang, China, in 2006. She is currently working toward the Ph.D. degree in space information system with Space Engineering University, Beijing, China.

Her research interests are remote sensing image processing, object detection, computer vision, and pattern recognition.



**Chunhong Pan** (Member, IEEE) received the B.S. degree in automatic control from Tsinghua University, Beijing, China, in 1987, the M.S. degree from the Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, Shanghai, China, in 1990, and the Ph.D. degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2000.

He is currently a Professor with the National Laboratory of Pattern Recognition of Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision, image processing, computer graphics, and remote sensing.