H2Det: A High-Speed and High-Accurate Ship Detector in SAR Images

Mingming Zhu, Guoping Hu, Hao Zhou, and Shiqiang Wang

***Abstract*—Synthetic aperture radar (SAR) sensor is a vital platform for ship detection whose accuracy and speed are usually difficult to balance. An urgent problem to be solved is how to achieve high-speed detection while maintaining high-accurate. To address this problem, we propose a high-speed and high-accurate detector (H2Det) in SAR images. For one thing, we adopt fewer convolutional layers, CSP module, and rectangle filling to ensure model high-speed. For another, we propose spatial pyramid pooling, bottom-up path augmentation, and mosaic data augmentation to ensure model high-accurate. To establish an optimal H2Det, we conduct comparative studies on SAR ship detection dataset (SSDD). Moreover, we verify the effectiveness of these modules mentioned above through ablation studies. The experimental results on SSDD demonstrate that both accuracy and speed of the proposed method outperform other state-of-the-art methods and references. In addition, the strong migration ability of the proposed H2Det is shown on high-resolution SAR images dataset.**

***Index Terms*—Convolutional neural network (CNN), high-resolution SAR images dataset (HRSID), high-speed, SAR ship detection dataset (SSDD), ship detection, synthetic aperture radar (SAR).**

I. Introduction

DUE TO the all-day and all-weather ability to observe the ground, synthetic aperture radar (SAR) sensor has become a vital means for ship detection. Ship detection in SAR images has attracted the attention of more and more scholars. So far, many traditional SAR ship detection methods have emerged [1]–[5]. Wang *et al.* [1] proposed an SAR ship detector via constant false alarm rate (CFAR) and image pixel information. Liang *et al.* [2] first gained candidate regions, and then achieved object classification based on a new nonparametric estimation method. Wang *et al.* [3] used a various patch-based contrast method to achieve fast ship detection. Guo *et al.* [4] proposed an improved CFAR detector via generalized extreme value and reflection symmetry metric. Lin *et al.* [5] used a superpixel-level fisher vector to distinguish the ship object and background. However, the above traditional ship detection methods have serious bottlenecks in accuracy and speed due to the difficulties in designing proper hand-crafted features for various conditions and numerous parameters predefined for specific conditions. With the great success of convolutional neural network (CNN), numerous CNN-based object detectors have been proposed. These methods can be roughly divided into two categories: two-stage detectors and one-stage detectors. The two-stage detector represented by faster R-CNN [6] first acquires a series of regions of interest, and then classifies the candidate regions. The one-stage detector represented by You Look Only Once (YOLO) [7], single shot multiBox detector (SSD) [8], and RetinaNet [9] transforms the detection problem into a regression problem. New object detection algorithms keep emerging. Cao *et al.* [10] simplified the nonlocal network and built a global context network, which can be applied to faster R-CNN. Ghiasi *et al.* [11] used neural architecture search to explore a new and optimal feature pyramid structure instead of manually design, which can be applied to RetinaNet. Pang *et al.* [12] proposed an object detector named Libra R-CNN, which can be used to reduce the imbalance during the network training. In addition, other detectors are also emerging, such as FoveaBox [13], RepPoints [14], VFNet [15], and so on.

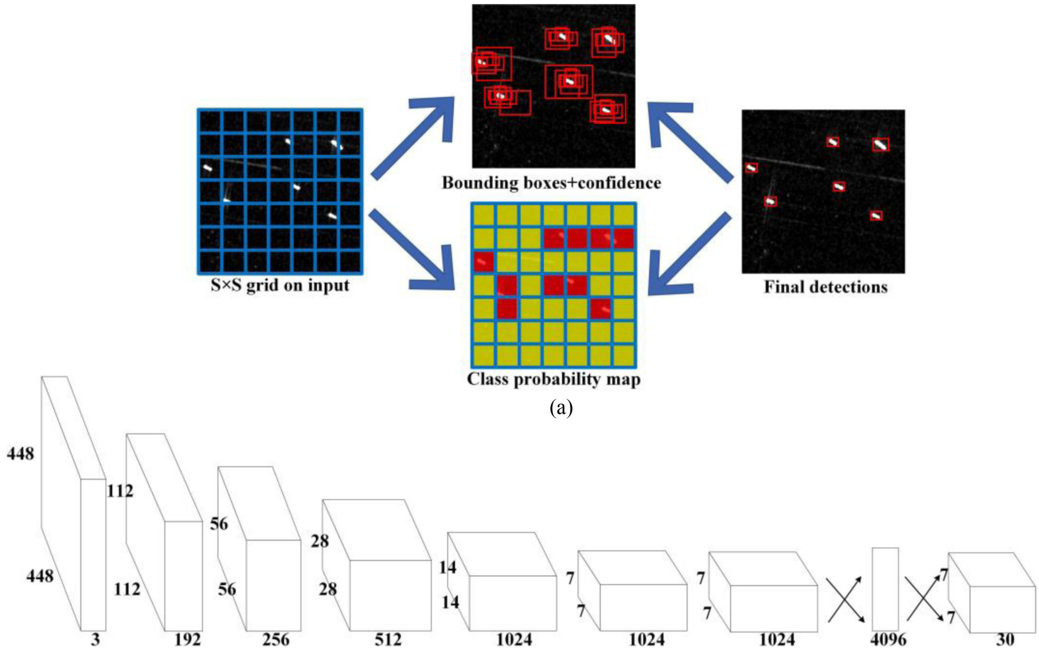
In recent years, with the emergence of large SAR images datasets, such as SAR ship detection dataset (SSDD) [16], high-resolution SAR images dataset (HRSID) [17], and large-scale SSDD-v1.0 [18], some deep-learning-based ship detection algorithms have been continuously proposed. To handle the multiscale and multiscene problems in SAR ship detection, Jiao *et al.* [19] proposed an improved faster R-CNN framework by using dense connection. Similarly, Cui *et al.* [20] also proposed a ship detection algorithm via a dense attention pyramid network to solve the multiscale difficulty. Chen *et al.* [21] proposed an improved SSD model for ship detection based on attention mechanism and feature fusion. Zhang *et al.* [22] also improved SSD model named lightweight feature optimizing network to achieve SAR ship detection.

However, most SAR ship detection algorithms are optimizing and improving detection accuracy at the expense of time cost. In addition, our investigation finds that only a few scholars such as Zhang *et al.* [23]–[26], Li *et al.* [27], and Mao *et al.* [28] have carried out research on detection speed. It is obviously unwise to sacrifice speed for improving accuracy. This is because certain occasions such as emergency military deployment and rapid maritime rescue require both high-speed and high-accurate. Thus, to address this problem, we present a high-speed and high-accurate detector (H2Det) in SAR images. The main contributions are as follows:

Manuscript received August 28, 2021; revised October 17, 2021; accepted November 24, 2021. Date of publication November 30, 2021; date of current version December 15, 2021. This work was supported by the Natural Science Foundation Research Project of Shaanxi Province, China under Grant 2020JM-345. (*Corresponding author: Guoping Hu.*)

The authors are with the Air Force Engineering University, Xi'an 710051, China (e-mail: ming_paper@163.com; hgp_kgd123@163.com; haozhou_kgd@163.com; wsq_kgd@163.com).

Digital Object Identifier 10.1109/JSTARS.2021.3131162



(a)

(b)

Fig. 1. YOLO. (a) Model of YOLO. (b) Network architecture of YOLO.

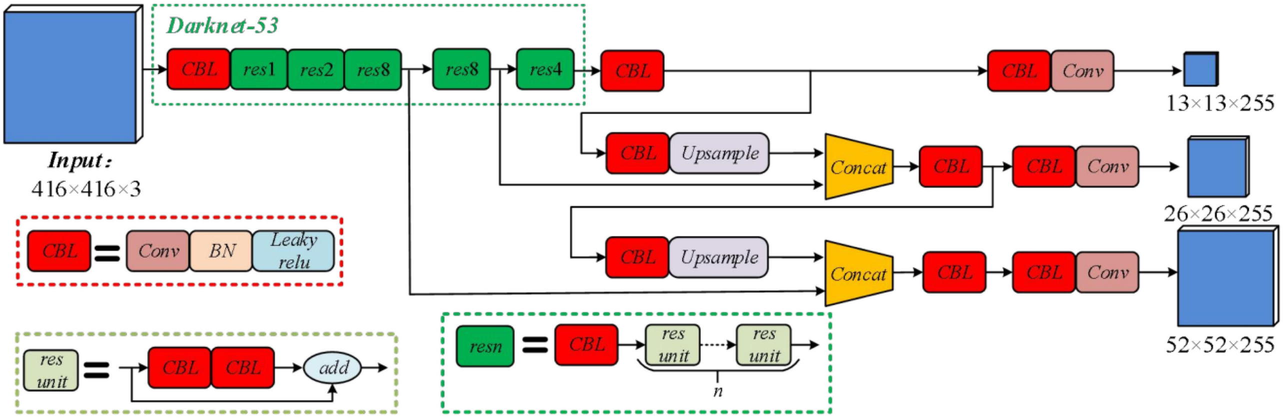


Fig. 2. Architecture of YOLOv3.

1) Fewer convolutional layers, CSP module, and rectangle filling are used for high-speed object detection.
2) Spatial pyramid pooling (SPP), bottom-up path augmentation (PA), and mosaic data augmentation are used for high-accurate object detection.
3) Both accuracy and speed of H2Det are better than other state-of-the-art methods and references.

II. Related Work

After adjusting the input image to a fixed size, YOLO directly obtains the positions and categories of all objects. As shown in Fig. 1(a), YOLO divides the image into S × S grids. If the center of an object falls into the grid, the grid is responsible for detecting the object. Each grid cell predicts B bounding boxes. Specifically, the predicted value of each bounding box contains five elements, i.e., (*x, y, w, h, c*), where (*x, y*) represent the center coordinates, (*w, h*) represent the width and height, respectively, and *c* is the confidence. In addition, each grid cell predicts *C* class probability values. Therefore, the network will eventually output a tensor of size S × S × (B∗5+C). As shown in Fig. 1(b), YOLO uses convolutional layers to extract features, and then uses fully connected layers to obtain positions and categories. For $S = 7$, $B = 2$, and $C = 20$, the size of the final output tensor of YOLO is 7 × 7 × 30. The YOLO algorithm is simple and fast, and has a low background false detection rate and a strong generalization migration ability. However, for small objects, YOLO's performance is not as good as expected. YOLOv3 [29] is the comprehensive expression of the YOLO algorithms proposed by Redmon *et al*. As shown in Fig. 2,

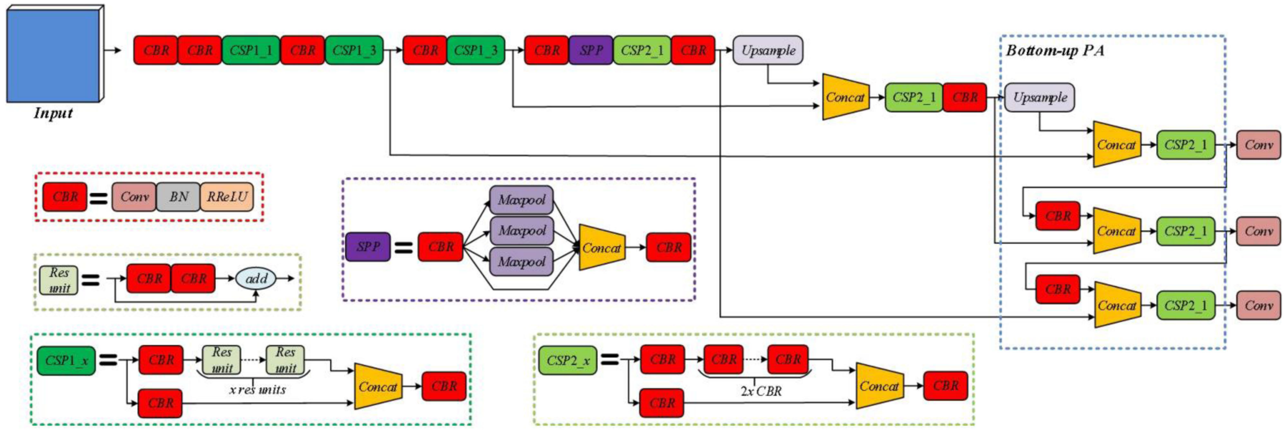


Fig. 3. Architecture of H2Det.

the most representative of YOLOv3 are Darknet-53 network and feature pyramid network (FPN). Darknet-53 uses a lot of residual connections, and uses convolutional layer instead of pooling layer to implement downsample. To enhance the detection accuracy for small object, YOLOv3 adopts FPN to perform object detection on multiple scales.

III. Proposed H2Det Method

The overall architecture of the proposed object detection method is illustrated in Fig. 3. Similar to YOLOv4 [30], we use Darknet-53 as the backbone network. To reduce the computational cost, i.e., floating-point operations (FLOPs), we use a 3×3 convolutional layer instead of the CBL and res1 module to implement downsample. Rectified linear units (ReLU) is [31] the earliest nonsaturated activation function, which is defined as follows:

$$f(x) = \begin{cases} 0, x < 0 \\ x, x \geq 0. \end{cases} \tag{1}$$

In this article, we use randomized Leaky ReLU (RReLU) as activation function, which can be considered as an improvement of ReLU. ReLU sets all negative values to zero. On the contrary, RReLU assigns a nonzero slope to all negative values, which is defined as follows:

$$f(x) = \begin{cases} kx, x < 0 \\ x, x \geq 0 \end{cases} \tag{2}$$

$$k \sim \cup(l, u), l, u \in [0, 1] \tag{3}$$

where k is a random value. In this article, we choose $l = 1/8$ and $u = 1/3$.

A. CSP Module

Huang *et al.* [32] proposed DenseNet which reused features and greatly reduced the amount of parameters. Xie *et al.* [33] introduced a new dimension, i.e., cardinality which is more effective than depth and width in improving network accuracy. Both CSPNet [34] and PRN [35] split the feature map into two parts, one of which is used to implement the convolution operation, and then is spliced with the other part. To enhance the feature perception ability of CNN and reduce the computational cost, we propose a CSP module. As shown in Fig. 3, CSP1 module uses two 1×1 convolutional layers to divide the input feature into two parts, and one part is concatenated with the other part after a series of residual units. And CSP2 module replaces the residual units in CSP1 module with a series of convolutional layers. CSP module enhances the learning ability of CNN and can ensure accuracy while reducing computational cost.

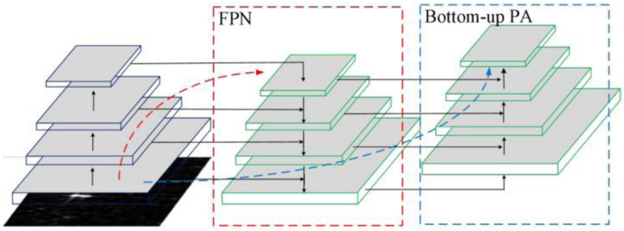


Fig. 4. Bottom-up PA.

B. Spatial Pyramid Pooling

We use SPP [36] to enhance the receptive field. First, a 1×1 convolutional layer is used to halve the number of channels of the input feature. Second, three parallel max-pooling operations with kernel size K × K are used for downsample. Then, the results of the previous step are used to concatenate with the original feature. Finally, a convolutional layer is used to restore the feature to the original scale.

C. Bottom-Up Path Augmentation

As we all know, the low-level features of CNN contain more location information, and the high-level features contain more semantic information. To improve the transmission speed and utilization of low-level information, we adopt bottom-up PA [37] on the basis of FPN. As shown in Fig. 4, the red-dashed line represents the transmission path of low-level information in FPN, and the green-dashed line is the new transmission path of low-level information. FPN transfers high-level semantic

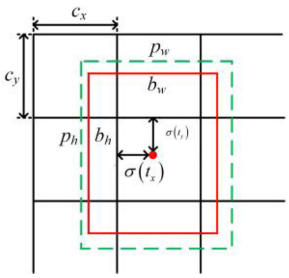


Fig. 5. Bounding box regression.

information back, and then realizes feature fusion. Bottom-up PA transfers low-level information directly upwards to improve the utilization of low-level information.

D. Model Establishment Details

1) Bounding Box Regression: In the bounding box regression process, the network will predict five values, namely t_o, t_x, t_y, t_w, and t_h. As shown in Fig. 5, assuming that the coordinate difference between the upper left corner of a grid cell and the upper left corner of the image is (c_x, c_y), and the width and height of the priori box corresponding to the cell are (p_w, p_h), then the predicted value is as follows:

$$\begin{cases} b_x = \sigma\left(t_x\right) + c_x \\ b_y = \sigma\left(t_y\right) + c_y \\ b_w = p_w e^{t_w} \\ b_h = p_h e^{t_h} \\ Pr\left(\text{object}\right) * IoU\left(b, \text{object}\right) = \sigma\left(t_o\right) \end{cases} \tag{4}$$

where IoU is Intersection over Union.

We set three priori boxes on each grid cell, so the output tensor is S × S × [3∗(5+C)] for bounding box prediction and class prediction. There are inevitably differences between different image data sets. In order to simplify the calculation and reduce the difficulty of network learning, we use K-means clustering instead of manual design to set the parameters of priori boxes. The distance measurement formula is as follows:

$$d\left(\text{box}, \text{centroid}\right) = 1 - \text{IoU}\left(\text{box}, \text{centroid}\right). \tag{5}$$

Existing bounding box regression usually uses l_n-norm loss without considering the evaluation index, i.e., IoU. Although IoU loss and generalized IoU loss introduce the IoU indicator, they still face the problems of slow convergence and inaccurate regression. Therefore, a CIoU loss is proposed, which has faster convergence speed and better regression performance.

Considering geometric factors, namely overlapping area, center point distance, and aspect ratio, the CIoU loss is defined as follows:

$$L_{\text{CIoU}} = 1 - \text{IoU} + \frac{\rho^2\left(\mathbf{b}, \mathbf{b}^{gt}\right)}{c^2} + \alpha v \tag{6}$$

where $\mathbf{b}$ and $\mathbf{b}^{gt}$ represent the center points of predicted bounding box c and ground-truth bounding box B^{gt}, respectively. ρ represents the Euclidean distance. c represents the diagonal distance of the smallest rectangle circumscribing B and B^{gt}. α is a tradeoff parameter, v is a parameter used to measure the consistency of the aspect ratio, and α and v are defined as

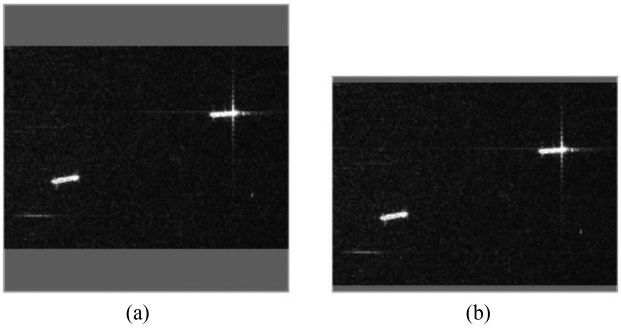


Fig. 6. Rectangle filling. (a) 640 × 640. (b) 640 × 480.

TABLE I
PSEUDOCODE OF THE RECTANGLE FILLING

```
Algorithm 1 Rectangle Filling
Input: fixed_size = (W, H); input_img = (w, h);
Output: out_img
1. Calculate the resized ratio: r = min (W/w, H/h)
2. new_w = round (w*r), new_h = round (h*r)
3. Calculate the fill value:
        dw = W-new_w, dh = H-new_h
        dw = mod (dw, 32), dh = mod (dh, 32)
        dw = dw/2, dh = dh/2
4. Resize the image: img = resize (input_img, new_w, new_h)
5. Fill the image: out_img = fill (img, dw, dh)
6. Return out_img
```

$$\alpha = \frac{v}{\left(1 - \text{IoU}\right) + v} \tag{7}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2. \tag{8}$$

2) Rectangle Filling: Different images may have different lengths and widths, so the original images need to be uniformly scaled and filled to a standard size. As shown in Fig. 6(a), the filled image has a lot of redundant information, which leads to a lot of meaningless candidate boxes. To reduce redundant information and speed-up network training, we use the rectangle filling method, as shown in Fig. 6(b). The filling value for the rectangle filling is (114, 114, 114). The downsample step of the entire network is 32. In fact, as long as the length of the image is divisible by 32, the input image does not have to be square. The pseudocode of the rectangle filling algorithm is shown in Table I.

3) Data Augmentation: Mosaic is a new data augmentation method that can enrich the background of objects and improve the performance of detecting small objects. As shown in Fig. 7, the specific flow of mosaic is as follows:

Step 1: Selecting four images at once.

Step 2: Performing operations such as flipping, zooming, and color gamut changes.

Step 3: Combining images.

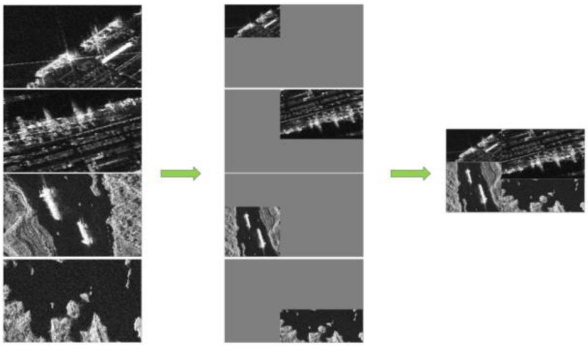


Fig. 7. Mosaic data augmentation.

IV. Experiments

The experimental platform is a private workstation with an Inter i9-9820X CPU and four GeForce RTX 2080Ti GPUs. We train all networks for 300 epochs with the initial learning rate, i.e., 0.01.

A. Dataset

SSDD: The SSDD [16] dataset is the earliest SSDD, which contains a total of 1160 SAR images from RadarSat-2, TerraSAR-X, and Sentinel-1 with ranging size from 391 × 216 to 549 × 494. We divide the entire dataset into two parts: training set (1044) and test set (116).

HRSID: The HRSID [17] dataset is a new large-scale HRSID for ship detection, which contains 5604 images and 16 951 ships with 800 × 800 pixels. We divide the entire dataset into two parts: training set (3642) and test set (1962).

Figs. 8 and 9 show the samples, shape distributions, and cluster results of two datasets. It can be seen that the samples and shape distributions of the two datasets are different. This illustrates the importance of K-means clustering for network training.

B. Evaluation Metric

The precision, recall, average precision (AP) and frames per second (FPS) are adopted as evaluation metrics [25]. The AP is defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{AP} = \int_0^1 P(R)dR \times 100\% \tag{11}$$

where TP, FP, and FN denote the numbers of true positive samples, false positive samples, and false negative samples, respectively. P(R) is the precision-recall curve.

FPS is defined as follows:

$$\text{FPS} = \frac{1}{\text{Time}}. \tag{12}$$

C. Comparative Study

To establish an optimal SAR object detection model named H2Det, we conduct comparative studies on SSDD dataset.

1) Analysis of Input Image Size: Table II shows the comparison results of H2Det method with different input image sizes, and we can obtain the following conclusions:

a) The detection accuracy (AP) is roughly proportional to the size of input image, that is to say, the larger the image size, the higher the detection accuracy. When the image size is increased to 608 × 608, the AP value reaches the maximum value i.e., 99.1%.

b) The detection speed (FPS) is inversely proportional to the size of input image, that is to say, the smaller the image size, the faster the detection speed.

c) The training time is proportional to the size of input image, that is to say, the larger the image size, the longer the training time.

Finally, we set the size of input image to 608 × 608, so as to achieve the purpose of balancing detection speed and accuracy.

2) Analysis of Activation Function: Table III shows the comparison results of H2Det method with different activation functions, and we can obtain the following conclusions:

a) The proposed H2Det method has the highest detection accuracy, i.e., 99.1% AP, which shows that the RReLU function is better than other activation functions.

b) Under different activation functions, the model parameters and GFLOPs have the same values, indicating that the model parameters and computational cost are not affected by the activation function.

Considering the detection accuracy, we choose RReLU as the activation function.

3) Analysis of Loss Function: Table IV shows the comparison results of H2Det method with different loss functions, and we can obtain the following conclusions:

a) The proposed H2Det method has the highest detection accuracy, i.e., 99.1% AP, which shows that the CIoU function is better than other loss functions.

b) Under different loss functions, the model parameters and GFLOPs have the same values, indicating that the model parameters and computational cost are not affected by the loss function.

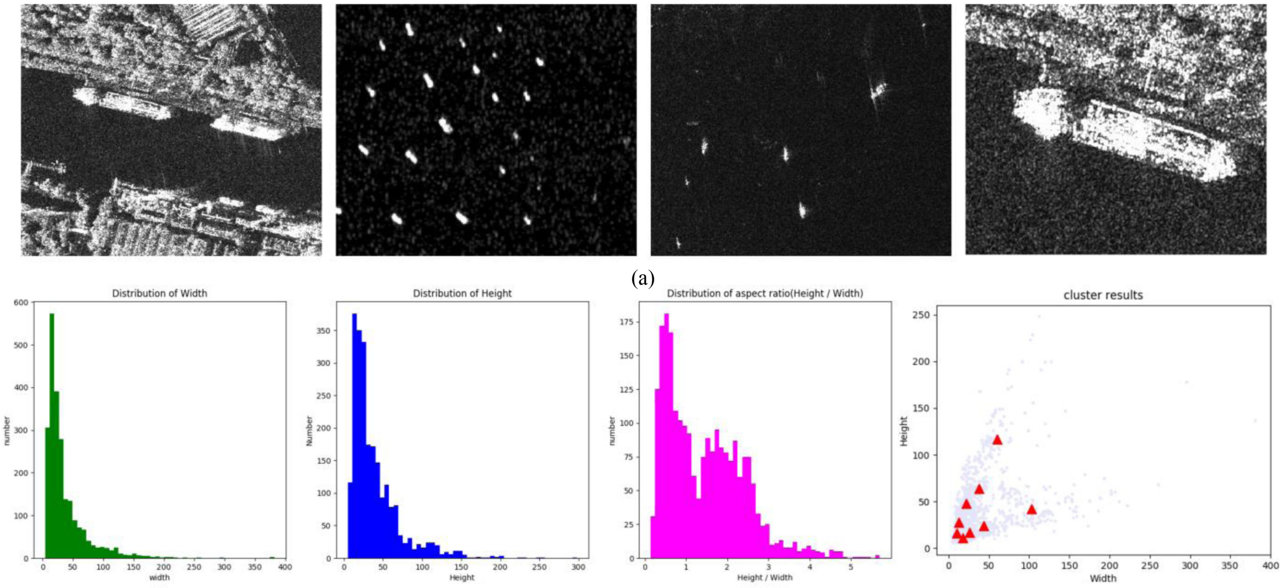
Considering the detection accuracy, we choose CIoU as the loss function.

4) Analysis of SPP Module: Table V shows the comparison results of H2Det method with different kernel sizes K in SPP module, and we can obtain the following conclusions:

a) The AP is the highest, when $K = [5, 9, 13]$. Enhancing the receptive field can improve the detection accuracy. However, when $K = 3$, AP did not increase. The possible reason is that the network training falls into the local optimal solution.

b) The model parameters and GFLOPs increase as the number of kernel increases, but the increases are relatively small.

In summary, we set $K = [5, 9, 13]$.

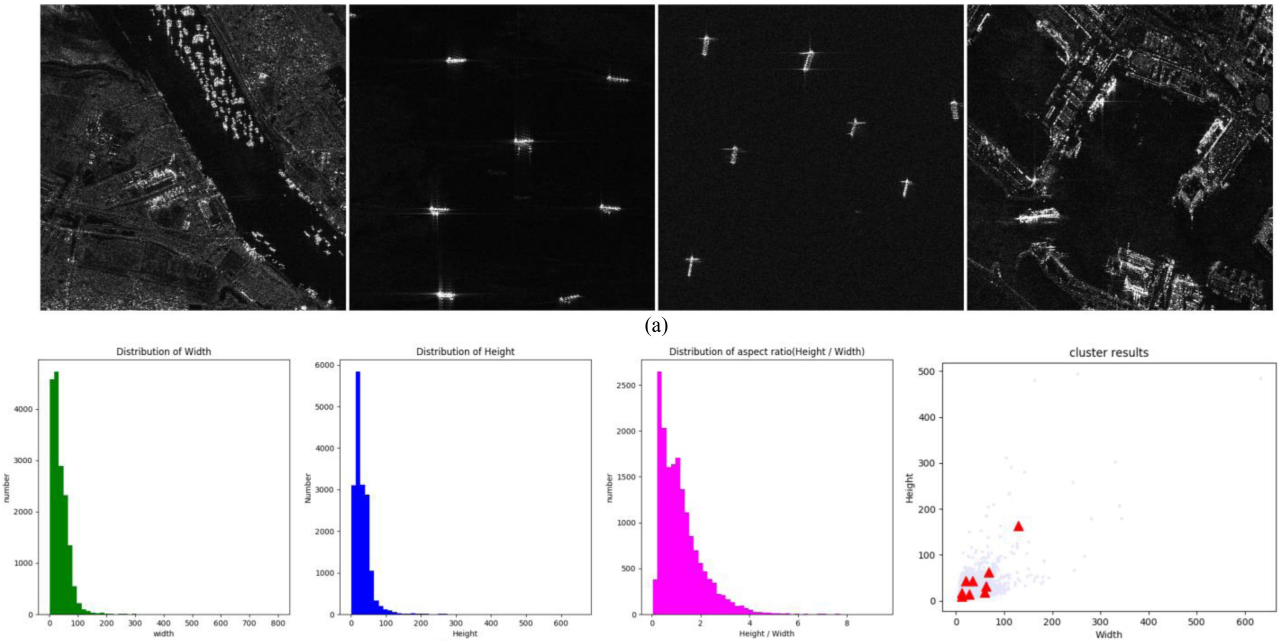


(a)

(b)

(c)

Fig. 8. SSDD dataset. (a) Samples. (b) Shape distribution. (c) Cluster results.



(a)

(b)

(c)

Fig. 9. HRSID dataset. (a) Samples. (b) Shape distribution. (c) Cluster results.

TABLE II
Results of Different Input Image Sizes

Input image size	Precision (%)	Recall (%)	AP (%)	Time (ms)	FPS	Training time (h)
160	90.0	85.8	88.7	1.4	714	0.408
320	92.3	87.5	91.9	1.9	526	0.448
480	94.0	95.2	97.2	2.6	384	0.538
512	95.8	97.8	98.2	3.1	323	0.561
544	95.8	97.8	98.5	3.3	303	0.615
576	96.5	96.6	98.9	3.5	286	0.647
608	95.7	97.0	99.1	3.6	278	0.680
640	97.8	95.7	99.0	3.8	263	0.724

TABLE III
RESULTS OF DIFFERENT ACTIVATION FUNCTIONS

Activation function	Precision (%)	Recall (%)	AP (%)	Parameters	GFLOPs
ReLU	94.1	96.1	97.9	7,051,318	15.8
PReLU	95.7	96.5	97.9	7,051,318	15.8
SiLU	97.8	95.7	98.3	7,051,318	15.8
LeakyReLU	97.8	94.4	98.4	7,051,318	15.8
Hardwish	96.9	95.7	98.7	7,051,318	15.8
RReLU	95.7	97.0	99.1	7,051,318	15.8

TABLE IV
RESULTS OF DIFFERENT LOSS FUNCTIONS

Loss function	Precision (%)	Recall (%)	AP (%)	Parameters	GFLOPs
IoU	95.7	96.5	97.7	7,051,318	15.8
GIoU	95.7	96.6	98.5	7,051,318	15.8
DIoU	95.7	96.1	98.3	7,051,318	15.8
CIoU	95.7	97.0	99.1	7,051,318	15.8

TABLE V
RESULTS OF DIFFERENT KERNEL SIZES

K	Precision (%)	Recall (%)	AP (%)	Parameters	GFLOPs
[3, 5]	93.7	96.1	97.4	6,920,246	15.7
[5, 9]	95.3	96.1	98.6	6,920,246	15.7
[3, 5, 7]	96.1	96.1	98..6	7,051,318	15.8
[5, 9, 13]	95.7	97.0	99.1	7,051,318	15.8
[5, 9, 13, 17]	94.8	95.2	97.9	7,182,390	15.9

TABLE VI
EVALUATION METRICS ON SSDD DATASET

Precision (%	Recall (%)	AP (%)	Time (ms)	FPS	Parameters	GFLOPs	Training time (h)	Model size (MB)
95.7	97.0	99.1	3.6	278	7,051,318	15.8	0.608	14.4

TABLE VII
ABLATION STUDY ON CONVOLUTIONAL COMPUTATIONAL LAYER

Convolutional computation Layer	Precision (%)	Recall (%)	AP (%)	Parameters	GFLOPs
×	95.7	96.6	98.3	7,191,750	18.1
√	95.7	97.0	99.1	7,051,318	15.8

TABLE VIII
ABLATION STUDY ON CSP MODULE

CSP module	Precision (%)	Recall (%)	AP (%)	Parameters	GFLOPs
×	96.0	93.9	97.2	7,796,051	18.5
√	95.7	97.0	99.1	7,051,318	15.8

TABLE IX
ABLATION STUDY ON SPP

SPP	Precision (%)	Recall (%)	AP (%)	Parameters	GFLOPs
×	96.1	94.8	98.5	6,395,190	15.3
√	95.7	97.0	99.1	7,051,318	15.8

TABLE X
ABLATION STUDY ON BOTTOM-UP PA MODULE

Bottom-up PA	Precision (%)	Recall (%)	AP (%)	Parameters	GFLOPs
×	98.2	95.7	98.4	6,017,974	14.5
√	95.7	97.0	99.1	7,051,318	15.8

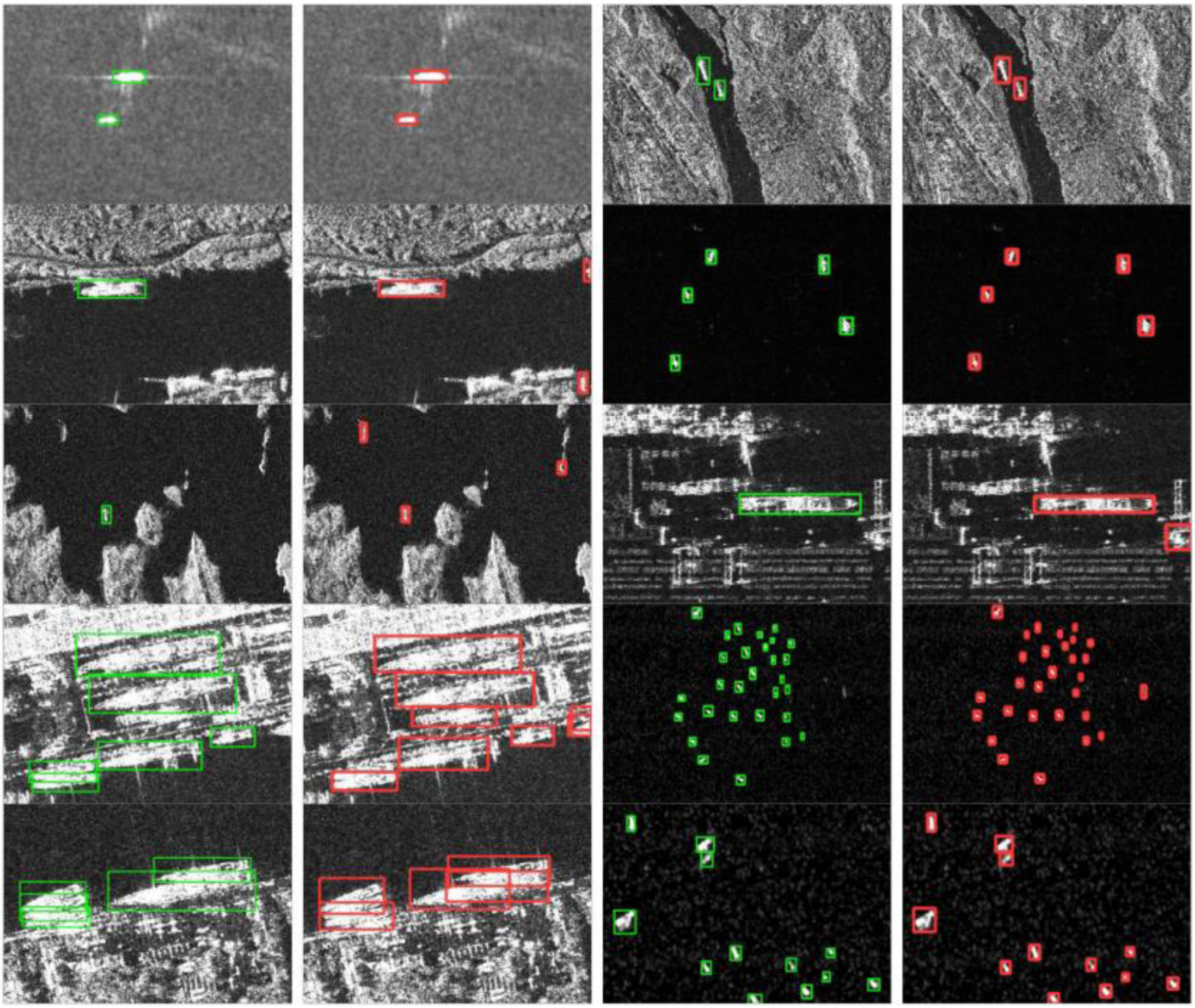


Fig. 10. Visual results on SSDD dataset. The green rectangle represents the ground-truth, and the red rectangle represents the detection result of the proposed method.

TABLE XI
ABLATION STUDY ON MOSAIC

Mosaic	Precision (%)	Recall (%)	AP (%)	Parameters	GFLOPs
×	91.1	92.7	96.5	7,051,318	15.8
√	95.7	97.0	99.1	7,051,318	15.8

D. Detection Results on SSDD

In order to visually demonstrate the detection performance of H2Det, Fig. 10 shows the detection results on SSDD dataset. It can be seen from Fig. 10 that H2Det cannot only accurately detect offshore ships, but also inshore ships. However, there are still some false detections or missed detections. For one thing, it may be because of background interference. For another, ships are so densely distributed that they missed detection.

Table VI shows the evaluation metrics on SSDD dataset. As illustrated in Table VI, we can obtain the following conclusions:

1) The proposed H2Det method achieves 99.1% AP, which is fully competent for SAR ship detection.
2) The proposed H2Det method achieves 278 FPS, which far exceeds the requirements of real-time object detection.
3) The entire training time is only 0.68 h, and the model size is only 14.4 MB, which can meet the requirements of mobile or embedded devices.

E. Ablation Study

In order to verify the effectiveness of the proposed modules, we carried out the following ablation studies:

1) *Ablation study on convolutional computation layer:* As shown in Table VII, the parameters and GFLOPs are reduced after adopting the 3 × 3 convolutional layer instead of CBL and res1 module. In addition, the AP of H2Det method is higher than before.

TABLE XII
Comparison With Other State-of-the-Art Methods

Methods	**AP (%)**	**Time (ms)**	**FPS**	**Model size (MB)**
RepPoints	83.8	54.9	18	294.0
SSD	90.4	22.6	44	190.0
RetinaNet	90.6	54.1	19	290.0
Foveabox	91.1	48.8	21	289.3
NAS-FPN	92.4	47.2	21	542.0
GHM	93.0	54.3	18	290.0
Dynamic RCNN	93.2	54.3	18	330.2
YOLOv3	93.3	18.8	53	492.5
CARAFE	93.8	58.8	17	375.0
Weight Standardization	93.8	84.0	12	340.5
Generalized Attention	94.1	55.6	18	358.6
Guided Anchoring	94.5	72.5	14	334.9
Free Anchor	94.6	54.3	18	290.0
Faster RCNN	95.1	53.8	19	330.2
Libra RCNN	95.9	57.1	18	332.3
ATSS	96.0	54.1	19	256.3
FSAF	96.0	51.8	19	289.3
PAA	96.4	79.4	13	256.3
VFNet	96.5	58.5	17	261.1
H2Det	99.1	3.6	278	14.4

TABLE XIII
Evaluation Methods of Different References on SSDD Dataset

Method	**Precision (%)**	**Recall (%)**	**AP (%)**	**Time (ms)**	**FPS**
Reference [23]	87.9	96.2	94.1	9.0	111
Reference [24]	95.7	97.8	96.9	8.7	115
Reference [26]	90.4	96.7	96.1	4.5	222
Reference [25]	87.0	98.4	97.1	4.3	233
H2Det	95.7	97.0	99.1	3.6	278

TABLE XIV
Evaluation Methods of Different Methods on HRSID Dataset

Methods	**AP (%)**	**Time (ms)**	**FPS**	**Model size (MB)**
RepPoints	79.4	45.9	22	294.0
SSD	73.8	16.2	62	190.0
RetinaNet	84.2	42.7	24	290.0
Foveabox	81.6	37.3	27	289.3
NAS-FPN	72.6	40.8	25	542.0
GHM	85.1	43.1	23	290.0
Dynamic RCNN	80.6	43.1	23	330.2
YOLOv3	86.7	17.3	58	492.5
CARAFE	81.4	46.1	22	375.0
Weight Standardization	80.2	67.6	15	340.5
Generalized Attention	80.5	43.7	23	358.6
Guided Anchoring	90..8	57.8	17	334.9
Free Anchor	87.5	43.5	23	290.0
Faster RCNN	79.9	42.2	24	330.2
Libra RCNN	79.1	44.8	22	332.3
ATSS	83.3	43.3	23	256.3
FSAF	87.1	41.2	24	289.3
PAA	82.7	55.9	18	256.3
VFNet	80.5	46.3	22	261.1
H2Det	92.0	3.9	256	14.4

2) *Ablation study on CSP module:* As shown in Table VIII, the AP improves by 1.9% after adding CSP module, and the parameters and GFLOPs decrease. The main reason is that the CSP module can enhances the learning ability of CNN and reduce computational cost.

3) *Ablation study on SPP:* As shown in Table IX, SPP module improves the AP from 98.5% to 99.1%, which suggests that SPP module is effective for enhancing the receptive field. At the same time, SPP also increases model parameters and GFLOPs, but this is acceptable.

4) *Ablation study on bottom-up PA*: As shown in Table X, bottom-up PA improves the AP from 98.4% to 99.1%, which suggests that bottom-up PA is effective for improving the utilization of low-level information. At the same time, the model parameters and GFLOPs are increased, but this is acceptable.

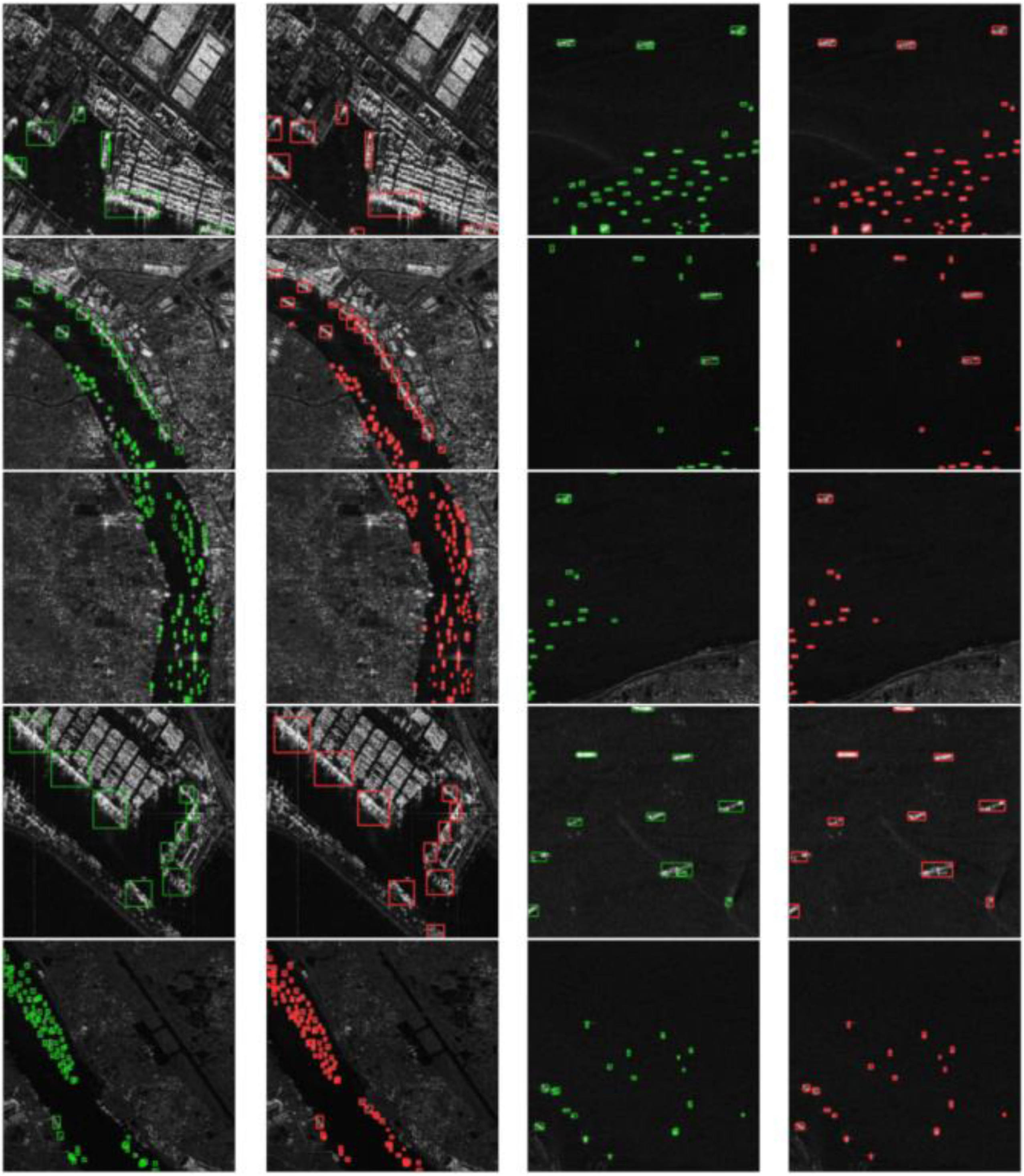


Fig. 11. Visual results on HRSID dataset. The green rectangle represents the ground-truth, and the red rectangle represents the detection result of the proposed method.

5) *Ablation study on mosaic*: As shown in Table XI, the AP is increased from 96.5% to 99.1% by adopting mosaic data augmentation, and the computational cost has not increased. The main reason is that mosaic data augmentation enriches the background of objects and improves the performance of detecting small objects.

F. Comparison With Other State-of-the-Art Methods

To demonstrate the superiority of the proposed method, we compare it with other state-of-the-art methods such as RepPoints [14], SSD [8], RetinaNet [9], Foveabox [13], NAS-FPN [11], GHM [39], Dynamic RCNN [40], YOLOv3 [29], CARAFE [41], Weight Standardization [42], Generalized Attentionn [43], Guided Anchoring [44], Free Anchor [45], Faster RCNN [6], Libra RCNN [12], ATSS [46], FSAF [47], PAA [48], and VFNet [15], which are implemented by object detection toolbox MMDetection. According to the descending order of AP, Table XII shows the evaluation metrics of different methods on SSDD dataset. As illustrated in Table XII, we can draw the following conclusions:

1) The AP of H2Det is significantly higher than other state-of-the-art methods. The highest AP value among other methods is 96.5%, but it is still 2.6% lower than the proposed method H2Det.
2) The FPS of H2Det is also significantly higher than other state-of-the-art methods. Among all methods, only SSD, YOLOv3, and H2Det achieve real-time detection. The detection speed of H2Det is about five and six times that of the other two methods, respectively. In addition, the speed of H2Det is more than 10 times that of other remaining methods.
3) The model size of H2Det is significantly smaller than other methods. This is mainly because H2Det has fewer parameters and computation costs.
4) H2Det can achieve high-accurate and high-speed object detection, but other methods cannot achieve a good balance between speed and accuracy. Therefore, the proposed method is superior to other state-of-the-art methods.

G. Compared With References

To demonstrate the superiority of H2Det, we also compare with previous work evaluated on SSDD dataset. Zhang *et al.* [23] proposed a novel high-speed SAR ship detection method based on depthwise separable convolution neural network, which consists of a depthwise convolution and a pointwise convolution. Zhang *et al.* [24] proposed a high-speed and high-accuracy SAR ship detection method called SARShipNet-20, which combines channel attention and spatial attention. Zhang and Zhang [25] proposed a lightweight SAR ship detector named ShipDeNet-20 with 20 convolution layers and 0.82 MB model size. Zhang *et al.* [26] proposed a high-accurate and high-speed SAR ship detection method named HyperLi-Net-based five external modules and five internal mechanisms. As illustrated in Table XIII, the proposed method is superior to other references and can achieve a better balance between speed and accuracy.

H. Migration Ability

Table XIV shows the evaluation metrics of different method on HRSID dataset. Consistent with the previous conclusions, H2Det's AP, FPS, and model size are better than other methods. Overall, the proposed method achieved state-of-the-art results: 92.0% AP at a speed of 256 FPS, indicating that the proposed method has strong migration ability. It can be seen from Fig. 11 that the proposed method can accurately detect most objects, but there are also some false detections and missed detections.

V. Conclusion

In this article, we have proposed H2Det for ship detection in SAR images, whose purpose is to achieve high-speed detection while maintaining high-accurate. The convolution computational layer and CSP module reduce model parameters and computational costs while improving detection accuracy. SPP and bottom-up PA improve the detection accuracy at the expense of a small amount of computational cost. Mosaic data augmentation improves detection accuracy without adding additional computational costs. In terms of speed and accuracy, H2Det has achieved better detection performance than other methods on SSDD dataset and HRSID dataset. All in all, H2Det achieves a balance of speed and accuracy.

References

[1] C. Wang, F. Bi, W. Zhang, and L. Chen, "An intensity-space domain CFAR method for ship detection in HR SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 4, pp. 529–533, Apr. 2017.

[2] Y. Liang, K. Sun, Y. Zeng, G. Li, and M. Xing, "An adaptive hierarchical detection method for ship targets in high-resolution SAR images," *Remote Sens.*, vol. 12, no. 2, 2020, Art. no. 303.

[3] X. Wang, C. Chen, Z. Pan, and Z. Pan, "Fast and automatic ship detection for SAR imagery based on multiscale contrast measure," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 12, pp. 1834–1838, Dec. 2019.

[4] R. Guo, J. Cui, G. Jing, S. Zhang, and M. Xing, "Validating GEV model for reflection symmetry-based ocean ship detection with Gaofen-3 dual-polarimetric data," *Remote Sens.*, vol. 12, no. 7, 2020, Art. no. 1148.

[5] H. Lin, H. Chen, K. Jin, L. Zeng, and J. Yang, "Ship detection with superpixel-level fisher vector in high-resolution SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 247–251, Feb. 2020.

[6] 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.

[7] J. Redmon, S. K. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015, *arXiv: 1506.02640*.

[8] W. Liu *et al.*, "SSD: Single shot multibox detector," 2015, *arXiv: 1512.02325*.

[9] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[10] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 1971–1980.

[11] G. Ghiasi, T. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7029–7038.

[12] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 821–830.

[13] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi, "FoveaBox: Beyound anchor-based object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 7389–7398, 2020

[14] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "RepPoints: Point set representation for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9656–9665.

[15] H. Zhang, Y. Wang, F. Dayoub, and N. Svnderhauf, "VarifocalNet: An IOU-aware dense object detector," 2021, *arXiv: 2008.13367*.

[16] J. Li, C. Qu, and J. Shao, "Ship detection in SAR images based on an improved faster R-CNN," in *Proc. SAR Big Data Era, Models, Methods Appl.*, Nov. 2017, pp. 1–6.

[17] 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. 120234–120254, 2020.

[18] 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, 2020, Art. no. 2997.

[19] J. Jiao *et al.*, "A densely connected end-to-end neural network for multiscale and multiscene SAR ship detection," *IEEE Access*, vol. 6, pp. 20881–20892, 2018.

[20] Z. Cui, Q. Li, Z. Cao, and N. Liu, “Dense attention pyramid networks for multi-scale ship detection in SAR images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8983–8997, Nov. 2019.

[21] C. Shiqi, Z. Ronghui, and Z. Jun, “Regional attention-based single shot detector for SAR ship detection,” *J. Eng.*, vol. 2019, no. 21, pp. 7381–7384, 2019.

[22] X. Zhang *et al.*, “A lightweight feature optimizing network for ship detection in SAR image,” *IEEE Access*, vol. 7, pp. 141662–141678, 2019.

[23] T. Zhang, X. Zhang, J. Shi, and S. Wei, “Depthwise separable convolution neural network for high-speed SAR ship detection,” *Remote Sens.*, vol. 11, no. 21, 2019, Art. no. 2483.

[24] X. Zhang *et al.*, “High-speed and high-accurate SAR ship detection based on a depthwise separable convolution neural network,” *J. Radars*, vol. 8, no. 6, pp. 841–851, 2019.

[25] T. Zhang and X. Zhang, “ShipDeNet-20: An only 20 convolution layers and <1-MB lightweight SAR ship detector,” *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 7, pp. 1234–1238, Jul. 2021.

[26] T. Zhang, X. Zhang, J. Shi, and S. Wei, “HyperLi-Net: A hyper-light deep learning network for high-accurate and high-speed ship detection from synthetic aperture radar imagery,” *ISPRS J. Photogramm. Remote Sens.*, vol. 167, pp. 123–153, 2020.

[27] Y. Li, S. Zhang, and W.-Q. Wang, “A lightweight faster R-CNN for ship detection in SAR images,” *IEEE Geosci. Remote Sens. Lett.*, to be published, doi: 10.1109/LGRS.2020.3038901.

[28] Y. Mao, Y. Yang, Z. Ma, M. Li, H. Su, and J. Zhang, “Efficient low-cost ship detection for SAR imagery based on simplified U-Net,” *IEEE Access*, vol. 8, pp. 69742–69753, 2020.

[29] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, *arXiv: 1804.02767*.

[30] A. Bochkovskiy, C. Wang, and H. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, *arXiv: 2004.10934*.

[31] V. Nair and G. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[32] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.

[33] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5987–5995.

[34] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I. H. Yeh, “CSPNet: A new backbone that can enhance learning capability of CNN,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 1571–1580.

[35] C.-Y. Wang, H.-Y. M. Liao, P.-Y. Chen, and J.-W. Hsieh, “Enriching variety of layer-wise learning information by gradient combination,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 2477–2484.

[36] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[37] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8759–8768.

[38] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-IoU loss: Faster and better learning for bounding box regression,” 2019, *arXiv: 1911.08287*.

[39] B. Li, Y. Liu, and X. Wang, “Gradient harmonized single-stage detector,” 2018, *arXiv: 1811.05181*.

[40] H. Zhang, H. Chang, B. Ma, N. Wang, and X. Chen, “Dynamic R-CNN: Towards high quality object detection via dynamic training,” 2019, *arXiv: 2020.06002*.

[41] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, “CARAFE: Content-aware reassembly of features,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3007–3016.

[42] S. Qiao, H. Wang, C. Liu, W. Shen, and A. Yuille, “Micro-batch training with batch-channel normalization and weight standardization,” 2019, *arXiv: 1903.10520*.

[43] X. Zhu, D. Cheng, Z. Zhang, S. Lin, and J. Dai, “An empirical study of spatial attention mechanisms in deep networks,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6687–6696.

[44] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, “Region proposal by guided anchoring,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2960–2969.

[45] X. Zhang, F. Wan, C. Liu, X. Ji, and Q. Ye, “Learning to match anchors for visual object detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2021.3050494.

[46] 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. 9756–9765.

[47] C. Zhu, Y. He, and M. Savvides, “Feature selective anchor-free module for single-shot object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 840–849.

[48] K. Kim and H. Lee, “Probabilistic anchor assignment with IoU prediction for object detection,” 2020, *arXiv: 2007.08103*.

Mingming Zhu received the M.S. degree in information and communication engineering from the Air Force Engineering University, Xi’an, China, in 2018, where he is currently working toward the Ph.D. degree in information and communication engineering.

His research interests include radar signal processing, deep learning, image processing, and object detection.

Guoping Hu received the graduate and M.S. degrees from the Air and Missile Defense College, Xi’an, China, in 1985 and 1990, respectively, and the Ph.D. degree in electrical engineering from Xidian University, Xi’an, China, in 2010.

He is currently a Professor and the Director with the Institute of Radar Anti-Stealth Technology, Air Force Engineering University, Xi’an, China. His research interests include radar signal processing, radar anti-stealth technology, wireless communication technology, and image processing.

Hao Zhou received the M.S. and Ph.D. degrees in information and communication engineering from the Air Force Engineering University, Xi’an, China, in 2015 and 2019, respectively.

His research interests include low altitude target detection, direction finding, and deep learning.

Shiqiang Wang received the Ph.D. degree in information and communication engineering from Air Force Engineering University, Xi’an, China, in 2012.

He is currently an Assistant Professor with the Air and Missile Defense College, Air Force Engineering University, Xi’an, China. His research interests include intelligent information processing, radar signal processing, deep learning, and radar engineering.